Trajectory Studies through Numerical Simulation of Ion Thruster Grid Region Plasma with PIC-DSMC Approach in 3D

Emre Turkoz*, Firat Sik† and Murat Celik‡
Bogazici University, Istanbul, 34342, Turkey

Abstract

Proper design of the accelerator grids plays a crucial role in the performance of an ion thruster. A PIC-DSMC code is developed to simulate the accelerator grid region plasma on a three-dimensional domain in cylindrical coordinates to assess the performance of ion thruster grids. The solution domain is a 30^o slice of a 2 mm radius cylindrical domain with 5 mm axial length. This shape provides the smallest representative domain for the assumed ion thruster accelerator grid configuration. Two species, ions and neutrals, are simulated with macro particles whereas electrons are handled with an analytical model. Collisions between the heavy species are handled using DSMC approach. Electric potential is evaluated by solving the Gauss' Law, which takes the form of the Poisson's equation. The code is implemented using the C++ programming language with multi-core parallelization. Preliminary results for various particle trajectories are presented.

Keywords: ion thruster - ion optics - plasma modeling

1 Introduction

Ion thrusters provide thrust by the acceleration of charged particles between electrostatic grids and the proper design of the accelerator grid systems are important for the performance of ion thrusters. A performance assessment of the thruster grids can be conducted through numerical simulations. The PIC-DSMC model developed in this work is used to simulate the accelerator grid region plasma of a two grid system of an ion thruster. The developed model is aimed to be used in the ion thruster design studies including the design of the BURFIT-80 ion thruster to be built at Bogazici University Space Technologies Laboratory.[1]

The subject of the ion optics studies are the investigation of the physics between the electrostatic grids that are located at the end of the discharge chamber to accelerate the charged ions out of the thruster to generate thrust. An indigenous modeling effort to simulate intra-grid and near plume plasma of an ion thruster are presented in this paper.

In the pursuit of building an RF ion thruster in our facilities,[1] a fluid model for the discharge plasma has already been developed.[2,3,4] Now with the ion optics model described in this paper, a full capability to model both the discharge chamber plasma and the grid region plasma of an RF ion thruster is being achieved. Parts of the developed ion optics model is described in a previous work.[5]

*Graduate Student, Department of Mechanical Engineering, emre.turkoz@boun.edu.tr.
†Undergraduate Student, Department of Mechanical Engineering, firats@boun.edu.tr.
‡Assistant Professor, Department of Mechanical Engineering, murat.celik@boun.edu.tr.

A numerical model for the grid region plasma involves the evaluation of the electric field imposed by the voltage difference between the grids. In this intra-grid region, the quasi-neutrality inevitably breaks down when the ions are extracted from the discharge chamber due to the externally applied potential difference. Thus, a quasi-neutral plasma assumption cannot be made to simulate the plasma in the grid region.

In the literature, there are several ion optics models: CEX2D,[6] developed at Jet Propulsion Laboratory, is capable of simulating the two-grid ion optics in two dimensions, whereas CEX3D,[7] an upgrade of CEX2D,[6] is capable of simulating the two-grid ion optics in three dimensions. Another ion optics model called ffx, developed Farnell,[8] handles both particle motion and grid sputtering effects. Another model, igx,[9] employs a useful approach for simulating the hexagonal grid patterns in three dimensions which is also the approach incorporated to the work presented in this paper. The implementation of the particle model and the collision algorithm in this work is inspired from Celik's work on plasma thruster plumes.[10]

In an ion thruster the acceleration of ions are achieved by the electrostatic potential difference applied between two or more adjacent grids. In order explain the theory, a two-grid system, consisting of a screen and an accelerator grid, is investigated. A hole pair from screen and accelerator grids is shown in Figure 1.

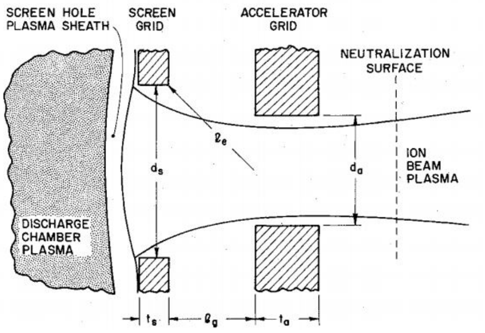

Figure 1: Representation of grid holes of the two-grid system[11]

Since ions directly affect the electric potential distribution between the grids, the ion current extraction would be restricted by the Child-Langmuir space-charge limit, and it is formulated as:

$$\mathbf{J} = \frac{4\sqrt{2}}{9}\varepsilon_0 \left(\frac{e}{m_i}\right) \frac{V_a^{3/2}}{l_g^2} \tag{1}$$

where l_g is the gap distance between the grids, V_a is the electrostatic potential difference between the grids, ε_0 is the permittivity of free space, e is the elementary charge and m_i is the mass of an ion.

If the space-charge limited current density is multiplied by the beam area, the maximum possible current can be calculated. For a beam with diameter D, the current is formulated as:

$$\mathbf{I} = \frac{\pi D^2}{4}\mathbf{J} = \frac{\pi}{4}\frac{4\sqrt{2}}{9}\varepsilon_0 \left(\frac{e}{m_i}\right)\frac{D^2}{l_g^2} V_a^{3/2} = P V_a^{3/2} \tag{2}$$

where P is called as the *perveance* of the extraction system. Taking into account the three-dimensional geometry effects, the following correction to calculate the perveance value is suggested:[11]

$$P = \frac{I_h}{V_a^{3/2}} \left(\frac{l_e}{d_s}\right) \tag{3}$$

where I_h is the beam current per hole, d_s is the screen grid aperture diameter and l_e is defined as:

$$l_e = \sqrt{l_g^2 + (d_s^2/4)} \tag{4}$$

2 Numerical Model

In this section, the implementation details of the developed ion optics code are presented. In the considered design, grids are assumed to be manufactured such that their holes are concentric with hexagons as shown in Fig. 2. A representation of the geometry where the plasma parameters are resolved in 3D is given in Fig. 2.

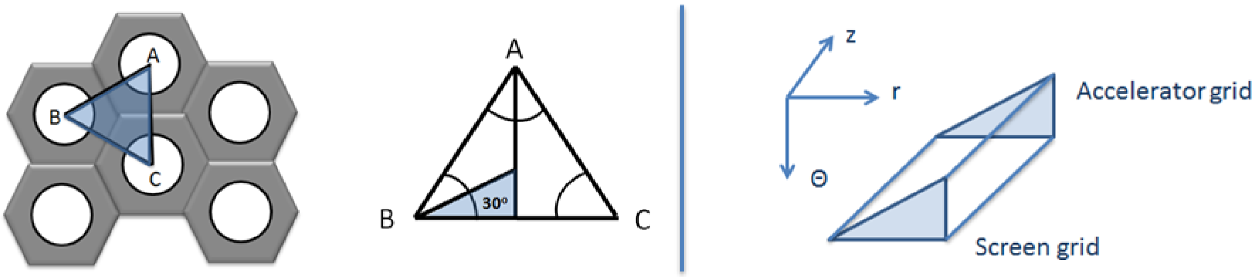

Figure 2: Ion grids geometry and the solution domain to be used in the model and presented with the introduction of the igx[9] ion optics code

2.1 Solving the Electric Potential

The equation solved to evaluate the electric potential distribution throughout the domain is the Poisson's equation:

$$\nabla^2\phi = -\frac{\rho}{\varepsilon_0} \tag{5}$$

where ρ is the net charge density and ε_0 is the permittivity of free space.

For the wedge geometry presented in Fig. 2, a structured rectangular grid is used for the sides of the 30^o angle wedge for the computational study. The grid and an example boundary condition set is presented in Fig. 3.

The electric potential equation (5) is extended to cylindrical coordinates as follows:

$$\nabla^2\phi = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial\phi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2\phi}{\partial\theta^2} + \frac{\partial^2\phi}{\partial z^2} = -\frac{\rho}{\varepsilon_0} \tag{6}$$

where r denotes the radial coordinate. The right hand side of the equation, the charge density, is calculated by weighting each macroparticle's charge to the cell nodes. According to the weighting scheme employed in cylindrical coordinates, the accumulated charge at one node of the mesh is formulated as follows

$$\rho_n = \rho_n + \frac{q_p}{V_{cell}}\left(1 - \frac{|(z_n - z_p)\pi(r_n^2 - r_p^2)(\theta_n - \theta_p)|}{V_{cell}}\right) \tag{7}$$

where ρ_n denotes the charge density of the node, q_p is the charge that the particle is carrying in Coulombs, and z, r and θ denotes the axial, radial and azimuthal coordinates, respectively. Use of ρ_n on both sides of the equation is done to indicate that this is actually a summation over each particle

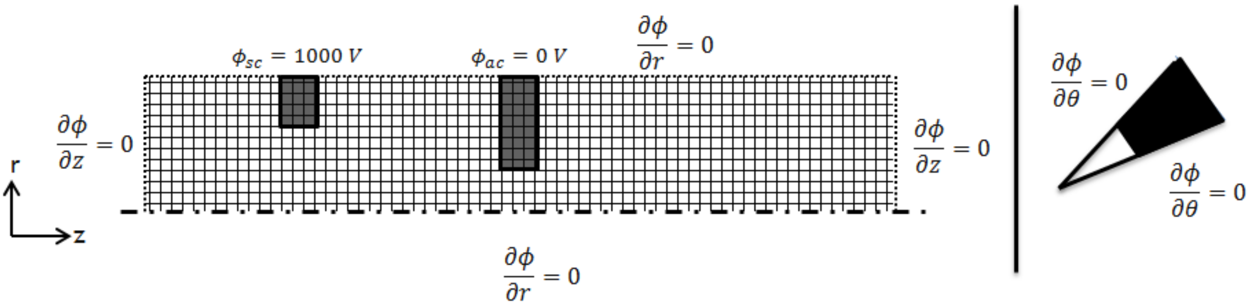
$\phi_{sc} = 1000\ V$
$\phi_{ac} = 0\ V$
$\frac{\partial\phi}{\partial r} = 0$
$\frac{\partial\phi}{\partial z} = 0$
$\frac{\partial\phi}{\partial z} = 0$
$\frac{\partial\phi}{\partial r} = 0$
r
z
$\frac{\partial\phi}{\partial\theta} = 0$
$\frac{\partial\phi}{\partial\theta} = 0$

Figure 3: Solution domain: Electric potential equation is solved in 3D cylindrical coordinates on a wedge shaped domain with a 30^o angle[9]

which is located in a cell that harbors this node. The subscript p denotes the particle and the subscript n denotes the node considered. V_{cell} denotes the cell volume in cylindrical coordinates:

$$V_{cell} = \pi \Delta r (r_L + r_U) \Delta z \Delta \theta \tag{8}$$

where r_L is the radial position of the lower node, and r_U is the radial position of the upper node of the cell. Similarly, the weighting scheme is employed to interpolate the potential gradients calculated on the mesh nodes to the particles. This scheme is as follows:

$$(\nabla\phi)_p = (\nabla\phi)_p + (\nabla\phi)_n \left(1 - \frac{|(z_n - z_p)\pi(r_n^2 - r_p^2)(\theta_n - \theta_p)|}{V_{cell}}\right) \tag{9}$$

By utilizing the weighting scheme, the total electric field on a macro particle is calculated with the contribution coming from the 8 nodes that constitute the cell that the particle is located. In this study, ions are tracked as macroparticles along with the neutral particles.

As stated, the solution domain is formulated in three dimensions, which are azimuthal, radial and axial directions on a cylindrical coordinate system. To solve for the PDE for the electric potential, the second-order finite differencing scheme is applied to evaluate the coefficients. On a structured mesh, where the mesh sizes Δr, Δz, $\Delta\theta$ are equal on the grid, the finite differencing scheme takes the form for the central nodes:

$$\nabla^2\phi = \frac{\phi_{i-1,j,k} - 2\phi_{i,j,k} + \phi_{i+1,j,k}}{\Delta z^2} + \frac{\phi_{i,j-1,k} - 2\phi_{i,j,k} + \phi_{i,j+1,k}}{\Delta r^2} + \frac{\phi_{i,j,k-1} - 2\phi_{i,j,k} + \phi_{i,j,k+1}}{r^2\Delta\theta^2} + \frac{\phi_{i,j+1,k} - \phi_{i,j-1,k}}{2\Delta r} \tag{10}$$

where i denotes the node index in the axial direction, j denotes the node index in the radial direction, and k denotes the node index in the azimuthal direction.

It can be observed from this expression that the potential of a node is directly affected by its 6 neighboring nodes in each direction. The representation of these 6 neighboring nodes and 1 center node is depicted in Fig. 4.

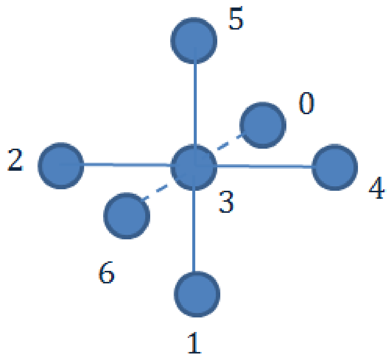

Figure 4: The 7-stencil second order finite differencing in three-dimensions

Let us call the number of nodes in the axial direction nz, number of nodes in the radial direction nr on one azimuthal slice of the structured grid in the 3D cylindrical coordinates. The numbering in Fig. 4 starts from the back face neighbor of the center node, which is denoted with the number 3. Between Node 0 and Node 3 there are $nr \times nz$ grid nodes. Node 1 denotes the radial neighbor on the south side, which has nz number node nodes in between with the Node 3. Node 2 is the west neighbor of the central node and one node behind the Node 3, and Node 4 is the east neighbor of the central node and one node ahead. Node 5 is the north neighbor and nz number of nodes ahead of Node 3. And Node 6 is the $nr \times nz$ grid nodes ahead of Node 3.

This numbering provides the opportunity to build a coefficient matrix that consists of nonzero diagonals and zero everywhere else. The generated coefficient matrix has therefore a pentadiagonal sparsity pattern. The preferred sparse matrix storage scheme is compressed diagonal storage (CDS) and it is implemented as described in.[12] To solve these linearized systems, the utilization of an iterative solution scheme is mandatory. Jacobi, Gauss-Seidel,[13] GMRES[14] and ILU preconditioned GMRES[15] methods are implemented in the software framework as solvers. These solvers are implemented considering the matrix storage scheme used. Among these solvers, it is observed that ILU-GMRES is the only one that can reduce the residual to the level below the desired tolerance value within a reasonable computational time.

The electric field is calculated as the gradient of the potential:

$$\mathbf{E} = -\nabla\phi = -\left(\frac{\partial\phi}{\partial r}\hat{\mathbf{r}} + \frac{1}{r}\frac{\partial\phi}{\partial\theta}\hat{\theta} + \frac{\partial\phi}{\partial z}\hat{\mathbf{z}}\right) \tag{11}$$

After the electric potential and the electric field are evaluated, the ions move according to the force applied on them.[5] The particle motion is performed with the commonly used Leapfrog algorithm.[16] The collision are handled using DSMC approach as described in a previous work.[5] DSMC collisions are implemented with elastic collision cross sections obtained from the literature. Three types of heavy species collisions, Xe-Xe elastic, Xe-Xe$^+$ elastic, and CEX collisions, are implemented.

2.2 Initializing Particles

Particle initialization is performed according to the assumption that particle velocities are distributed according to Maxwellian. Two particle species used in this work are ions and neutrals as stated before. Ions entering the domain from the discharge plasma have a drift velocity, which is equal to the Bohm velocity in axial direction whereas neutrals move within the grid region without any directed velocity. But their initial velocity is always taken positive in the axial direction.

A probabilistic method is used to specify the velocity for the incoming particles. In the scope of this method, a distribution function, which is Maxwellian in our case, is used to represent particle velocity probabilities:

$$\int_a^b f(x)dx = 1 \tag{12}$$

where a and b is the lower and upper limit of the velocities. Additional to this, a cumulative distribution function is introduced and defined as:

$$F(x) = \int_a^x f(x)dx \tag{13}$$

The cumulative distribution function gives the integrals of the distribution function from the lower limit of the parameters to a given reference x value.

Maxwellian velocity distribution for a single direction is formulated as:

$$f_v(v_i) = \sqrt{\frac{m}{2\pi kT}} exp\left[-\frac{mv_i^2}{2kT}\right] \tag{14}$$

where m is the mass of the particle, k is the Boltzmann's constant, T is the species temperature and v_i is the velocity at a particular direction. If a particle has a directed drift velocity in a particular direction, Maxwellian distribution takes the form:

$$f_v(v_i) = \sqrt{\frac{m}{2\pi kT}} exp\left[-\frac{m(v_i - a_i)^2}{2kT}\right] \tag{15}$$

where a_i is the drift velocity in this direction. This form of the Maxwellian distribution is applied for ions that enter the system with Bohm velocity.

A procedure is used to utilize these definitions and to find the probability of a particle having a specified velocity. This procedure is named as *Acceptance-Rejection Method.* In this procedure, first a random fraction, R_1, is generated. A reference velocity value is then picked as $v_{ref} = a + R_1(b - a)$ for a specific particle. Then the value of the cumulative distribution function $F(v_{ref})$ is calculated. To determine the probability that the particle has this velocity, a second random fraction, R_2 is generated. If the new random fraction, R_2, is larger than the cumulative distribution function value $F(v_{ref})$, the particle is assigned with the velocity v_{ref}. If the new random fraction, R_2, is smaller than $F(v_{ref})$, the process is repeated until the particle is assigned with the proper velocity component.

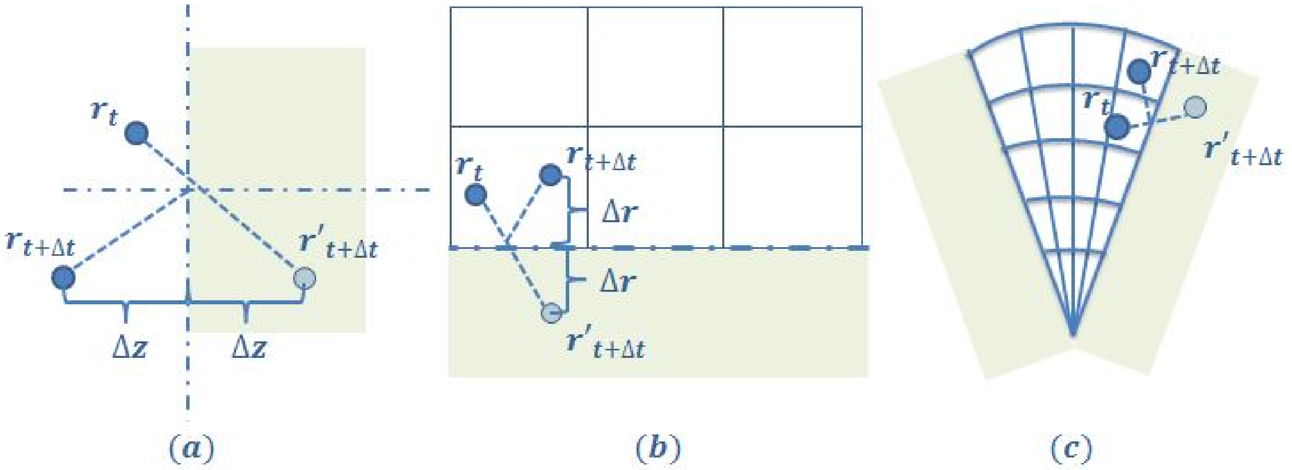

Figure 5: (a) Representation of the position change after axial reflection. (b) Representation of the position change after radial reflection from the centerline. (c) Representation of the position change after azimuthal reflection.

The problem configuration in this study requires also a little adjustment to the procedure described above. This adjustment will be explained after the details are given of the procedure to assign the positions of the initialized particles. The routine that is used to determine the initial positions of particles are the same for ions and neutrals. We impose that both of these species are entering the system from the left boundary of the domain, at the axial coordinate, $z = 0$. The remaining two coordinates are calculated by using two random fractions in a very simple manner: The radial coordinate of the particle, $r = r_{min} + R_1(r_{max} - r_{min})$, whereas the azimuthal coordinate, $\theta = \theta_{min} + R_2(\theta_{max} - \theta_{min})$.

In the velocity assignment procedure, there is a slight difference between neutral and ion species. For the ions, the cumulative distribution function value is calculated assuming a drift velocity, which is equal to the Bohm velocity, for Maxwellian distribution. For neutrals, the cumulative distribution function is calculated as usual with no directed drift velocity, so that the mean velocity is zero. But the assigned axial velocity is always positive for neutrals to prevent the unnecessary reflection in the first time step.

After velocity components are determined for each particle in the Cartesian (x, y, z) coordinates, these components are converted into the cylindrical coordinates (r, θ, z) and stored in this way. Suppose the velocity components in Cartesian coordinates are expressed as $\mathbf{v} = v_x\hat{i} + v_y\hat{j} + v_z\hat{k}$. Their counterparts in cylindrical coordinates are: $\mathbf{v} = v_r\hat{r} + v_\theta\hat{\theta} + v_z\hat{z}$. The transformation to cylindrical coordinates is formulated as:

$$v_z = v_z \tag{16}$$

$$v_r = v_x cos(\theta) + v_y sin(\theta) \tag{17}$$

$$v_\theta = (-v_x sin(\theta) + v_y cos(\theta))/r \tag{18}$$

This transformation is performed for each particle initialized. It can be seen that the azimuthal velocity component is stored as rad/s. It is converted into m/s when necessary by multiplying this value with the radial coordinate of the particle.

2.3 Collisions with the Grid Walls

Collisions with the grid walls are handled so that particles colliding with the walls are reflected back into the system with the same momentum.

The code implemented understands that a collision with grid walls occurred during the phase where the particles are relocated and reassigned to the cells they are located in. If a particle is found to be located in one of the grid cells, its location and velocity are changed accordingly.

The reflection scheme is performed as represented in Figure 5. In this figure the shaded area represents the grid wall. A particle that has the position **r** at time t is marched in time with a specific velocity and goes through a grid wall. During the particle reallocation process, the axial distance between the particle inside the grid and the grid wall is calculated. This value is shown with Δz. A calculation scheme is developed to understand whether the particle has entered from the left or right side of the grid wall. According to this scheme, the distance of the particle to the both ends of the grid is calculated. The distance of the particle to the left end is named as Δz_L, and the distance of the particle to the right end is named as Δz_R. A representation of these distances is depicted in Figure 5. So the following conditions apply to calculate the Δz value to calculate new particle position:

If $\Delta z_L < \Delta z_R$ then $\Delta z = \Delta z_L$, if $\Delta z_R < \Delta z_L$ then $\Delta z = -\Delta z_R$. The minus sign in the second expression is to take the direction of the coordinate system into account.

In this scheme it is assumed that the particle can not go through a distance that is equal to the half width of a grid within a time step.

Then the new position of the particle is calculated as:

$$\mathbf{r}_{t+\Delta t} = \mathbf{r}'_{t+\Delta t} - (2\Delta z)\hat{\mathbf{z}} \tag{19}$$

The particle's velocity is assumed to undergo a similar transformation, which will be explained below. The particle stores the velocity vector of the previous half-time step. If we express the calculated velocity as:

$$\mathbf{v'}_{t+\Delta t/2} = \dot{r}\hat{\mathbf{r}} + r\dot{\theta}\hat{\theta} + \dot{z}\hat{\mathbf{z}} \tag{20}$$

Then the transformed velocity is:

$$\mathbf{v'}_{t+\Delta t/2} = \mathbf{v'}_{t+\Delta t/2} - 2\dot{z}\hat{\mathbf{z}} \tag{21}$$

Reflection from grids is performed during the loop over cells. The cells that fall on the grid locations are tagged with a label, and the reflection is performed when there is a particle detected in these cells.

2.4 Reflection: Particles Crossing the Radial Boundaries

Particles crossing the centerline ($r = 0$) and the top boundary ($r = r_{max}$) are handled with the simple principle that for each particle leaving the domain, there should be another one entering the domain because of the axisymmetry.

The code implemented understands that a particle has crossed the centerline when its radial coordinate obtains a negative value. Similarly if a particle has a radial coordinate larger than r_{max}, the particle is reflected from the top boundary.

The reflection scheme is performed as represented in Figure 5. In this figure the green area represents the region below the centerline where $r < 0$. A particle that has the position **r** at time t is marched in time with a specific velocity and goes below the centerline. During the particle reallocation process, the radial distance between the particle below the centerline and the centerline itself is calculated. This value is shown with Δr. Then the new position of the particle is calculated as:

$$\mathbf{r}_{t+\Delta t} = \mathbf{r}'_{t+\Delta t} + (2\Delta r)\hat{\mathbf{r}} \tag{22}$$

Similarly from the top boundary, the particle is reflected with the same formula given above, with the Δr value which is calculated as: $\Delta r = -(r' - r_{max})$. The minus sign is used to take the direction of the coordinate system into account.

The particle's velocity is assumed to undergo a similar transformation. The particle stores the velocity vector of the previous half-time step. If we express the calculated velocity as:

$$\mathbf{v}'_{t+\Delta t/2} = \dot{r}\hat{\mathbf{r}} + r\dot{\theta}\hat{\theta} + \dot{z}\hat{\mathbf{z}} \tag{23}$$

Then the transformed velocity is:

$$\mathbf{v}_{t+\Delta t/2} = \mathbf{v}'_{t+\Delta t/2} - 2\dot{r}\hat{\mathbf{r}} \tag{24}$$

Reflection from radial, azimuthal and the axial left boundaries is performed during the relocation of each particle to its corresponding cell. There is a routine at each time step to find the cells that the particles are located in after the motion from the previous time step. During this relocation, the particles that cross the boundary are reflected back.

2.5 Reflection: Particles Crossing the Azimuthal Boundaries

Particles crossing the azimuthal boundaries ($\theta = 0$ and $\theta = \theta_{max}$) are handled assuming that for each particle that crosses the boundary, there is another one that enters the domain from the same location due to the symmetry condition.

The azimuthal reflection is depicted in Figure 5. The code implemented understands that a particle has crossed the boundaries by doing an *if-check* on the θ coordinate of each particle. If θ coordinate is negative or larger than θ_{max}, the reflection procedure is applied. The azimuthal coordinate of the particle after the position is updated with the velocity is denoted with $\theta'_{t+\Delta t/2}$. In the scope of this reflection, the following changes are performed on the azimuthal location and velocity of the macro particle:

If $\theta'_{t+\Delta t/2} < 0$:

$$\theta_{t+\Delta t/2} = \theta'_{t+\Delta t/2} + 2\Delta\theta \tag{25}$$

where $\Delta\theta = -\theta'_{t+\Delta t/2}$. For the reflection from the other boundary, if $\theta'_{t+\Delta t/2} > \theta_{max}$:

$$\theta_{t+\Delta t/2} = \theta_{end} - 2\Delta\theta \tag{26}$$

where $\Delta\theta = \theta'_{t+\Delta t/2} - \theta_{max}$.

The velocity component at both boundaries is updated as follows. If we express the calculated velocity as:

$$\mathbf{v}'_{t+\Delta t/2} = \dot{r}\hat{\mathbf{r}} + r\dot{\theta}\hat{\theta} + \dot{z}\hat{\mathbf{z}} \tag{27}$$

Then the transformed velocity is:

$$\mathbf{v}_{t+\Delta t/2} = \mathbf{v}'_{t+\Delta t/2} - 2r\dot{\theta}\hat{\theta} \tag{28}$$

3 Preliminary Results and Discussions

One of the domains investigated within the scope of this work is depicted in Figure 6. The purpose of the current study is to demonstrate the capabilities of the developed code such as particle tracking. The geometric configuration depicted in Figure 6 is held to be same at all simulation runs except for the location of the acceleration grid, thus the distance between the two grids.

The solution domain is divided into 62 x 62 x 10 nodes, which are in axial, radial and azimuthal directions, respectively. Poisson's equation is solved to evaluate the electric potential. The evaluated potential field is used to calculate the electric field on the computational cell nodes and these values are then weighted on to each macroparticle to determine the resulting electric force.

The solution domain considered for the discussed runs are adapted from the work of Farnell.[8] The modeled domain is 5mm in axial direction and 2m in radial direction. The screen and acceleration

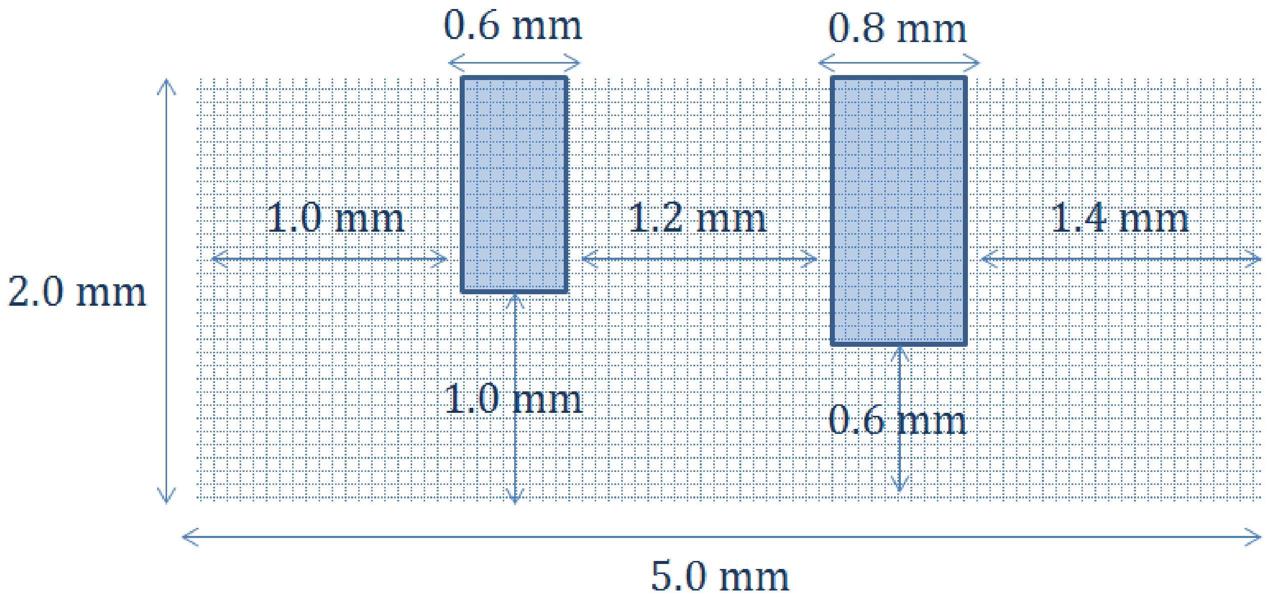

Figure 6: Dimensions of the solution domain for simulations

grids are set to 2241 V and -400 V, respectively. The potential in the plume region (right wall) is assumed to be at 0 V. The upstream discharge plasma potential is assumed to be 25 V above the screen grid potential. The radial and azimuthal boundaries have symmetry boundary conditions.

Each macro particle is marched in time and a loop over particles locates the cells where each particle is located in. Each cell object contains a neutral and an ion particle list. Cell objects also contain flags to indicate whether they correspond to screen or accelerator grids, or they are boundary cells at the edges of the computational domain in three directions. If a particle gets into these cells, it is reflected while retaining its momentum as depicted in Figure 5. For the presented work only ion and neutral particles are assumed and the particle initialization is implemented as discussed in Section 2.2.

A series of simulations are performed to investigate the particle trajectories and the flatness of the beam, and perveance for a particular grid separation.

Initially, when there are no ions in the system, the Gauss' Law reduces to the form of the Laplace's equation. The resulting electric potential field is depicted in Figure 7. This solution indicates the

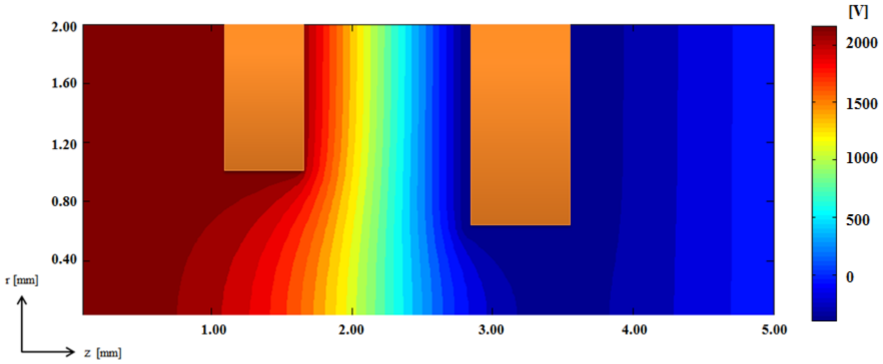

Figure 7: Electric potential fields in the solution domain at t=0 sec.

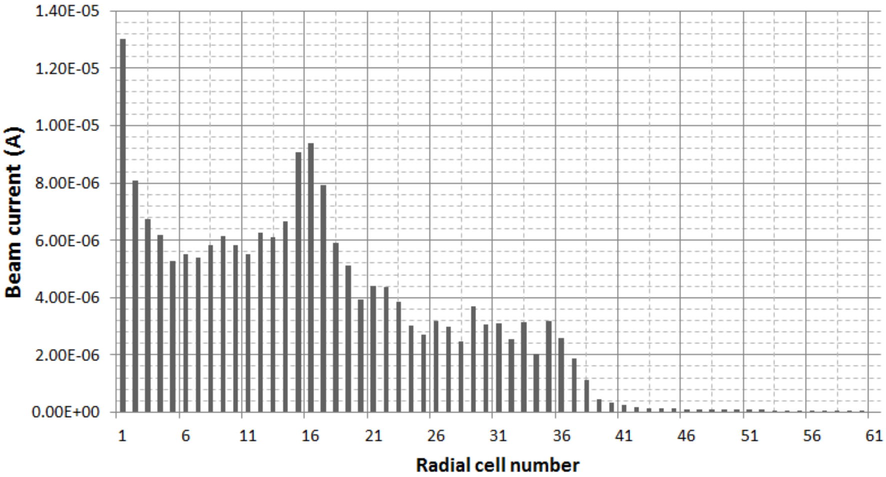

Figure 8: Distribution of the exit plane beam current for the discharge plasma ion number density: $n_i = 1.0e + 16$

existence of an almost uniform axial electric field between the screen and the accelerator grids.

An important parameter in the ion grid system design is the radial current distribution of the ions exiting the grid system. One of the goals is to keep the ions from hitting the grid walls and providing a low ion beam divergence. The ion beam current values through each computational cell at the exit plane (right boundary of the computational domain) are presented in Figure 8 for the bulk plasma density $n_i = 1.0e + 16$ m^{-3}. The desired profile for minimum beam divergence would have a small width around the hole centerline.

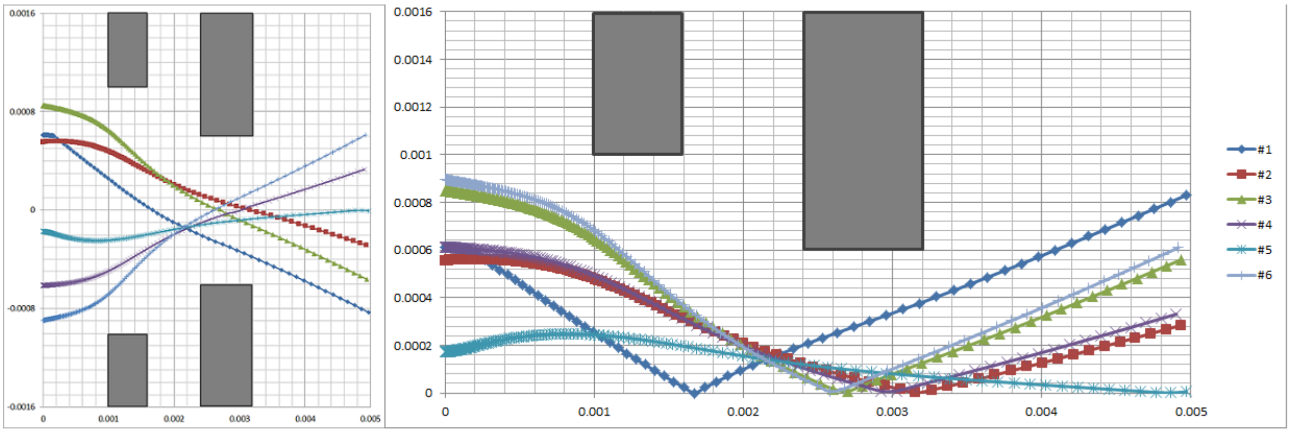

Figure 9: Trajectories of arbitrary ion macro particles obtained from the PIC-DSMC model (the trajectories shown on the left appears as centerline reflections as depicted in the figure on the on the right)

One of the utilities of the presented PIC-DSMC code is the ability to track particles along their short journey within the solution domain. Randomly selected particles are tracked by printing their radial and axial coordinates at each time-step. The azimuthal coordinates are not taken into account because of the axisymmetry assumption in azimuthal direction. Some of the obtained trajectories are depicted in Figure 9. As seen from the figure, use of the reflection at the centerline (r=0) boundary, the trajectories on the left figure can be represented as particles coming from the other side of the centerline on the right figure.

The flatness of these trajectories would be a desirable design objective as it means a smaller beam divergence and higher axial kinetic energy percentage. For example, particle #5 has velocity mostly in the axial direction whereas particles #1 and #7 have significant radial velocity components and would contribute to beam divergence angle to increase. Also, seen from the figure all particles, except for particle #5, has trajectories that crosses the center axis and show the examples of centerline reflections for the code as described in Section 2.4. Thus, to obtain an optimal configuration, the electric field topology and the geometry has to be adjusted to adjust the cumulative of the ion particle trajectories.

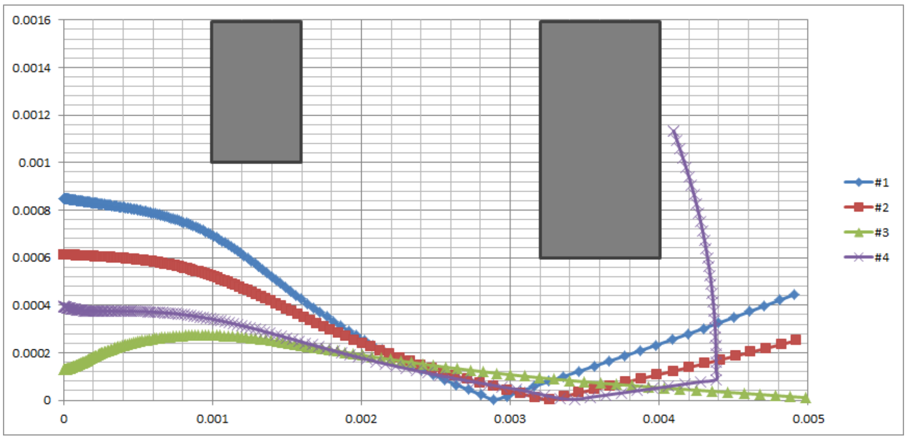

Figure 10: Trajectories of arbitrary ion macro particles obtained from the PIC-DSMC model with an alternative geometry configuration

As an alternative study, the geometry is changed slightly by moving the accelerator grid 0.6 mm to the right, thus making the grid separation distance 1.8 mm. Thus, different particle trajectories are obtained as few of which are shown in Figure 10. In this figure, the trajectory of the particle #4 shows that it undergoes an elastic collision with a neutral and as a result changes its trajectory and hits the front side of the acceleration grid. In general, it is unlikely that an ion undergoes such a large angle direction change due to an elastic collision with a neutral particle, however it is possible, and this trajectory is a suitable example of such a collision event. Additionally, particle #4 is also an example of ions becoming neutrals when they interact with the grid walls.

4 Conclusion

A simulation platform to be used in the design of ion thruster grid systems is developed. The motion of macro particles is handled using the cylindrical coordinates in three-dimensions. The solution domain is chosen so that it represents the smallest symmetric section of a two-grid system that is manufactured in a hexagonal pattern with circular concentric grid holes. The code is implemented using the C++ programming language and uses the object oriented programming extensively. DSMC collision algorithm is implemented to simulate the collisions between the heavy particle species.

The developed model will need additional capabilities such as improved handling of the electrons to simulate ion optics better as well as a sputtering model to increase its utility in ion thruster grid design. Nevertheless, the model is a valuable tool for design and analysis of ion thruster grids.

Acknowledgment

This work was supported in part by The Scientific and Technological Research Council of Turkey under Projects TUBITAK-112M862 and TUBITAK-113M244 and in part by Bogazici University Scientific

Research Projects Support Fund under Projects BAP-6184 and BAP-8442.

References

[1] B. Yavuz, E. Turkoz, M. Celik, in *6th International Conference on Recent Advances in Space Technologies (RAST)*, 619–624 (IEEE, 2013)

[2] E. Turkoz, M. Celik, in 49th *Joint Propulsion Conference* (San Jose, CA, 2013), also AIAA-2013-4110

[3] E. Turkoz, M. Celik, in 33rd *International Electric Propulsion Conference* (Washington DC, USA, 2013), also IEPC-2013-294

[4] E. Turkoz, M. Celik, Plasma Science, IEEE Transactions on **42**, 1, 235 (2014)

[5] E. Turkoz, F. Sik, M. Celik, in 50th *Joint Propulsion Conference* (Cleveland, OH, 2014), also AIAA-2014-3412

[6] J. R. Brophy, I. Katz, J. E. Polk, et al., AIAA Paper **4261**, 2002 (2002)

[7] J. R. Anderson, I. Katz, D. Goebel, *Numerical simulation of two-grid ion optics using a 3D code* (Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2004)

[8] C. Farnell, *Performance and Lifetime Simulation of Ion Thruster Optics*, Ph.D. Dissertation, Colorado State University, Fort Collins, CO, USA (2007)

[9] Y. Nakayama, P. J. Wilbur, in *27th International Electric Propulsion Conference* (Pasadena, CA, 2001), also IEPC-01-099

[10] M. Celik, M. M. Santi, S. Y. Cheng, et al., in 28th *International Electric Propulsion Conference* (Toulouse, France, 2003), also IEPC–03–134

[11] D. C. Rovang, P. Wilbur, Journal of Propulsion and Power **1**, 3, 172 (1985)

[12] R. Barrett, M. W. Berry, T. F. Chan, et al., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, vol. 43 (SIAM, 1994)

[13] L. Fausett, *Applied Numerical Analysis Using MATLAB, Second Edition* (Pearson Prentice Hall, 2008)

[14] Y. Saad, M. H. Schultz, SIAM Journal on Scientific and Statistical Computing **7**, 3, 856 (1986)

[15] J. Meijerink, H. A. van der Vorst, Mathematics of Computation **31**, 137, 148 (1977)

[16] J. M. Fox, *Advances in Fully-Kinetic PIC Simulations of a Near Vacuum Hall Thruster and Other Plasma Systems*, Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, USA (2007)