

Parallel codes using particles decomposition and view factor model methods for the particle in cell-Monte Carlo collision (PIC-MCC) simulation on cylinder hall thruster

IEPC-2019-473

[replace xxx above with the submission number you received in the acceptance message]

*Presented at the 36th International Electric Propulsion Conference
University of Vienna • Vienna, Austria
September 15-20, 2019*

Ruojian Pan¹, Zhuoheng Li², Shuai Cao³, Junxue Ren⁴

School of Astronautics, Beihang University, Beijing, 100083, China

Haibin Tang⁵

School of Space and Environment, Beihang University, Beijing, 100083, China

Abstract: Particle in cell-Monte Carlo collision method is accurate for electric propulsion simulation, but so time-consuming mainly because of two aspects. The first one is that the huge number of particles make the simulation dramatically slow in a serial way. The other is that the neutrals move much slower than other species, delaying the whole simulation into convergent. In this paper, focusing on these two problems, we develop a set of parallel codes and use a new method to simulate neutrals and simulate a cylinder hall thruster for testing. The parallel codes are based on the particle decomposition, which is suitable for relatively small scale plasma simulation. The highest speedup of the parallel PIC-MCC code comes to 16 compared with the serial one, showing good performance. On the other side, view factor model is combined into the serial PIC-MCC code. This method use view factor to calculate and update the 3D neutral distributions periodically, making the simulation both accurate and fast. The physical progress is more close to the real after combining view factor model with PIC-MCC code.

Nomenclature

A_s	=	area of the sphere
B	=	magnetic inductive strength
$\bar{C}$	=	mean neutral velocity based on the temperature
E_{inc}	=	electron incident energy
E	=	magnitude of electric field
F	=	the ratio of artificial heavy particles (ions and neutrals) to the real ones
$G_{a,b}$	=	geometry factor to define block between a and b
i_c	=	grid number on axial position
j_c	=	grid number on radial position
n	=	charge density in the EP chamber

¹ Master Candidate, School of Astronautics, prjbuaa@buaa.edu.cn.

² Bachelor, School of Astronautics, zhuohengli@foxmail.com.

³ Ph.D. Candidate, School of Astronautics, shuaicao@buaa.edu.cn.

⁴ Associate Professor, School of Astronautics, rjx_buaa@163.com.

⁵ Professor, School of Space and Environment, thb@buaa.edu.cn.

n_k	=	neutral density for inner nodes
q	=	charge accumulated on the ceramic surface
r_c	=	accurate position on radial position
$R_{a,b}$	=	radiation incident on surface b coming from surface a
ν_{ne}	=	neutral-electron collision frequency
ν_{tot}	=	total electron collision frequency
V	=	view factor
W	=	velocity of the particle
y_m	=	unit neutral flux from surface m
Y_m	=	neutral flux from surface m
z_c	=	accurate position on axial position
β	=	neutral loss rate
φ	=	electric potential
Ψ	=	geometry configuration
γ	=	the ratio of artificial permittivity to the real one
σ	=	electron-atom cross section
ς	=	transparency for neutrals
ρ	=	charge density on the BN ceramic

I. Introduction

Particle in cell-Monte Carlo (PIC-MCC) has been widely used in the electric propulsion (EP). PIC-MCC tracks every particle and treat all physical and chemical progress as the consequence of collisions, and gets the simulation result by the statistical laws. Based on the fewer physical assumptions, PIC-MCC can get accurate and time-advancement results, which makes the physical progress clearer to analyze. However, compared with the fluid and the hybrid particle method, the full particle method for electric propulsion (EP) simulation is accurate but time-consuming, which costs weeks or months.

There are mainly two reasons why it costs so much time to simulate EP thrusters by PIC-MCC. The first one is that tracking all kinds of species as particles would increase the computational load dramatically because the plasma density in the EP chamber would usually be at $10^{17} \sim 10^{20} \text{ m}^{-3}$. Three main approaches have been researched to accelerate the simulations, i.e., reducing the mass of heavy particles^{1,2}, increasing the space vacuum permittivity^{2,3} and the self-similarity model^{4,5}. To simulate different kinds of thrusters, including the Hall thrusters^{6,7,8}, the ion thrusters^{9,10,11}, and the magnetoplasmadynamic thrusters (MPDT)^{12,13}, one or more acceleration methods have been used. These methods have been proved to be effective to reduce the computational time but have some influences on the accuracy of the physical parameter. Increasing the space vacuum would cause a thicker sheath near the chamber walls. Furthermore, when reducing the mass of heavy particles, a relatively lower density distribution of plasma would appear in the chamber. To avoid or reduce the errors caused by the acceleration methods, parallel computing technology is used to speed up the simulation. DECYK *et al.* developed a general concurrent plasma particle-in-cell algorithm¹⁴, which could simulate various kinds of plasma. The algorithm of DECYK *et al.* used the domain decomposition and the uniform meshes, which was brought into the EP thruster's simulation¹⁵. Martino *et al.* developed a parallel method of PIC using particle decomposition instead of domain decomposition¹⁶ and it is more suitable for small scale simulation.

The second reason is that treating the neutrals like ions or electrons as particles could remarkably increase the simulation time. Previous researches have used direct simulation Monte Carlo (DSMC)^{17,18,19} to simulate atoms directly, or the original PIC-MCC^{20,21} method where atoms are equally tracked like ions and electrons. When computing with serial code, this type of full PIC method costs several weeks or months even if acceleration methods are used. The reason is that the neutrals move much slower than the ions and electrons which are pushed by the electric and magnetic fields. The slow speed causes slow convergence, delaying the entire simulation which gets convergent at the time when all the species come to stable quantities. To reduce the simulation time, researchers have adopted different approaches to calculate the neutrals. The neutral distribution change is much smaller and slower than the change of ions and electrons, so Mahalingam *et al.* applied a fixed-neutral model^{9,22} to avoid the high computation of the neutrals. In this method, neutral change is assumed to be neglected so that the neutral distribution has been set to be uniform in the discharge chamber or to be a constant background distribution simulated by a lower gas flow rate²³. Fixed-neutral model has great advantages on the computational speed and has been successfully applied on the PIC modeling of the ion thrusters²⁴, the Hall thrusters^{17,6}, and the hollow cathodes²⁵. It could make a fast simulation and provide a great extent of the thruster information. However, the changing of neutral distribution would undoubtedly

influence the physical process and simulation accuracy. Adam *et al.* established a fluid²⁶ method combining with PIC-MCC code to simulate the neutral distribution change. It is a 1D model, where neutral distribution only changes on the axial direction. The fluid model updates the new neutral distribution during every PIC iteration by solving a differential equation. All parameters in the equation (discharge frequency, atom speed, recombination speed) are determined by the PIC-MCC code. This model updates the neutral distribution by a hybrid way to improve the accuracy and has been used in Hall thruster simulation²⁷. However, a 1D model is still too simplified for the real physical condition.

A new method, view factor model, has been proposed in recent years. View factor model has already been used in the thermal radiation simulation and generalized into neutral distribution calculation due to its fast calculation speed. Compared to the MCC and DSMC method, view factor model only needs geometry factors and ionization parameters to solve the neutral distribution. Thus, the calculation cost is little and relatively constant because of the fixed calculation progress. Compared with the methods proposed by Mahalingam *et al.* and Adam *et al.*, view factor model could provide 3D neutral distribution, providing more detailed neutral information. Arako *et al.* used the view factor model in hybrid TURF code to analyze the sensitivity of ion flux to neutral density in the Hall thruster plume but did not take ionization into concern²⁸. Katz *et al.* established a whole fluid model combined with the view factor model to calculate the neutral atom distribution²⁹. Wirz in UCLA used the view factor model as a sub-model in their lab's code by receiving and sending messages from and to other sub-models to simulate the ion thrusters^{30,31}. The view factor model could propose a fast and accurate way to calculate neutrals, while PIC-MCC method has advantages in accuracy but a slow computational speed when simulating neutrals as particles. However, there has been little research concentrating on the application of view factor model on the PIC-MCC method, which might provide a both fast and accurate method to the EP simulation. In this paper, view factor model will be applied into the PIC-MCC code to simulate neutral changing for a cylinder hall thruster (CHT) in Joint Laboratory of Plasma and Propulsion (JLPP) in Beihang University.

This paper consists of following parts. In **Method and simulation** section, the basic information of PIC-MCC and Cylinder Hall Thruster (CHT) in JLPP would be introduced. The parallel algorithm and the application of view factor model on PIC-MCC is also stated in detail. In **Results and discussions** section, we set three cases, A, B and C, which all simulate the CHT but have different simulation settings. The serial (case A) and parallel codes (case B) are compared to find out the correction of the parallel algorithm, the speed-up and the bottleneck of the particle decomposition. The code combined with the view factor model (case C) are compared with that taking neutrals as a background distribution (case A). The difference of two cases on simulation results are described and analyzed in detail. In **Conclusion** section, the possible application of the parallel codes and the view factor model is discussed.

II. Method and simulation

A. Brief introduction of the PIC-MCC code in JLPP

The JLPP2.5 code is a complete model developed by Joint Laboratory of Plasma & Propulsion (JLPP) in Beihang University. Two modes are developed for simulation and there are no differences between these two methods except for the neutrals' treatment. One mode consists in treating atoms as particles, while the other fixes neutrals as a background distribution to achieve high-speed calculation. The code has been successfully used for several papers, including researching the magnetic mirror effect in Hall thruster⁶, the energy distribution on hollow cathode^{25,32}, the magnetic fields at the exit of magnetoplasmadynamic thruster (MPDT)¹³, and other physical effects in MPDT^{33,34}.

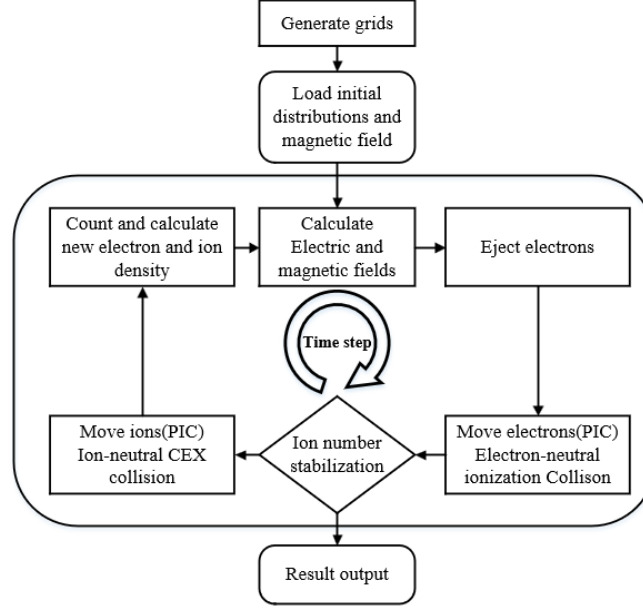


Figure 1. The scheme of the numerical algorithm of particle-in-cell and Monte Carlo collision.

JLPP2.5 is a 2.5D axisymmetric PIC-MCC system used for simulating EP. It divides the EP's axisymmetric plane into finite grids and updates the particle movements and electric field during each time step. There are four main procedures during each step.

(1) Using particle density parameters to solve the Poisson equation $\varepsilon_0 \nabla^2(-\phi) = e(n_i - n_e)$ and getting the newly updated electric field.

(2) Injecting the particles into the EP discharge chamber and calculating the movement driven by electric field E and magnetic field B.

(3) Using the MCC method to simulate the collisions between particles.

(4) Calculating the density of the new particles on the nodes. If the particle densities are convergent after iterations, we exit the loop and output the results. Schematic of PIC-MCC numerical algorithm is shown in Fig.1.

The collision types from the procedure (3) are referred to in Table 1.

Table 1. Different types in our PIC-MCC code, which contain the collisions between neutrals and electrons, ions (primary and secondary) and electrons, and collisions between ions and neutrals.

Particle Type	Collision Type
$\text{Xe} + e^-$	Elastic collision $\text{Xe} + e^- \rightarrow \text{Xe} + e^-$
	Excitation collision $\text{Xe} + e^- \rightarrow \text{Xe}^* + e^-$
	Ionization collision $\text{Xe} + e^- \rightarrow \text{Xe}^+ + 2e^-$
$\text{Xe}^+ + e^-$	Excitation collision $\text{Xe}^+ + e^- \rightarrow \text{Xe}^{++} + e^-$
	Ionization collision $\text{Xe}^+ + e^- \rightarrow \text{Xe}^{++} + 2e^-$
	Compound collision $\text{Xe}^+ + 2e^- \rightarrow \text{Xe} + e^-$
$\text{Xe}^{++} + e^-$	Excitation collision $\text{Xe}^{++} + e^- \rightarrow \text{Xe}^{+++} + e^-$
	Compound collision $\text{Xe}^{++} + 2e^- \rightarrow \text{Xe}^+ + e^-$
$\text{Xe}^+ + \text{Xe}$	Elastic collision $\text{Xe}^+ + \text{Xe} \rightarrow \text{Xe}^+ + \text{Xe}$
$\text{Xe}^{++} + \text{Xe}$ (Xe^+ for example)	Charge exchange collision $\text{Xe}^+ + \text{Xe} \rightarrow \text{Xe} + \text{Xe}^+$

For thrusters containing ceramic, the complex plasma-wall interaction is strongly related to the emission of secondary electrons. In our code, an improved SEE model for boron nitride (BN) material³⁵ is applied, which has shown reasonable agreement with the experimental result. Four events may occur when an electron collides with the BN wall: (1) the incident electron is absorbed by the wall; (2) the incident electron is reflected by the wall; (3) the incident electron knocks out a secondary electron; (4) the incident electron knocks out two secondary electrons. Eq.(1) gives the corresponding probabilities of the four events. Constant parameters' information would be stated in the thruster for simulation part.

$$\begin{cases} P_1(E_{\text{inc}}) = 0.5 \exp(E_{\text{inc}}^2 / X_0^2) \\ P_2(E_{\text{inc}}) = 0.5 \exp(E_{\text{inc}}^2 / X_1^2) \\ P_3(E_{\text{inc}}) = 1 - \exp(-E_{\text{inc}}^2 / X_2^2) \\ P_4(E_{\text{inc}}) = 1 - P_1(E_{\text{inc}}) - P_2(E_{\text{inc}}) - P_3(E_{\text{inc}}) \end{cases}, \quad (1)$$

where E_{inc} is the electron incident energy.

The cross-magnetic-field electron current to the anode is greater than could be accounted for through mere classical collisional diffusion processes in Hall thrusters. This open problem is generally attributed to the anomalous diffusion of electrons, also called the Bohm diffusion. A self-consistent method is introduced in order to quantify the Bohm diffusion coefficient by using the inherent information of the code and the least artificial parameters. According to Cappelli and Waltz's results^{36,37}, the anomalous transport in plasma devices seems to be controlled by the shear in the $\mathbf{E} \times \mathbf{B}$ drift velocity. A modified model given by Fox³⁸ has yielded promising results in conventional Hall thruster simulations. In³⁸, Bohm diffusion was incorporated into the neutral-electron collision frequency ν_{ne} . So the total electron collision frequency ν_{tot} can be given by:

$$\nu_{\text{tot}} = \begin{cases} \nu_{\text{ne}} + \left(1 - \frac{\gamma_{E \times B}}{\gamma_{\text{max}}}\right) \nu_{\text{Bohm}}, & \gamma_{E \times B} < \gamma_{\text{max}} \\ \nu_{\text{ne}}, & \gamma_{E \times B} > \gamma_{\text{max}} \end{cases} \quad (2)$$

where γ_{max} represents the maximum linear growth rate of the anomalous transport which is set as $\pi \times 1 \times 10^6 \text{ s}^{-1}$ and $\gamma_{E \times B}$ is the rate of shear in the $\mathbf{E} \times \mathbf{B}$ drift velocity. This velocity shear can be calculated from the axial derivate of the expected azimuthal drift velocity, or:

$$\gamma_{E \times B} = \frac{d}{dz} \left| \frac{\mathbf{E} \times \mathbf{B}}{|\mathbf{B}|} \right|, \quad (3)$$

Furthermore, in order to reduce the calculation, we adopt the Null Collision method to avoid searching all particles information in a time step¹⁸³⁹.

B. Cylinder Hall Thruster (CHT) for simulation

Compared with annular Hall thrusters (AHT), cylindrical Hall thrusters (CHT) are more advantageous in decreasing the erosion rate and raising the volume-surface ratio as well by shortening or cancelling the central magnet stem. CHT is small and light making itself consume less power and its untraditional magnetic topography results in a new mechanism of electron confinement and thruster operation. In JLPP, Beihang University, a low-power cylindrical Hall thruster in hundreds of watts is designed with special consideration in its magnetic circuit designing and simulation. The design drawing and the ignition of the thruster are shown in **Fig.2** (a) and (b). After completing the experiment system, the operation performances, discharge characteristics and plume diagnostics has been studied. The experimental results showed that at 100W level, the thrust, specific impulse and anode efficiency reached 4.48 mN, 1500 s and 32% respectively.

This thruster has also been applied to research the magnetic mirror effect on CHT performance using PIC-MCC code. In this paper, we use the PIC-MCC code based on the work of Jiang *et al.*⁶ and apply the parallelization and view factor model to the simulation. For these simulation cases, all the initial settings are the same. The computational domain and the potential boundaries are set as **Fig.3**, the charge density (ρ) on the BN ceramic depends on the charge in the chamber (n) and the charge accumulated on the ceramic surface (q).

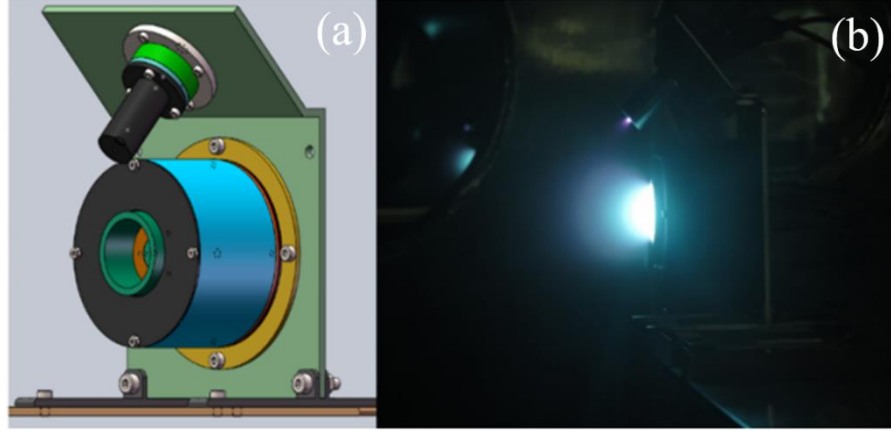


Figure 2. The designed structure (a) and ignition (b) of the CHT in JLPP, Beihang University.

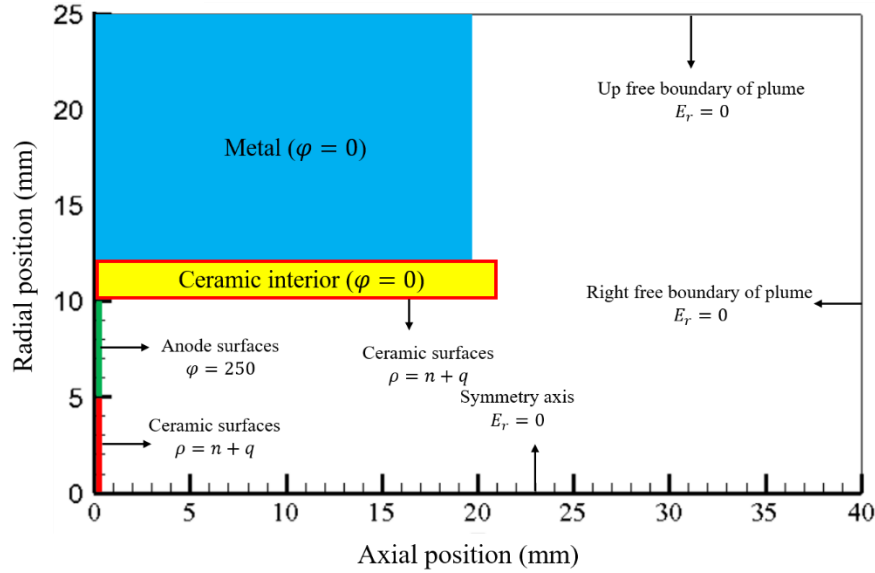


Figure 3. Simulation domain and boundaries settings for CHT in JLPP

For the SSE on BN ceramic surface in Eq.(1), constant parameters used in our code are set as follows: $X_0 = 43.5$, $X_1 = 30$ and $X_2 = 127.9^{35}$. For electrons emitted by the wall, their energy is picked randomly between 1–2eV as in ⁴⁰ and the velocity distribution function is assumed to be half-Maxwellian.

C. Parallel algorithm

In our serial code, the particles data are stored in the linked lists and the density information are stored in the two-dimension arrays. The electric field solving method is the Dynamic Alternating Direction Implicit (DADI) ²⁰ and the collisions are simulated by the Monte Carlo method. As information like velocities, positions and energies are calculated and irritated particle by particle, it is so time-consuming to tracking such large number of particles. For parallel coding to accelerate the simulation, message Passing Interface (MPI) is used in this paper.

For parallel PIC-MCC codes, mainly three parts of PIC-MCC module can be paralleled, including particle pushing module, electric field solving (DADI) module and MCC module. However, the importance of these three parts are not the same. The particle pushing module directly influenced the algorithm frame and the data structure of the whole PIC-MCC code so it is the most important. Electric solving module and collision module both rely on the grid-meshing method depending on the particle pushing module.

Two approaches has been researched for parallel particle pushing, domain composition ¹⁴ or particle decomposition¹⁶. Domain decomposition divides the simulation domain into different computational processes, where each process only stores particle information (electric potential, electric filed, particle densities etc.) in the specific

simulation domain. For domain decomposition, electric field is solved by all processes cooperatively and the particles are pushing in each process separately. When a particle passes across the domain boarder, communication happens between two processes, to delete the particle in the former process then add the particle into the new process. Domain decomposition method has the advantage of good extensibility, because the communication is one-by-one but not one-to-all. When the number of simulation particle increases, the communication increment is little and the meshing method can keep the same. So, this kind of decomposition is suitable for really-large scale plasma simulation. However, if the simulation domain is divided uniformly, particle numbers in each process would have huge amount difference as **Fig.4** (a). So, early researchers need to balance the particle number loach in each process by dynamic decomposition code like **Figure 4** (b)⁴¹.

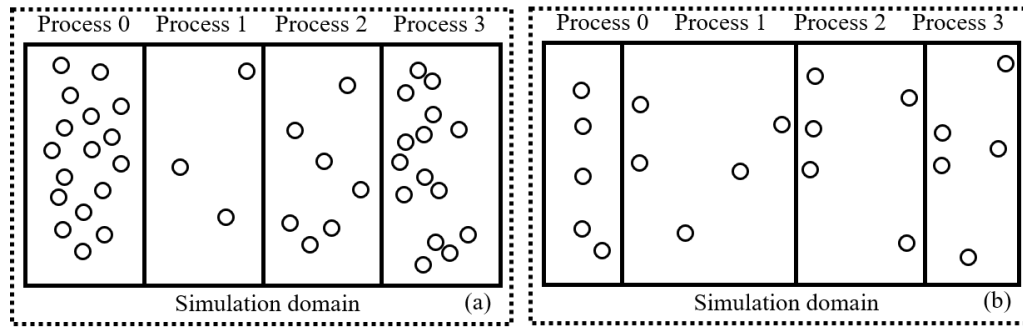


Figure 4. 1D uniform domain decomposition (a) and dynamic decomposition (b) for four process

The other approach is the particle decomposition. Different from the domain decomposition, the particles are handed out into different processes uniformly, which has natural particle number balance the processes. As **Fig.5** shown, the number of different colors of particles are the same, they are uniformly distributed to four processes and natural computational balanced. For each process in particle decomposition, it has its own linked lists and particle information (electric potential, electric filed, particle densities etc.). Each process solves the electric field of whole domain and simulates the particles of its own. Communication occurs when the particle densities, velocities put together before the electric field solving and the electric field information collects before the particle pushing module. However, the communication cost would increase dramatically as the simulation grids increase because the communication method is multi-to-multi. So, compared with the domain decomposition, particle decomposition is suitable for small scale simulation and easier to accomplish. As it is the first version of parallel code of our lab, we choose the particle decomposition approach to speed up our serial code.

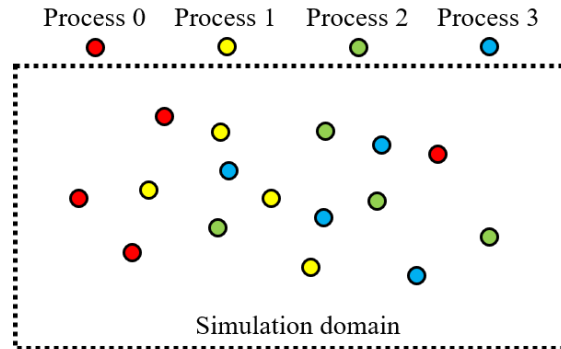


Figure 5. The particle decomposition parallel method

The whole simulation scheme is shown in **Fig.6**, where the blue parts include communication.

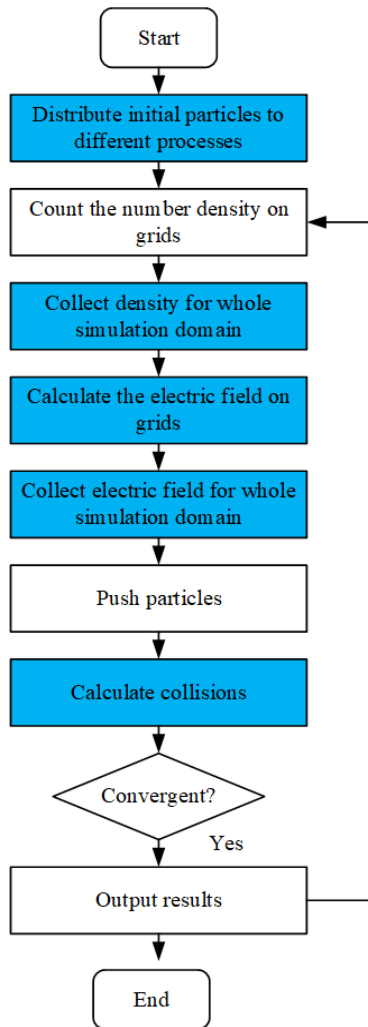


Figure 6. Parallel PIC-MCC scheme using particle decompositions

To make the MCC code paralleled, we concentrate two aspects. The collision in the serial code use three linked listed to solve the collisions like **Fig.7**.

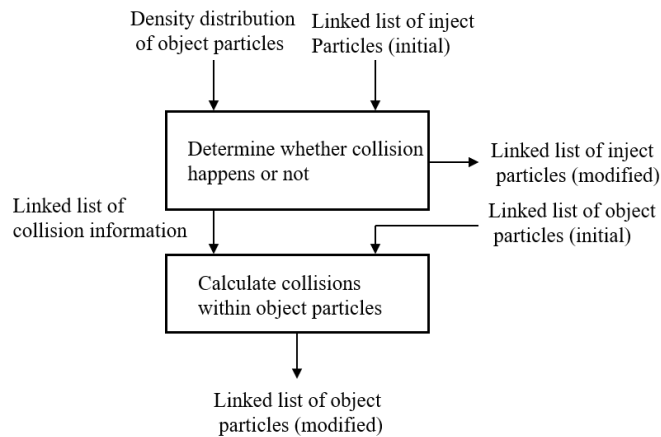


Figure 7. Collision simulating method

After simulating the injecting particles, we get a linked list containing collision information (position, velocity change etc.). When simulating the object particles, we need pick out the particles in the collision information list one

by one, and traverse the whole object particle list to find out a particle near the collision particle. It is time-consuming and hard to be paralleled. Thus, collision arrays are set instead of collision list. The values of kinetic energy changing are dispersed into different energy level, each is contained in a collision array as **Fig.8**. The arrays' size (line $\times$ column) equals to the grids number of simulation domain. The elements in these arrays represent the collision times on that grid.

Linked list of collision information

List head
Collision 1 $z_{c1}, r_{c1}, \Delta\epsilon_{inc1}$
Collision 2 $z_{c2}, r_{c2}, \Delta\epsilon_{inc2}$
Collision 3 $z_{c3}, r_{c3}, \Delta\epsilon_{inc3}$
...
...
...

Arrays of collision information

$\Delta\varepsilon_{inc} \in (0, 1]eV$	$\Delta\varepsilon_{inc} \in (1, 2]eV$		$\Delta\varepsilon_{inc} \in (30, +\infty)eV$																											
j_{c1}	j_{c2}		j_{c3}																											
<table border="1"> <tr><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	1	0	0	0	0	0	0	<table border="1"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> </table>	0	0	0	0	0	0	0	1	0		<table border="1"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	1	0	0	0	0	0
0	0	1																												
0	0	0																												
0	0	0																												
0	0	0																												
0	0	0																												
0	1	0																												
0	0	0																												
1	0	0																												
0	0	0																												
i_{c1}	i_{c2}		i_{c3}																											

Figure 8. The relationship between linked list and arrays of collision information

where $\Delta\epsilon_{inc} = \epsilon_{inc} - \epsilon_{scat}$ is the value of kinetic energy changing, (z_c, r_c) is the accurate position of the particle, (i_c, j_c) is the line number and column number of the collision position.

Thus, we can modify the object particles simulation into following:

- (1) Pick out an object particle from the linked list, find out the position grid (i_c, j_c)
- (2) Search collision particles on (i_c, j_c) in all the collision information arrays. If the element value of the collision arrays is not 0 on (i_c, j_c) , reduce 1 for this element then update the corresponding object particle.
- (3) Repeat (1) and (2) until the whole object particle list is traversed.

Thus, the object particle list is traversed only once, and the whole progress can be paralleled. For particle decomposition codes, we set a main process (usually process 0) to control all other processes whether to calculate collision or not. Assume there is M energy levels and P processes, which are mark as rank0 rank1...rank(P-1). Then the paralleled scheme is shown as **Fig.9** (three ranks for example).

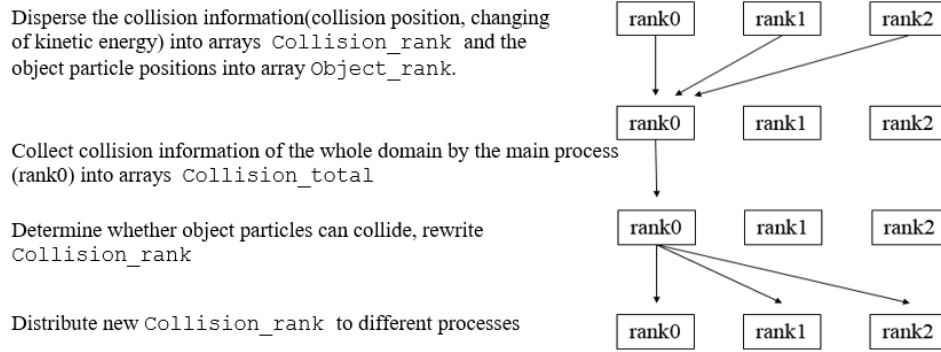


Figure 9. Parallel DADI algorithm

In our serial code, we use five-point difference format as **Fig.10**²⁰ to get potential expression of point P. The we DADI, which equals to solve several tridiagonal equations as Eq.(4) and (5), to solve the electric potential.

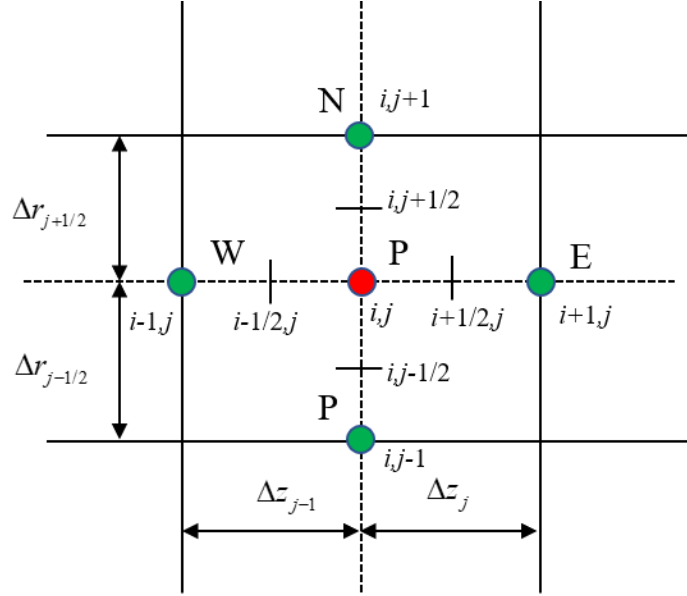


Figure 10. Five-point difference format

$$a_{E,i} \phi_{i+1,j}^{k+1/2} + a_{P,i} \phi_{i,j}^{k+1/2} + a_{W,i} \phi_{i-1,j}^{k+1/2} = RHS_{i,j}^k \quad (4)$$

$$a_{N,j} \phi_{i,j+1}^{k+1} + a_{P,j} \phi_{i,j}^{k+1} + a_{S,j} \phi_{i,j-1}^{k+1} = RHS_{i,j}^{k+1/2}, \quad (5)$$

where $a_{E,i}$ $a_{P,i}$ $a_{W,i}$ $a_{N,i}$ $a_{P,i}$ $a_{S,i}$ are related to the geometry division and point positions, which has been described in detail in Ref²⁰. Typical tridiagonal equations can be written as Eq(6), where $a_1=c_n=0$.

$$\begin{bmatrix} b_1 & c_1 & & 0 \\ a_2 & b_2 & c_2 & \\ & a_3 & b_3 & \cdot \\ & & \cdot & \cdot & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \cdot \\ \cdot \\ \phi_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ \cdot \\ \cdot \\ d_n \end{bmatrix}, \quad (6)$$

In practical using, we often use Thomas method⁴², which is the fastest to solve tridiagonal equations and the time complexity is only $O(n)$. Eq.(6) can be transferred into Eq.(7) (elimination process):

$$\begin{bmatrix} 1 & u_1 & & 0 \\ & 1 & u_2 & \\ & & 1 & \cdot \\ & & \cdot & \cdot & u_{n-1} \\ 0 & & & & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \cdot \\ \cdot \\ \phi_n \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_n \end{bmatrix} \quad (7)$$

where

$$\begin{cases} u_1 = c_1 / b_1, & y_1 = d_1 / b_1 \\ u_i = c_i / (b_i - u_{i-1}a_i), & i = 2, 3, \dots, n-1 \\ y_i = (d_i - y_{i-1}a_i) / (b_i - u_{i-1}a_i) \end{cases} \quad (8)$$

Then the back substitution progress is shown as Eq.(9):

$$\begin{cases} \varphi_n = y_n \\ \varphi_i = y_i - u_i x_{i+1} \end{cases} \quad (9)$$

As the Thomas method is the fastest algorithm to solve tridiagonal equations, we focused on the parallelization on it and chose the ADI Algorithm on a Ring of Parallel Processors⁴³ to solve the electric field. We distribute the equation set as **Fig.11**, where number 1,2,3 represent different processes.

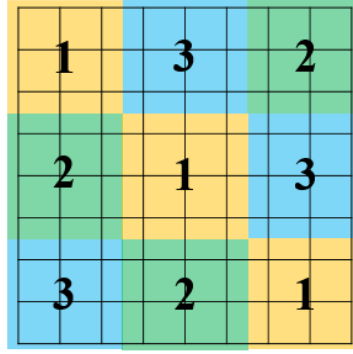


Figure 11. Equation distribution method of ADI Algorithm on a Ring of Parallel Processors

Assume there is P processes, for each line and column of vector in DADI, it needs all P processes to solve one ADI iteration. The communication time of each line or column iteration is always $P-1$, because there is no need to add communicate with other processes when transfer from line iteration to the column iteration. As the local data is enough to solve the equations in the very process, it is suitable for both shared and distributed memory systems. The algorithm flow chat is shown in **Fig.12**.

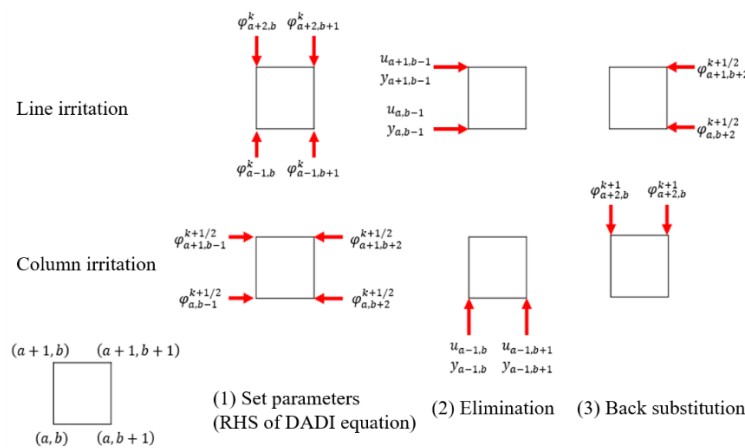


Figure 12. The data transfer for one line irritation and column irritation

Assume the computational domain is $N \times N$, the computational cost for a complete DADI irritation using P processes is listed as **Table.2**. Addition or subtraction times are ignored as the computational cost of multiplication or division is much higher.

Table 2. Computational cost for one ADI step irritation

Progress	Computational cost	Communication times
Elimination	$4N^2 \times 2$	$N(P-1)/P \times 2$
Back substitution	$N^2 \times 2$	$N(P-1)/P \times 2$

D. View factor model

View factor is a geometry factor and a concept in heat transfer theory, which has been frequently used in thermal radiation calculation⁴⁴⁴⁵. For two diffusing faces a and b which are isothermal and physical-properties uniform, the radiation incident on surface b coming from surface a is defined as $R_{a,b}$, while the total effective radiation of surface a is defined as $R_{a,all}$. Thus, the definition of thermal view factor Ψ_{ab} is

$$\Psi_{ab} = \frac{R_{a,b}}{R_{a,all}} \quad (10)$$

For two finite surfaces (**Fig.13**), $\Psi_{A_1A_2}$ is calculated in the integral form as Eq(11), where dA refers to the differential parts of a surface A and n refers to the normal vector of each differential surfaces.

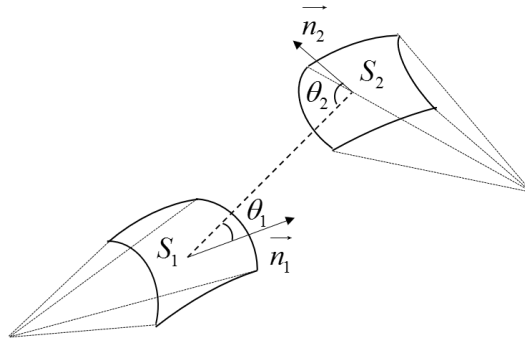


Figure 13. Definition of view factor

$$\Psi_{A_1A_2} = \int_{A_1} \int_{A_2} \frac{\cos \varphi_1 \cos \varphi_2}{\pi r^2} dA_1 dA_2 \quad (11)$$

Two types of view factor need to be calculated³¹. First, for a differential surface dA_1 and a finite surface A_2 , which is used to determine the factor between two wall elements. $\Psi_{dA_1A_2}$ is calculated in Eq.s(12):

$$\Psi_{dA_1A_2} = \frac{1}{\pi} \int_{A_2} \cos(\theta) dA_s = \frac{A_s}{\pi} \cos(\theta) \quad (12)$$

Second, for the view factor from a differential (or finite) surface to a point, which is used to determine the factor of wall elements to inner nodes. Ψ_{AO} is calculated in Eq.(13)-(14).

$$\Psi_{AO} = \frac{A_s}{A_{sphere}} = \frac{A_s}{4\pi} \quad (13)$$

$$A_s = \gamma_1 + \gamma_2 + \gamma_3 - \pi \quad (14)$$

For a typical EP thruster where plasma density varies from $10^{17} \sim 10^{20} \text{m}^{-3}$, the mean free path of a neutral atom for interatomic and charge-exchange collisions is sufficiently long to warrant a collision-free approximation⁴⁶. View factor can be generalized to determine the steady neutral atom flux balance⁴⁷, which makes possible feasibility to apply on the PIC-MCC simulation on the EP thrusters. To apply the view factor model on EP simulation, the assumptions are concluded into the following statements.

- (1) Neutral atoms are added by the gas source (including actual gas sources and the chamber-walls reemitting) and the recombination of ions and electrons, while they are lost only by ionization.
- (2) The structure of the EP chamber and real gas sources are axisymmetric.
- (3) The temperature-based speed of neutral atoms remains constant due to a uniform wall temperature assumption.
- (4) Compared with the recombination on the chamber wall, the recombination in the chamber is ignored.

To determine the neutral atom lost, we generalize a new view factor V , which is a product of the thermal view factor Ψ , the geometry factor G and the neutral ionization fraction β as Eqs.(15)-(17).

$$V = G\beta\Psi \quad (15)$$

$$V_{mh} = G_{mh}\beta_{mh}\Psi_{mh} \quad (16)$$

$$G_{mh} = \begin{cases} 0 & \text{(blocked)} \\ 1 & \text{(unblocked)} \end{cases} \quad (17)$$

where G_{mh} is used to evaluate whether there is a block between two surfaces m and h . β_{mh} is the ratio of the final and initial neutral gas flow. According to assumption(3), β is defined as the ratio between the final neutral density (n_o^f) and initial neutral density (n_o^i) as Eq.(18), as a parameter to determine the ionization rate in the EP chamber.

$$\beta \equiv \frac{n_o^f}{n_o^i} \quad (18)$$

The neutrals are treated as particles passing through a field of electrons, where the local ionization rate is related to the primary electron ionization rates K_{iz}^p and secondary electron ionization rates K_{iz}^s as Eq.(19). β is calculated by integrating the ionization parameters along the neutral flow path.

$$\dot{n}_i = n_o n_e K_{iz} \quad (19)$$

where n_o , n_p , n_s represent the density of neutrals, primary electrons and secondary electrons respectively. Neutral loss rate $\dot{n}_o$ is the negative of ion generation rate $\dot{n}_i$. So by solving the differential equation, β between two surfaces can be integrated as Eqs.(20)-(21).

$$n_o^f = n_o^i \exp\left[-\int (n_e K_{iz}) dt\right] \quad (20)$$

$$\beta = \exp\left[-\int (n_e K_{iz}) dt\right] \quad (21)$$

We design a view factor model algorithm which is suitable for our PIC-MCC code to calculate and update the neutral distribution in the cylinder EP simulation domain. For coupling the PIC-MCC code and the view factor model, we divide the PIC-MCC simulation regime to fit the view factor model. View factor model and PIC-MCC algorithm are respectively used to simulate the neutrals and the other particles (ions and electrons). While the neutrals change much slower than the charged particles, it could be assumed quasi-steady and be updated periodically. Thus, the PIC-MCC code provides the ionization information, including electrons energy and density distribution, for the neutral calculation, and the view factor model updates the background neutrals for the particle simulation. Finally, the combined simulation would converge when all the distributions of neutrals, ions and electrons keep stable. The coupling model is described in detail as follows.

Firstly, the EP chamber is divided into surfaces so that the view factors and ionization parameters could be calculated. Surfaces are divided into 1D wall elements, then the inner nodes are generated on the 2D solution plane as **Fig.14** (a). After 1D segmentation, the 1D elements are divided into 2D triangles as **Fig.14** (b). The EP chamber is firstly divided into 1D elements, each of which could present a ring of the chamber wall. The 2D solution plane is selected to do the simulation and calculate physical parameters. Inner nodes (yellow points) are generated on the plane at the cross points of the 1D element division lines. Then 2D triangles are used to approximate all the 1D wall elements. The green points are the geometric center of each triangle.

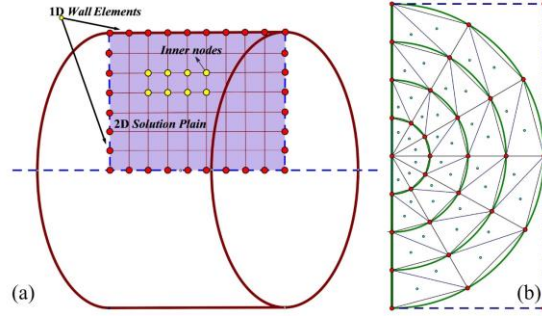


Figure 14. (a) 1D and 2D division for the simulation plane (b) 2D triangles approximation for 1D wall elements

Secondly, a matrix representing the interrelationship of the wall elements to determine the neutral flow balance is established. By continuity, the summation of the thermal view factors of an element is 1. For a given wall element, h , the thermal view factors are normalized using an identical factor to satisfy the neutral-flux continuity as Eq.(22).

$$\sum_m \Psi_{hm} = \sum_m \sum_{t_m} \Psi_{ht_m} \equiv 1 \quad (22)$$

where t_m refers to the triangle elements of 1D element m .

With all the view factors between wall elements defined, an expression for the neutral flow continuity between the wall elements is determined. For steady-state continuity, when the combination effect is ignored, the neutral flow (Y) of a given wall element (h) must equal to the sum flow from all other wall elements (m), gas feed from the wall (s_w) as Eq.(23)

$$Y_h = \sum_m V_{mh} Y_m (1 - \zeta_{0,m}) + \sum_{s_w} V_{s_w h} Y_{s_w} + Y_{rh} \quad (23)$$

where $\zeta_{0,m}$ is the transparency of wall element, m .

Y_h, Y_m, Y_{sh} represent the neutral flow from wall element h, m and the real gas source, respectively. Y_{rh} is the recombination part which occurs on wall elements h . Eq.(23) is suitable for all wall elements so we could determine the neutral flux of the whole EP surfaces by an invertible matrix(24).

$$\begin{bmatrix} 1 & (-V_{21}(1-\zeta_{0,2})) & \cdot & \cdot \\ (-V_{12}(1-\zeta_{0,2})) & 1 & (-V_{32}(1-\zeta_{0,2})) & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} \sum_{s_w} V_{s_w 1} Y_{s_w} \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \quad (24)$$

Thirdly, all neutral view factors, V , would be calculated to make the information of matrix complete. For the neutral view factor (V), β is significantly important to be calculated. According to Eq(12), K_{iz} need to be determined where K is the ionization rates of electrons and could be calculated as Eq.(25).

$$K = \int_0^\infty w f(w) \sigma(w) dw \quad (25)$$

where w is the velocity of particles, $f(w)$ is the distribution function of w , and $\sigma(w)$ is the function of collision cross section. K is calculated in a statistic form⁴⁸ as Eq.(26) and averaged every N steps in Eq.(27), where all parameters could be obtained from the PIC-MCC code.

$$K = \sigma \left(\frac{1}{2} m \langle w \rangle^2 \right) \langle w \rangle \quad (26)$$

$$K = \sum_{i=1}^{5000} \sigma_i w_i / N \quad (27)$$

As we assume a constant velocity of neutrals, β can be dispersed into Eq.(28)

$$\beta = \sum_n (K_{iz} n_s \cdot dt)_n \quad (28)$$

In Eq(28), the neutral integral path is divided into n equal divisions, so that calculation time for each division is dt . PIC-MCC method is used to obtain the electron density (n_s, n_p) from each grid, and to calculate the electrons velocity, w , and electron-atom collision cross section, $\sigma(w)$, in Eq.(16).

The 3D integral path is projected into the 2D solution plane using a solid ring method. As the integral path is divided into n equal divisions, we use the area allocation method to get the in-cell dispersed points from related grids.

Distinguishing whether there is any block between two specific differential surfaces, geometry factor G is determined to 0 (blocked) or 1(unblocked). Thus, all the parameters and every $V_{mh} = G_{mh} \beta_{mh} \Psi_{mh}$ are calculated, the neutral densities on each grid could be determined. y_m is defined as the unitary neutral flow on wall elements m , as Eq.(29).

$$y_m = Y_m / A_m \quad (29)$$

The density from the wall and the real gas source both contribute to the density of an inner node k as Eq.(30). $\bar{C}$ refers to the mean speed of the neutrals from the wall or the real gas source, which is determined by the local temperature as Eq.(31).

$$n_k = 4 \sum_m V_{mk} (1 - \zeta_{o,m}) \left(\frac{y_m}{\bar{C}_m} \right) \quad (30)$$

$$\bar{C} = \sqrt{\frac{8kT}{\pi m}} \quad (31)$$

To determine the density at a wall element, h , we use both the incident and outgoing flux for the unit hemisphere surrounding the element. The resulting expression for the neutral density of a wall element (n_h) can be determined as Eq.(32)

$$n_h = 2 \frac{y_h}{\bar{C}_h} + 2 \sum_m V_{mh} (1 - \zeta_{o,m}) \frac{y_m}{\bar{C}_m} \quad (32)$$

The entire process of view factor model algorithm combined with PIC-MCC method is explained in **Fig.15**. Updating neutral distribution by view factor model is applied every N time steps during the PIC-MCC method. Before calculating the neutrals, the EP chamber is divided into 1D elements and then into 2D triangles. The averaged

ionization parameters, which determine the parameter K , are from PIC-MCC method. Thus, the neutral ionization fraction for elements m and h (β_{mh}), could be integrated on the neutral moving path by parameter K and electron densities (n_s and n_p). Then Ψ_{mh} , V_{mh} and the neutral flux matrix could be calculated and established. By solving the matrix, the neutral flux for each element is determined. As the input parameters for the next-step PIC-MCC iteration, the new neutral distribution is calculated from the flux data. Only when the ions, electrons and neutrals all come to convergent, the iterations of this algorithm would be terminated.

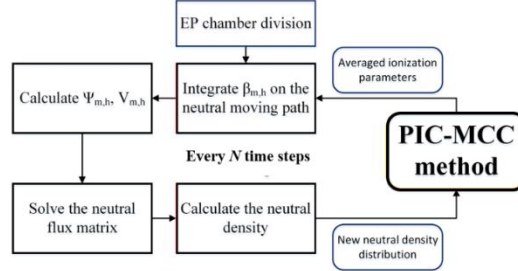


Figure 15. Algorithm scheme of view factor combining with PIC-MCC code

III. Results and discussions

We set three simulation cases, A, B and C, to simulate on CHT. Besides the differences listed in **Table.3**, all the settings are all the same for three cases. The anode neutral flow for three cases are all 3sccm. We used moderate acceleration method, only adopt an artificial permittivity $\gamma^2 = 20^2$, while no heavy particle mass reduction or self-similarity method are used. For the macro particle, we set 10^7 (macro weight) real particles in one macro one.

Table 3. Cases settings for A, B and C

CHT in JLPP, Beihang University	Serial or parallel	Treatment for the neutrals
Case A	Serial	Background distribution
Case B	Parallel	Background distribution
Case C	Serial	Udate per 10000 steps using view factor model

Case A and B are used to test the parallel accuracy and efficiency to the serial codes. Case A and C are to verify the validity and influence of the view factor model on the CHT simulation. The simulation parameters are listed in **Table.4**.

Table 4. Simulation parameters for serial (case A) and parallel (case B) codes

Parameters	Values
Error limits for DADI	10^{-12} for DADI model only, 10^{-3} for complete PIC codes
Grids width	2×10^{-4} m
Grids number	200×125
Time step	5×10^{-12} s
Frequency of solving electric field	1 time/10 time steps
Frequency of particle pushing	Electron: 1 time/1 time step Ion: 1 time/10 time steps
Frequency of collision calculating	1 time/1 time step
Compiler / environment	gcc -O2 on Ubuntu 18.04

A. The accuracy and efficiency of the paralleled code

(I) Accuracy validation

A presents the serial PIC without view factor model, B is the parallel PIC without view factor model, while C is the serial PIC with view factor model. The electric field solving module is tested independently first on the cloud machine, which information is listed in **Table.5**.

Table 5. The machine parameters of validation test for case A and B

Item	Value
Company	Ali Cloud
CPU	Intel Xeon Platinum 8163
Cores of CPU	12(virtual cores)
Memory	24GB

The electric field without charges solved by serial code is shown in **Fig.16**. We set four processes for parallel DADI solving, the results for each separated process (rank0-3) are shown in **Fig.17** and the total electric field in **Fig.18**.

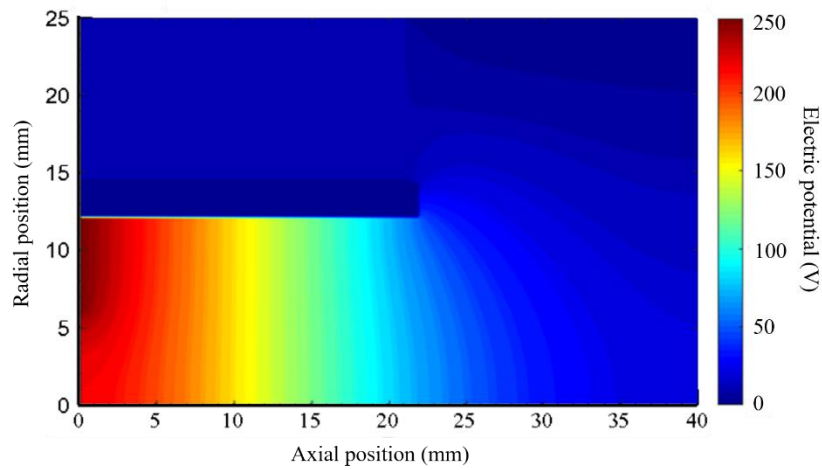


Figure 16. Electric potential module simulated by serial code

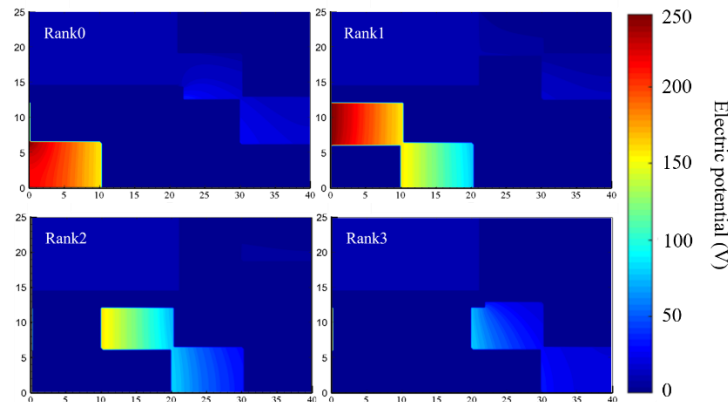


Figure 17. Electric potential module simulated by four independent processes (parallel code)

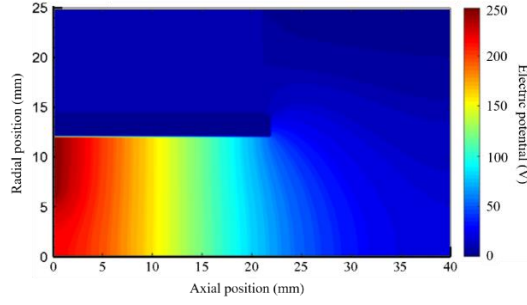


Figure 18. Electric potential module simulated by parallel code (sum of four processes)

The distribution of difference between serial and parallel code $\Delta\varphi = |\varphi_{parallel} - \varphi_{serial}|$ (**Fig.19**) illustrates that the accuracy of parallel code for DADI is high as the error is quite close to the error limit 10^{-12} V.

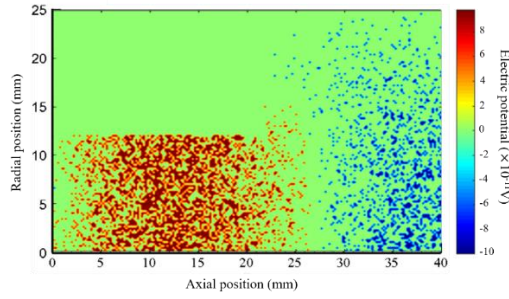


Figure 19. The potential difference of serial and parallel code

We then set eight processes on the cloud machine to run the complete PIC-MCC parallel code. For the steady state of both simulations, the comparisons of electric potential (**Fig.20**), densities for electrons (**Fig.21**), single-charged ions (**Fig.22**) and double-charged ions (**Fig.23**) appear little differences, which verify the accuracy of the parallel algorithm.

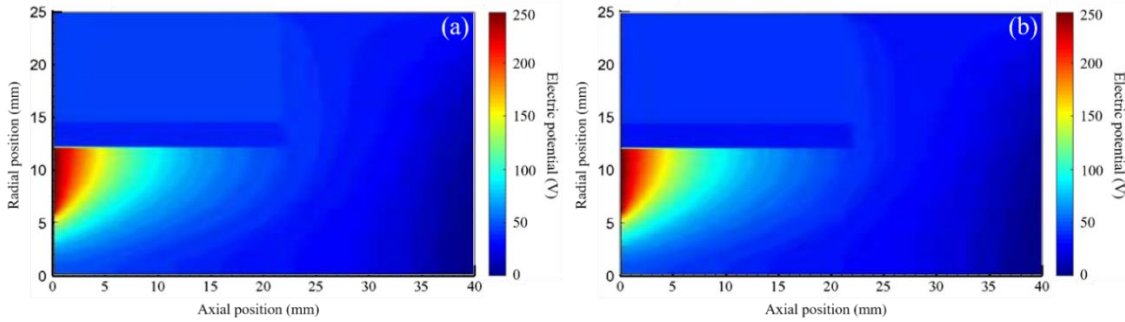


Figure 20. (a) Electric potential for serial (case A) and (b) parallel code (case B)

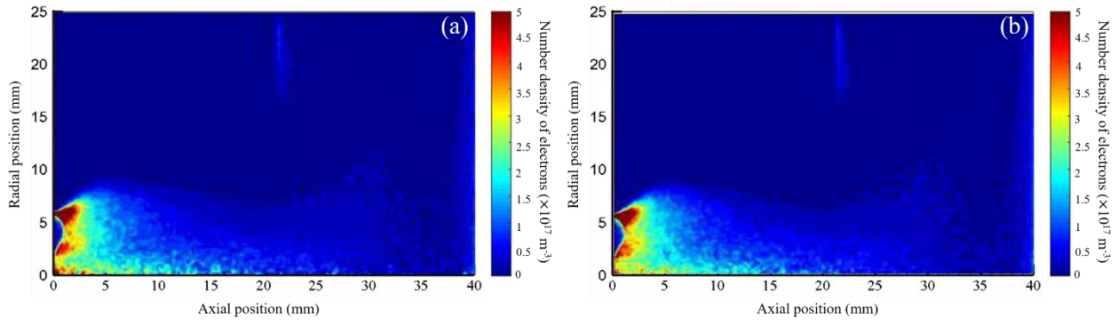


Figure 21. (a) Electrons density for serial (case A) and (b) parallel code (case B)

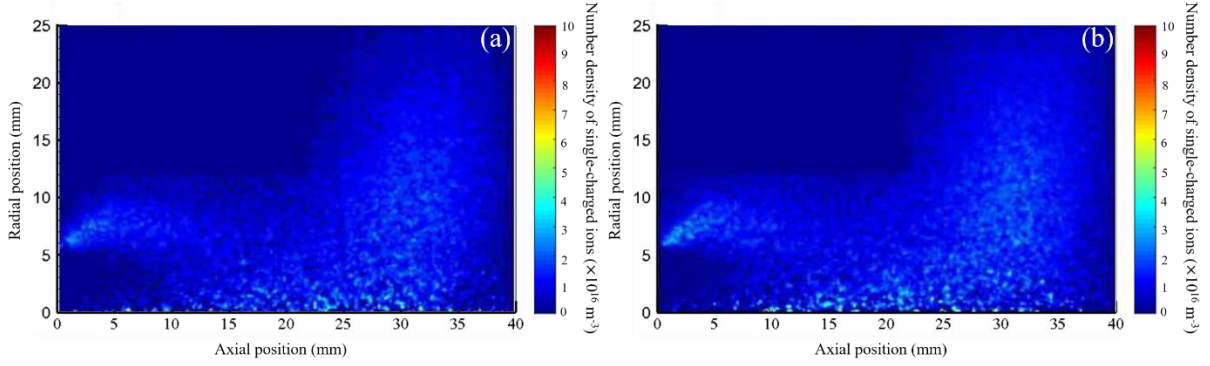


Figure 22. (a) Single-charged ions density for serial (case A) and (b) parallel code (case B)

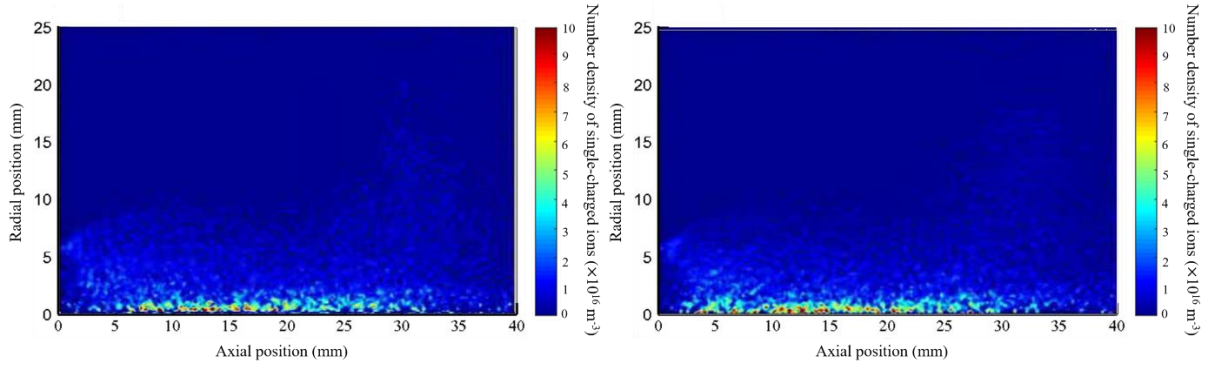


Figure 23. (a) Double-charged ions density for serial (case A) and (b) parallel code (case B)

The convergent curve of serial and parallel code (**Fig.24**) shows that the whole physical progress is also almost the same.

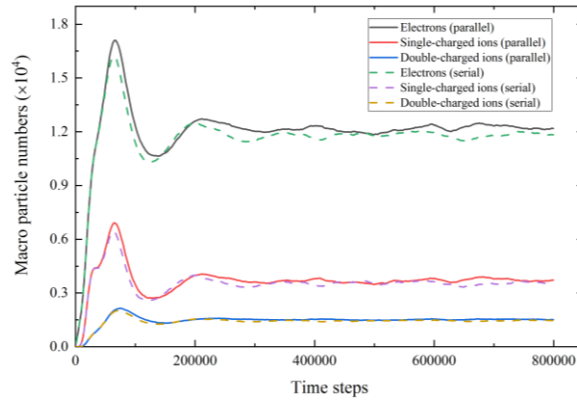


Figure 24. Convergent curve of for serial (case A) and parallel code(case B)

For this case, serial codes reach 103,000 time steps when the parallel codes reach 800,000 time steps. The speedup of 8 cores parallel codes is 7.8, showing good parallel performance.

(II) Performance limit

To test the performance limit of parallel codes, we used the physical machine of Amazon Web Service (AWS), which has less performance loss due to the virtualization. Its information is listed in the **Table.6**.

Table 6. The information of ASW machine

Item	Value
Cloud machine	AWS c5.18xlarge instance
CPU	Intel Xeon Platinum 8124M

Cores of CPU	36(physical core)
Memory	144GB

We runned our simulation on ASW for different process numbers (**Fig.25**). The particle numbers for each process numbers are almost the same, which further improves the validity of the parallel codes.

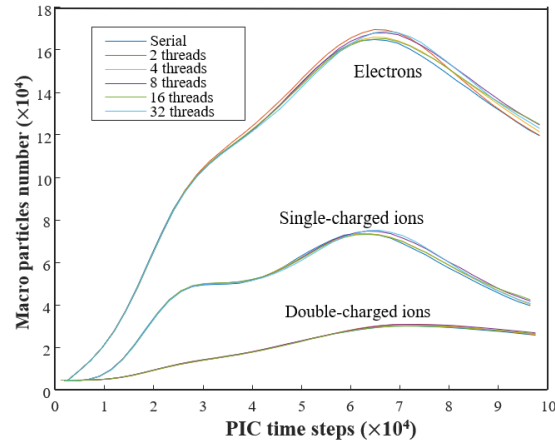


Figure 25. The macro numbers for electons and ions (from serail to 32 processed parallel code)

The speedup per 100 PIC time steps shows the parallel performance is quite good under 16 cores, the maximum speedup appears at the 32 processes case. However, for 32 cores case, the unit speedup(=actual speedup/cores) is small, as the communication time increases with the core numbers (**Fig.26**) and the difficult DADI penalization.

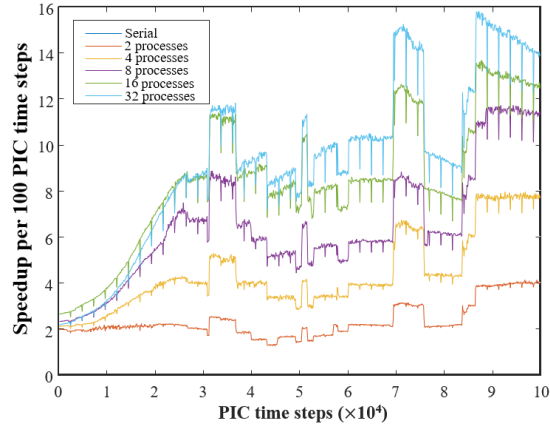


Figure 26. Speedup for different processes

The time increment caused by communication with the increasing cores is acceptable as (**Fig.27**). The time fluctuation at 2 processes maybe is caused by the machine.

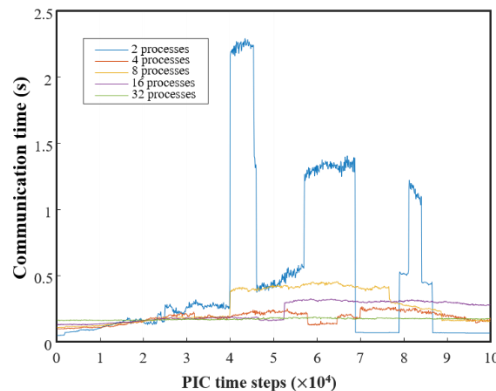


Figure 27. Communication time for different processes

Although more cores have been applied into the parallel code, the maximum speedup is still not high enough. That is because the existing code is sensitive to the grids numbers, and more importantly, the DADI solving time, which is the slowest part in the whole parallel procedure. In conclusion, in our existing parallel code, the work until now developed a set of particle decomposition parallel code, which is suitable for small scale EP thruster due to the communication consuming. The maximum speedup is about 16 and the most efficient and most cost-effective scheme is to choose a 16 process instance to do parallel simulation.

B. The simulation difference when using view factor model

For convenience, we slightly changed the geometry structure of the simulation domain, as **Fig.28**, ignored the protrusion of the BN ceramic. The red boundaries represent EP walls and the blue ones are the free boundaries where neutrals can pass freely. we set the transparency as 0 and 1 for EP walls and free boundaries.

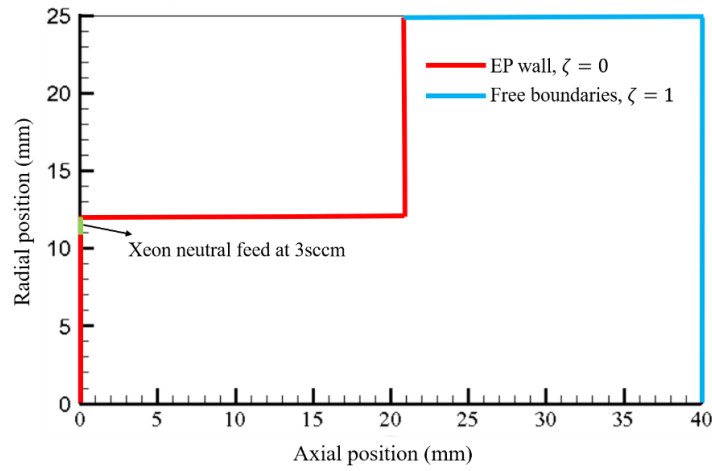


Figure 28. Simulation settings for view factor model

A 3sccm simulation without ionization is compared with view factor model and COMSOL software.

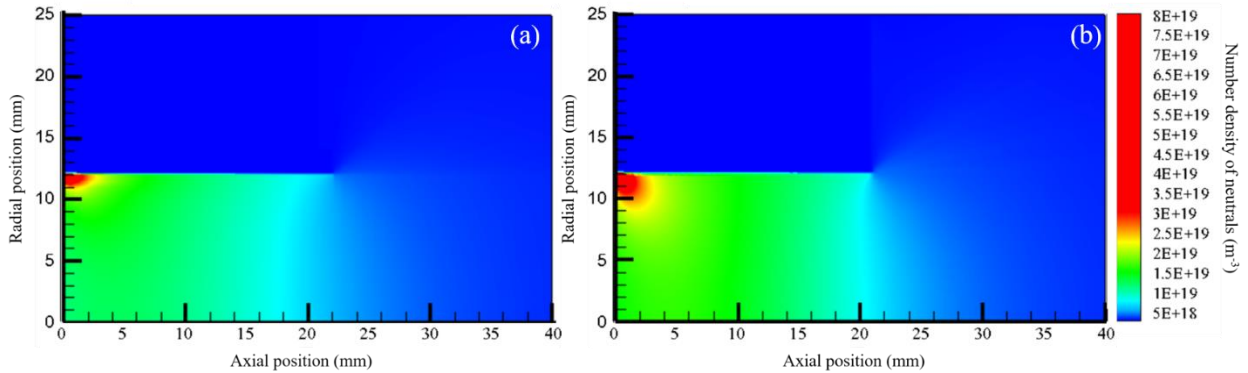


Figure 29. 3sccm Xeon neutral density simulated by COMSOL(a) and view factor model(b)

As **Fig.29** shows, the tendency is the same, which is further supported by the density along the axial position at $r=10\text{mm}$ (**Fig.30**). However, the distribution near the gas feed is different as we set the feed as a gas pipe feeding in COMSOL and a neutral radiation source in view factor model. Thus, the shape appears different and the density simulated by view factor is higher. The whole density simulated by view factor model is higher than the COMSOL near the exit of the EP chamber, because the gas in view factor model can transfer to more area than in COMSOL.

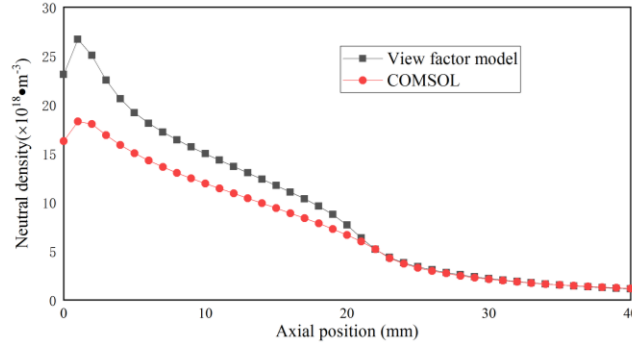


Figure 30. Neutral number density along r=10mm

The neutral density distribution of initial and stable state of neutrals (**Fig.31**) illustrates that the trend of neutral change little but the magnitude is smaller after the ionization progress in the CHT chamber.

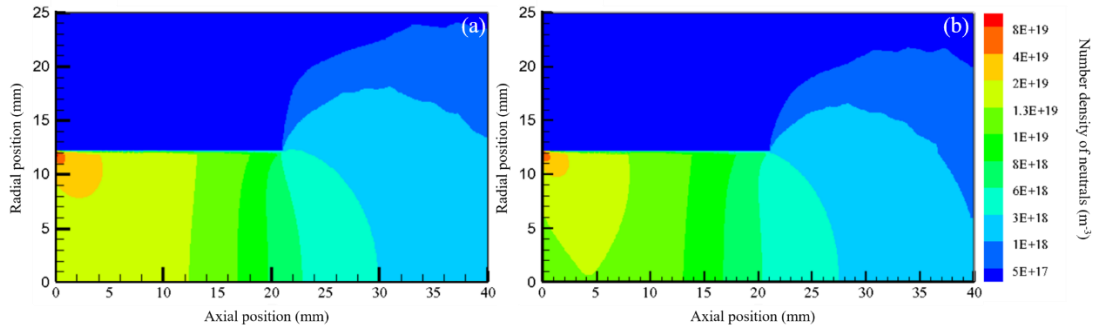


Figure 31. Neutral density distribution of initial (a) and stable state (b)

An intermediate variable to calculate β , β_{iv} , is defined in Eq(33) to explain the neutral consumption.

$$\beta_{iv} = K_{iz} n_e \quad (33)$$

β_{iv} represents the velocity and density information of electrons in the discharge chamber. Large β_{iv} appears at the position where a large amount of electrons or fast electrons exist, causing easier ionization for neutrals passing through. As Eq(28), $\beta = \sum_n [\beta_{iv} \cdot dt]_n$, **Fig.32** illustrates the ionization rate tendency using β_{iv} distribution. The value of β_{iv} is large along the trajectories of electrons, however, is small near the area of neutral gas feed and plume area. Neutrals are ionized when pass along the high β_{iv} area, which shows an integral effect. Thus, it can be explained that the neutral distribution shape is not obviously changed.

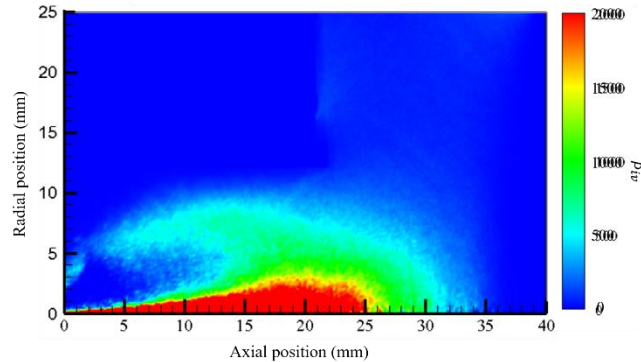


Figure 32. The intermediate variable β_{iv} distribution

The convergent curves of serial codes with and without view factor model are shown as **Fig.33**. The total macro numbers of electrons and ions simulated by the view factor are greater than the code without the model. It is caused by the higher density of neutrals simulated by view factor model, which can produce more ionization. It costs more

steps for PIC-MCC codes with view factor model to reach a stable state as the escalating effect is more obvious. The escalating effect is common in the ignition of hall thrusters, so the updating neutrals per 10000 time steps can provide more accurate physical progress information than the PIC-MCC code treating neutrals as background distribution.

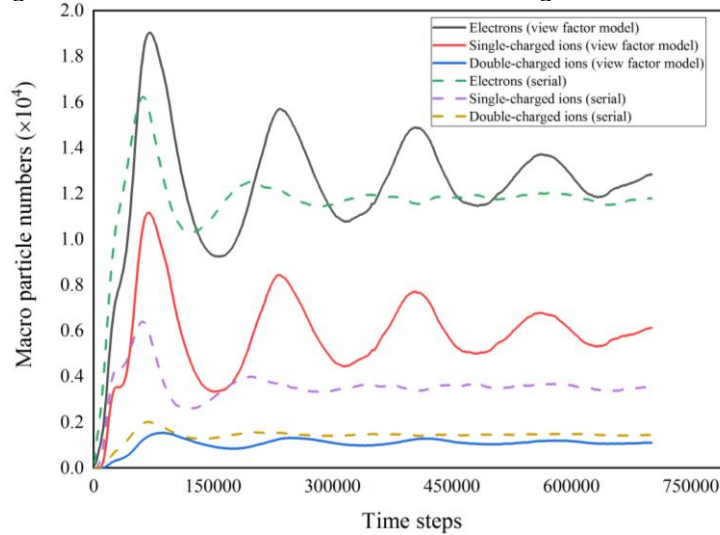


Figure 33. The convergent curve of serial code with (case C) and without (case A) view factor model

IV. Conclusion

Based on the existing serial PIC-MCC code in JLPP, we have developed a set of parallel codes using particle decomposition method, which is suitable for relatively small scale of plasma simulation. The parallel codes are natural computational balance for different processes and have a good performance, especially under 16 parallel processes. The codes can also be applied to other EP thrusters like ion thrusters, pulsed plasma thrusters, etc. However, due to the difficulties of DADI parallelization and the communication time, the highest speedup is about 16, which is still not high enough to simulate high density plasma devices. So we will develop more complex parallel codes using domain decomposition. View factor model has been combined into the serial code, providing updating 3D neutrals distributions instead of fixed ones. The updating neutrals makes the simulation at the balance of fast (compared with the full PIC-MCC treating neutrals like particles) and accurate (compared with the PIC-MCC code with fixed-neutral model). This method can be also applied into other cylinder-shape plasma devices. For the future work, the view factor model will be also parallel, to further codes acceleration.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 11872093). Special thanks would be given to Ph.D. Candidate Zhiyuan Chen on the professional help on PIC-MCC coding.

References

- ¹Vatistas, G. H., Lin, S., and Kwok, C. K., "Reverse Flow Radius in Vortex Chambers," *AIAA Journal*, Vol. 24, No. 11, 1986, pp. 1872, 1873.
- ²Dornheim, M. A., "Planetary Flight Surge Faces Budget Realities," *Aviation Week and Space Technology*, Vol. 145, No. 24, 9 Dec. 1996, pp. 44-46.
- ³Terster, W., "NASA Considers Switch to Delta 2," *Space News*, Vol. 8, No. 2, 13-19 Jan. 1997, pp., 1, 18.
- All of the preceding information is required. The journal issue number ("No. 11" in Ref. 1) is preferred, but the month (Nov.)
1. Szabo JJ. Fully Kinetic Numerical Modeling of a Plasma Thruster. *Ph.d.thesis Massachusetts Inst.of Technol.* 2001.
2. Hirakawa M. Electron Transport Mechanism in a Hall Thruster. In: *Proceedings of the IEPC*. Cleveland; 1997:97-021.
3. Cho S, Komurasaki K, Arakawa Y. Kinetic particle simulation of discharge and wall erosion of a Hall thruster. *Phys Plasmas*. 2013;20(6). doi:10.1063/1.4810798
4. Taccogna F, Longo S, Capitelli M, Schneider R. Self-similarity in Hall plasma discharges: Applications to particle models. *Phys Plasmas*. 2005;12(5):053502. doi:10.1063/1.1877517
5. Liu H, Wu B, Yu D, Cao Y, Duan P. Particle-in-cell simulation of a Hall thruster. *J Phys D Appl Phys*. 2010;43(16). doi:10.1088/0022-3727/43/16/165202
6. Jiang Y, Tang H, Ren J, Li M, Cao J. Magnetic mirror effect in a cylindrical Hall thruster. *J Phys D Appl Phys*. 2018;51(3):035201. doi:10.1088/1361-6463/aa9e3e

7. Sullivan KU. PIC simulation of SPT Hall thrusters: high power operation and wall effects. 2004.
8. Ding Y, Sun H, Li P, et al. Application of hollow anodes in a Hall thruster with double-peak magnetic fields. *J Phys D Appl Phys*. 2017;50(33):335201. doi:10.1088/1361-6463/aa7bbf
9. Mahalingam S, Menart JA. Particle-Based Plasma Simulations for an Ion Engine Discharge Chamber. *J Propuls Power*. 2010;26(4):673-688. doi:10.2514/1.45954
10. Mahalingam S. Particle Based Plasma Simulation for an Ion Engine Discharge Chamber. 2007.
11. Mahalingam S, Choi Y, Loverich J, Stoltz P, Bias B, Menart J. Fully Coupled Electric Field/PIC-MCC Simulation Results of the Plasma in the Discharge Chamber of an Ion Engine. In: *47th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2011:1-24. doi:10.2514/6.2011-6071
12. Cheng J, Tang H-B, York TM. Energy conversion and transfer for plasmas in a magnetic expansion configuration. *Phys Plasmas*. 2014;21(6):063501. doi:10.1063/1.4881475
13. Li M, Tang H-B, Ren J-X, York TM. Modeling of plasma processes in the slowly diverging magnetic fields at the exit of an applied-field magnetoplasma dynamic thruster. *Phys Plasmas*. 2013;20(10):103502. doi:10.1063/1.4824619
14. Liewer PC, Decyk VK. A general concurrent algorithm for plasma particle-in-cell simulation codes. *J Comput Phys*. 1989;85(2):302-322. doi:10.1016/0021-9991(89)90153-8
15. Wang J, Cao Y, Kafafy R, Decyk V. Electric Propulsion Plume Modeling Using Parallel Supercomputers. In: *37th AIAA Plasmadynamics and Lasers Conference*. Vol 15. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2006:83-94. doi:10.2514/6.2006-3559
16. Di Martino B, Briguglio S, Vlad G, Sguazzero P. Parallel PIC plasma simulation through particle decomposition techniques. *Parallel Comput*. 2001;27(3):295-314. doi:10.1016/S0167-8191(00)00098-3
17. Mikellides IG, Katz I. Numerical simulations of Hall-effect plasma accelerators on a magnetic-field-aligned mesh. *Phys Rev E*. 2012;86(4):046703. doi:10.1103/PhysRevE.86.046703
18. Birdsall CK, Langdon AB. *Plasma Physics via Computer Simulation*. CRC press; 2004.
19. Serikov VV, Kawamoto S, Nanbu K. Particle-in-cell plus direct simulation Monte Carlo (PIC-DSMC) approach for self-consistent plasma-gas simulations. *IEEE Trans Plasma Sci*. 1999;27(5):1389-1398. doi:10.1109/27.799817
20. KONDRATYEV AA. 2D numerical modeling of pulsed plasma acceleration by a magnetic field. *Laser Part Beams*. 2000;18(2):S0263034600182163. doi:10.1017/S0263034600182163
21. Taccogna F, Schneider R, Longo S, Capitelli M. Kinetic simulations of a plasma thruster. *Plasma Sources Sci Technol*. 2008;17(2):024003. doi:10.1088/0963-0252/17/2/024003
22. Mahalingam S, Menart J. Computational Model Tracking Primary Electrons, Secondary Electrons and Ions in the Discharge Chamber of an Ion Engine. In: *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2005:AIAA 2005-4254. doi:10.2514/6.2005-4253
23. Verboncoeur JP, Langdon AB, Gladd NT. An object-oriented electromagnetic PIC code. *Comput Phys Commun*. 1995;87(1-2):199-211. doi:10.1016/0010-4655(94)00173-Y
24. Schneider R, Matyash K, Kalentev O, Taccogna F, Koch N, Schirra M. Particle-in-Cell Simulations for Ion Thrusters. *Contrib to Plasma Phys*. 2009;49(9):655-661. doi:10.1002/ctpp.200910070
25. Cao S, Ren J, Tang H, et al. Numerical simulation of plasma power deposition on hollow cathode walls using particle-in-cell and Monte Carlo collision method. *Phys Plasmas*. 2018;25(10):103512. doi:10.1063/1.5029945
26. Adam JC, Héron A, Laval G. Study of stationary plasma thrusters using two-dimensional fully kinetic simulations. *Phys Plasmas*. 2004;11(1):295-305. doi:10.1063/1.1632904
27. Ding Y, Sun H, Li P, et al. Influence of hollow anode position on the performance of a hall-effect thruster with double-peak magnetic field. *Vacuum*. 2017;143:251-261. doi:10.1016/j.vacuum.2017.06.030
28. Araki SJ, Air E, Base F. Sensitivity of Ion Flux on Spacecraft to Neutral Density in Hall Thruster Plume. In: *35th International Electric Propulsion Conference Georgia Institute of Technology •Atlanta, Georgia •USA October 8–12, 2017*. ; 2017:1-12.
29. Katz I, Mikellides IG. Neutral gas free molecular flow algorithm including ionization and walls for use in plasma simulations. *J Comput Phys*. 2011;230(4):1454-1464. doi:10.1016/j.jcp.2010.11.013
30. Huerta CE, Matlock TS, Wirz RE. Validation of a Plasma-Facing Surface Sputtering and Deposition View Factor Model. In: *AIAA SPACE 2015 Conference and Exposition*. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2015:1-9. doi:10.2514/6.2015-4503
31. Wirz RE. Discharge Plasma Process of Ring-cusp Ion Thruster. 2005;2005.
32. Cao S, Ren J, Tang H, et al. Modeling on plasma energy balance and transfer in a hollow cathode. *J Phys D Appl Phys*. 2019;52(28):285202. doi:10.1088/1361-6463/ab197e
33. Tang H-B, Cheng J, Liu C, York TM. Study of applied magnetic field magnetoplasma dynamic thrusters with particle-in-cell code with Monte Carlo collision. I. Computation methods and physical processes. *Phys Plasmas*. 2012;19(7):073107. doi:10.1063/1.4737098
34. Tang H-B, Cheng J, Liu C, York TM. Study of applied magnetic field magnetoplasma dynamic thrusters with particle-in-cell and Monte Carlo collision. II. Investigation of acceleration mechanisms. *Phys Plasmas*. 2012;19(7):073108. doi:10.1063/1.4737104
35. Yu D, Zhang F, Liu H, Li H, Yan G, Liu J. Effect of electron temperature on dynamic characteristics of two-dimensional sheath in Hall thrusters. *Phys Plasmas*. 2008;15(10):13-16. doi:10.1063/1.2988766

36. Cappelli M, Meezan N, Gascon N. Transport physics in Hall plasma thrusters. In: *40th AIAA Aerospace Sciences Meeting & Exhibit.* ; 2002:485.
37. Waltz RE, Candy JM, Rosenbluth MN. Gyrokinetic turbulence simulation of profile shear stabilization and broken gyroBohm scaling. *Phys Plasmas*. 2002;9(5):1938-1946.
38. Fox JM. Parallelization of a Particle-in-Cell Simulation Modeling Hall-Effect Thrusters. *Master Thesis Massachusetts Inst Technol*. 2005;(2003):139.
39. Vahedi V, Surendra M. A Monte Carlo collision model for the particle-in-cell method: applications to argon and oxygen discharges. *Comput Phys Commun*. 1995;87(1-2):179-198. doi:10.1016/0010-4655(94)00171-W
40. Goebel DM, Katz I. Basic Plasma Physics. In: *Fundamentals of Electric Propulsion*. Hoboken, NJ, USA: John Wiley & Sons, Inc.; 2008:37-90. doi:10.1002/9780470436448.ch3
41. Wang J, Polk J, Brophy J, Katz I. Three-Dimensional Particle Simulations of Ion-Optics Plasma Flow and Grid Erosion. *J Propuls Power*. 2003;19(6):1192-1199. doi:10.2514/2.6939
42. Thomas LH. Elliptic problems in linear difference equations over a network. *Watson Sci Comput Lab Rept, Columbia Univ New York*. 1949;1.
43. Ping ZJ. *Solving Partial Differential Equations on Parallel Computers*. world scientific; 1994.
44. Khor YK, Hung YM, Lim BK. On the role of radiation view factor in thermal performance of straight-fin heat sinks. *Int Commun Heat Mass Transf*. 2010;37(8):1087-1095. doi:10.1016/j.icheatmasstransfer.2010.06.012
45. Andrienko D, Surzhikov S, Shang JJ, Huang GP. On View-Factor Approach for Radiation Transfer Equation. In: *45th AIAA Plasmadynamics and Lasers Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2014:1-17. doi:10.2514/6.2014-2488
46. Dressler RA, Chiu Y, Pullins SH, Miller JS, Levandier DJ. Xenon charge exchange cross sections for electrostatic thruster models. *J Appl Phys*. 2002;91(3):984-991. doi:10.1063/1.1426246
47. Howell, J., Menguc, M., Siegel R. *Thermal Radiation Heat Transfer*. 6th ed. Boca Raton: CRC Press; 2015. doi:10.1201/b18835
48. Wordingham CJ, Taunay P-Y, Choueiri E. A Critical Review of Orificed Hollow Cathode Modeling: 0D Models. In: *53rd AIAA/SAE/ASEE Joint Propulsion Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics; 2017:1-52. doi:10.2514/6.2017-4888