Computers and Mathematics with Applications 75 (2018) 1887–1899

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

An improved immersed finite element particle-in-cell method for plasma simulation[☆]

Jinwei Bai [a], Yong Cao [a], Yuchuan Chu [a,b,*], Xu Zhang [c]

[a] Department of Mechanical Engineering & Automation, Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, Guangdong 518055, PR China
[b] Department of Mathematics & Statistics, Missouri University of Science & Technology, Rolla, MO 65401, USA
[c] Department of Mathematics & Statistics, Mississippi State University, Mississippi State, MS 39762, USA

ARTICLE INFO

Article history:
Available online 30 August 2017

Keywords:
Plasma simulation
IFE–PIC
Interface problem
Particle interpolation

ABSTRACT

The particle-in-cell (PIC) method has been widely used for plasma simulation, because of its noise-reduction capability and moderate computational cost. The immersed finite element (IFE) method is efficient for solving interface problems on Cartesian meshes, which is desirable for the PIC method. The combination of these two methods provides an effective tool for plasma simulation with complex interface/boundary. This paper introduces an improved IFE–PIC method that enhances the performance in both IFE and PIC aspects. For the electric field solver, we adopt the newly developed partially penalized IFE method with enhanced accuracy. For PIC implementation, we introduce a new interpolation technique to ensure the conservation of the charge. Numerical examples are provided to demonstrate the features of the improved IFE–PIC method.

1. Introduction

There are two classes of methods for plasma simulation. The first one is the traditional dynamic simulation [1], which is mainly used to obtain the distribution functions of particles by solving the Vlasov equation. The second one is the particle simulation method [2–4], which is used to track a large number of individual particles and to obtain the trajectory parameters and characteristics of plasma by statistical methods. Due to the enormous number of particles to be tracked and the limited computational resources, the development of the particle simulation method was quite slow. The particle simulation method entered a rapid developing period [5–7] since Birdsall and Langdon [8] introduced the particle-in-cell (PIC) method which utilizes the finite-sized particle (or cloud) instead of a huge number of real particles.

Immersed finite element (IFE) method is a finite element method for solving interface problems on uniform Cartesian meshes, which was first proposed by Li et al. [9]. Different from classical finite element methods using body-fitted meshes, the mesh of the IFE method is independent of the interface. However, the IFE basis functions around interfaces are modified to accommodate the interface jump conditions. The advantage of the IFE method is that structured Cartesian meshes can be used to solve interface problems with arbitrary interface geometry. For problems with a moving interface, IFE methods are especially advantageous since there is no need to regenerate the solution meshes repeatedly [10–12]. The IFE methods

[☆] This work was supported by National Natural Science Foundation of China [Grant numbers 10875034, 11175052]; NSF Grant DMS-1720425; China Postdoctoral Science Foundation (Grant number 2016M591517); Shenzhen Technology Project [JCYJ20150529115038093, JCYJ20160226201347750].

[*] Corresponding author at: Department of Mechanical Engineering & Automation, Harbin Institute of Technology, Shenzhen Graduate School, Shenzhen, Guangdong 518055, PR China.

E-mail address: ychuan.chu@hit.edu.cn (Y. Chu).

http://dx.doi.org/10.1016/j.camwa.2017.08.001
0898-1221/© 2017 Elsevier Ltd. All rights reserved.

have been developed for solving the second-order elliptic equations [13–17], elasticity equations [18–20], and Stokes equations [21], to name only a few.

In the past decade, the IFE method has been used together with the PIC method for plasma simulations [22–25]. The IFE method used as an electric field solver is performed on well-structured Cartesian meshes. This is particularly desirable for the PIC method because tracking a large number of plasma particles can be efficiently done in a uniform structured mesh. We refer to a few recent applications of the IFE–PIC method for different types of particle simulations, such as ion thruster [26,27], hall thruster [28], and lunar surface environment [29,30]. Also, the IFE–PIC method has been extended to handle unbounded interface problems with asymptotic boundary condition [31] and periodic boundary condition [32].

For the current IFE–PIC method, we noticed that there are two issues. First, the classical Galerkin IFE method is used as the field solver. As shown in [33], the classical IFE method is not accurate around the interface, because the IFE basis functions are discontinuous across the element boundaries, and the classical Galerkin formulation cannot control such discontinuity. Second, the particle interpolation method of the PIC algorithm is imperfect. The conventional interpolation approach applied on the interface element often leads to the non conservation of charge, because it neglects the fact that some nodes of the interface cells are inside the conductors. Similar problems occur in applying the electric field force to the particles on the interface elements.

In this paper, we introduce an improved IFE–PIC method that focuses on overcoming the problems mentioned above. As a remedy of discontinuity of IFE field solver, we adopt the newly developed partially penalized immersed finite element (PPIFE) method [33] to improve the accuracy of IFE methods near interfaces. For PIC interpolation, we introduce a two-step approach for particle interpolation that preserves the charge conservation. Comparing with conventional charge distribution in PIC, we add a correction step that redistributes the quantity distributed to the nodes inside the conductor to the remaining nodes in order to maintain charge conservation. In addition, we use IFE basis functions to calculate the electric field and force on the interface elements. The new approach can calculate the motion of particles more accurately.

The rest of the article is organized as follows. In Section 2, we recall the classical IFE method and the PIC interpolation. In Section 3, we present our improved IFE–PIC method. The improvement in the IFE solver part is the PPIFE method with additional penalty terms. The improvement in the PIC part includes the new particle interpolation scheme and the new method for the force deposit. In Section 4, we present some numerical experiments to compare the performance of traditional IFE–PIC method and improved IFE–PIC method. Brief conclusions will be given in Section 5.

2. Review of IFE–PIC method

In this section, we first recall the main steps in a typical IFE–PIC computational cycle. Then we will recall the classical IFE method and PIC interpolation that are widely used in the literature.

2.1. Main steps of IFE–PIC method

Real plasma particles are modeled as many macro-particles in the PIC method, and they follow the evolution of the orbits of individual particles in the self-consistent electromagnetic field. The field is then updated by solving the governing elliptic equation with discontinuous dielectric coefficients. The IFE–PIC method is an iteration of solving for the electromagnetic field and particle motion until the steady state is achieved.

In general, an IFE–PIC computational cycle consists of the following five steps:

Step 1. Initialization A series of initial settings of the simulation including domain, mesh, boundary condition, and initial position and velocity of particles are set up.

Step 2. Particle push The motion of the particle is induced by the particles themselves and the external fields $\boldsymbol{E}$. The trajectory of an individual charged particle is obtained by integrating the Newton–Lorentz equation

$$m\frac{d\mathbf{v}}{dt} = q(\boldsymbol{E} + \mathbf{v} \times \boldsymbol{B}), \tag{1}$$

where m, q and $\mathbf{v}$ are the mass, the charge, and the velocity of the particle, respectively. $\boldsymbol{B}$ denotes the static magnetic field.

Step 3. Charge deposit The change of the positions of particles leads to the change of the charge density ρ on each node. Thus, we need to calculate the charge density at each node according to the new positions of particles. The process of interpolating the particle charges on the discrete mesh points is called weighting. In traditional PIC method, we use the ratio of the area of the rectangle formed by the opposite cell vertex and the particle to the area of element as the weighting.

Step 4. Solving for potential After obtaining the charge density ρ on each node, the electric field should also be updated. The potential function $\Phi(\mathbf{x})$ can be described by the second-order Poisson's equation with discontinuous dielectric coefficient $\beta(\mathbf{x})$, which represents different types of material:

$$-\nabla \cdot (\beta \nabla \Phi) = \rho(\mathbf{x}). \tag{2}$$

To solve this equation, we use the IFE method as a field solver, because of its applicability of Cartesian mesh, which is desirable in the PIC simulation for fast tracking of particles' locations.

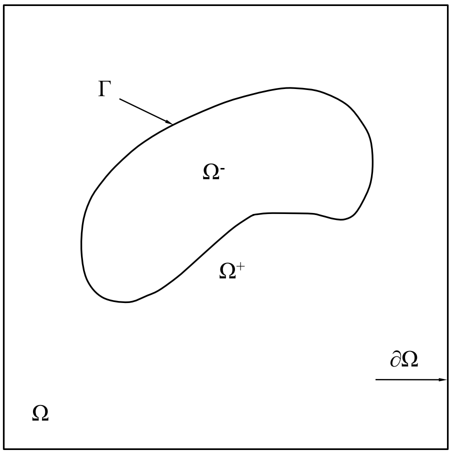


Fig. 1. A sketch of the domain for the interface problem.

Step 5. Force deposit After solving for the potential $\Phi(\mathbf{x})$, we calculate the electric field $\boldsymbol{E}$ at each node by

$$\boldsymbol{E}(\mathbf{x}) = -\nabla\Phi(\mathbf{x}). \tag{3}$$

Next, we need to deposit the electric field at nodes to the particles with arbitrary positions. The electric field of each node can be obtained using two-point difference method from the potential $\Phi(\mathbf{x})$ on each element node. The electric field of each particle can be obtained by interpolating.

We note that, in step 3, when the particle is located in an interface element, the current interpolation using the simple area-weighting will result in the non-conservation of charge. Also, in step 4, the classical IFE method may not be as accurate around interface as the rest of the domain due to the discontinuity of IFE basis functions. Finally, in step 5, the force deposit is not accurate in the interface element due to part of the element is the conductor. In this paper, our improvement of the IFE–PIC method will mainly focus on these three steps.

2.2. Classical IFE method for interface problems

The electric field $\Phi(\mathbf{x})$ is assumed to be governed by the following second-order elliptic equation:

$$-\nabla\cdot(\beta\nabla\Phi) = f, \quad \text{in } \Omega, \tag{4}$$

$$\Phi = g, \quad \text{on } \partial\Omega. \tag{5}$$

Here, we assume that $\Omega \subset \mathbb{R}^2$ is a rectangular domain separated by an interface curve Γ into two sub-domains Ω^-, Ω^+ such that $\overline{\Omega} = \overline{\Omega^+ \cup \Omega^- \cup \Gamma}$. See Fig. 1 as an illustration.

The coefficient $\beta(\mathbf{x})$ is discontinuous across the interface. Without loss of generality, we assume β is a piecewise constant function as follows:

$$\beta(\mathbf{x}) = \begin{cases} \beta^-, & \mathbf{x} \in \Omega^-, \\ \beta^+, & \mathbf{x} \in \Omega^+, \end{cases} \tag{6}$$

where $\mathbf{x} = [x, y]^t$. Across the interface Γ, the following interface jump conditions are satisfied:

$$[\![\Phi]\!]_\Gamma = 0, \tag{7}$$

$$\left[\!\!\left[\beta\frac{\partial\Phi}{\partial n}\right]\!\!\right]_\Gamma = 0. \tag{8}$$

Let $\mathcal{T}_h = \{T\}$ be a uniform triangular or rectangular mesh of the domain Ω with size h. If an element T is cut through by the interface, it is called an interface element; otherwise, it is said to be a noninterface element. The sets of interface elements and noninterface elements are denoted by $\mathcal{T}_h^i$ and $\mathcal{T}_h^n$, respectively. Standard linear or bilinear finite element functions are used on all noninterface elements. Special piecewise-polynomial basis functions are constructed on interface element to accommodate the interface conditions. To be more precise, we use the linear IFE method as an example. Assume that $T = \triangle A_1A_2A_3$ is an interface triangle, and the interface curve Γ intersects T at two points, denoted by D and E. The

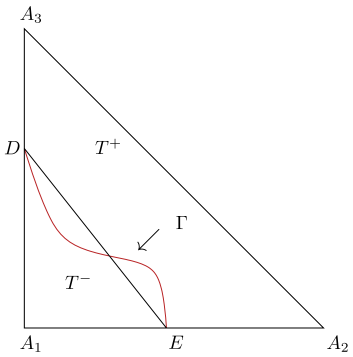


Fig. 2. An interface triangle.

element T is divided into two sub-elements T^+ and T^- by the line segment $\overline{DE}$. See Fig. 2 as an illustration. The local IFE basis functions $\phi_{i,T}, i = 1, 2, 3$ on T are defined as follows:

$$\phi_{i,T}(\mathbf{x}) = \begin{cases} \phi^+_{i,T}(\mathbf{x}) = a^+_i x + b^+_i y + c^+_i, & \mathbf{x} \in T^+, \\ \phi^-_{i,T}(\mathbf{x}) = a^-_i x + b^-_i y + c^-_i, & \mathbf{x} \in T^-, \end{cases} \quad \forall\, i = 1, 2, 3. \tag{9}$$

They satisfy the following conditions:

1. **Nodal-value conditions**

$$\phi_{i,T}(A_j) = \delta_{ij} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j, \end{cases} \quad \forall\, i, j = 1, 2, 3. \tag{10}$$

2. **Function-value continuity**

$$\phi^+_{i,T}(D) = \phi^-_{i,T}(D), \quad \phi^+_{i,T}(E) = \phi^-_{i,T}(E), \quad \forall\, i = 1, 2, 3. \tag{11}$$

3. **Flux continuity**

$$\beta^+ \frac{\partial \phi^+_{i,T}}{\partial \mathbf{n}} = \beta^- \frac{\partial \phi^-_{i,T}}{\partial \mathbf{n}}, \quad \forall\, i = 1, 2, 3, \tag{12}$$

where $\mathbf{n}$ is the normal vector of $\overline{DE}$.

The local linear IFE space is defined as

$$\tilde{\mathbb{P}}_1(T) = span\{\phi_{1,T}, \phi_{2,T}, \phi_{3,T}\}. \tag{13}$$

Denote the set of interior nodes on $\mathcal{T}_h$ by $\mathcal{N}_h$. On each node $\mathbf{x}_i, i = 1, 2, \ldots, N$, we define the global linear IFE basis function Φ_i such that

$$\Phi_i(\mathbf{x}_j) = \delta_{ij}, \quad \forall\, i, j = 1, 2, \ldots, N, \tag{14}$$

and

$$\Phi_i|_T \in \tilde{\mathbb{P}}_1(T), \quad \forall\, T \in \mathcal{T}^i_h, \quad \text{and} \quad \Phi_i|_T \in \mathbb{P}_1(T) = span\{1, x, y\}, \quad \forall\, T \in \mathcal{T}^n_h. \tag{15}$$

The global linear IFE space is formed as $S_h = span\{\Phi_i : i = 1, 2, \ldots, N\}$.

The classical (Galerkin) IFE method is to find $u_h \in S_h$ such that

$$a(u_h, v_h) = (f, v_h), \quad \forall\, v_h \in S_h, \tag{16}$$

where

$$a(u, v) = \sum_{T \in \mathcal{T}_h} \int_T \beta \nabla u \cdot \nabla v d\mathbf{x}, \quad (f, v) = \int_\Omega f v d\mathbf{x}. \tag{17}$$

The construction of the bilinear IFE spaces on rectangular meshes are similar, and we refer to [13,14,20] for more details.

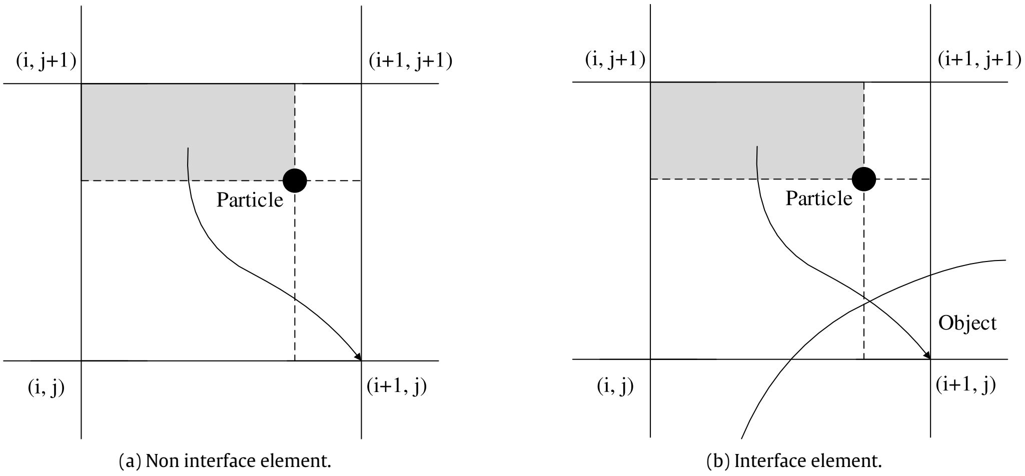


Fig. 3. Standard PIC deposition scheme of particle charge in a 2D simulation domain.

2.3. *Particle interpolation in traditional PIC method*

In the PIC method, particle interpolation is required at Step 3 and Step 5 described in Section 2.1. First, in charge deposit in Step 3, as shown in Fig. 3(a), the portion of the total particle charge assigned to a certain cell vertex is proportional to the area of the rectangle formed by the opposite cell vertex and the particle. Thus, the interpolation of the particle P, located at position X_P, to the node $X_{i,j}$ can be calculated by

$$q_{i,j} = q_P \frac{S(X_{i+1,j+1}, X_P)}{S(X_{i,j}, X_{i+1,j+1})}, \tag{18}$$

where $q_{i,j}$ is the amount of charge of node (i, j), q_P is the amount of charge of particle P, and $S(A, B)$ is the area of the rectangle whose diagonal is $\overline{AB}$.

On non-interface elements, the interpolation (18) works well. However, on interface elements, it will face some difficulty, as shown in Fig. 3(b), the charge should not be assigned to the node $X_{i+1,j}$, which is inside the conducting object. Directly applying (18) on interface elements will cause the non-conservation for the total charge and charge density.

Moreover, in Step 5 the electric field is required to be interpolated at the particles' positions, so that the electric field force can be obtained. The potential Φ at grid points are solved by the IFE method with the appropriate boundary conditions. Then, the electric field $\boldsymbol{E} = [E_x, E_y]^t$ can be obtained from the potential Φ using the following equation:

$$\boldsymbol{E}(x, y) = -\nabla\Phi(x, y). \tag{19}$$

The conventional approach for this interpolation uses the finite difference form for E_x and E_y:

$$(E_x)_{i,j} = \frac{\Phi_{i-1,j} - \Phi_{i+1,j}}{2\Delta x}, \quad (E_y)_{i,j} = \frac{\Phi_{i,j-1} - \Phi_{i,j+1}}{2\Delta y}, \tag{20}$$

where Δx and Δy are mesh sizes in the x- and y-directions, respectively. After the electric field is obtained, the forces caused by the field at mesh nodes can be deposited to the arbitrary particles positions.

When a particle is located in the interface element, the calculation (20) is apparently inaccurate, because the electric field of the four points that belong to different objects are discontinuous.

3. Improved IFE–PIC method

In this section, we present an improved IFE–PIC method. The improvement involves both the IFE solver and the PIC particle interpolation.

3.1. *Partially penalized IFE method*

Due to the discontinuity of IFE basis functions, classical IFE method using Galerkin formulation may generate large errors around the interface. Partially penalized immersed finite element (PPIFE) method is introduced in [33] that is known to greatly improve the accuracy of IFE solution around interface. The main idea of this method is to add penalty terms on interface edges to reduce the negative impact of the discontinuity introduced by IFE functions. In our improved IFE–PIC method, we adopt this new PPIFE scheme as the new field solver.

To present the PPIFE method, we need to introduce a few notations. Let $\mathcal{E}_h$ be the set of all interior edges of the mesh $\mathcal{T}_h$. If an edge $e \in \mathcal{E}_h$ intersects with the interface curve Γ, we call it an interface edge; otherwise, a noninterface edge. The sets of interface edges and noninterface edges are denoted by $\mathcal{E}_h^i$ and $\mathcal{E}_h^n$, respectively. For each interior edge $e \in \mathcal{E}_h$, it must be shared by two adjacent elements, denoted by $T_{e,1}$ and $T_{e,2}$. For a function u defined on $T_{e,1} \cup T_{e,2}$, the average and jump of u on e are defined as follows:

$$\{u\}_e = \frac{1}{2}\Big((u|_{T_{e,1}})_e + (u|_{T_{e,2}})_e\Big), \qquad [\![u]\!]_e = (u|_{T_{e,1}})_e - (u|_{T_{e,2}})_e. \tag{21}$$

The PPIFE method for solving (4)–(8) is to find $u_h \in S_h$ such that

$$a_\epsilon(u_h, v_h) = (f, v_h), \quad \forall v_h \in S_h, \tag{22}$$

where

$$a_\epsilon(u, v) = \sum_{T\in\mathcal{T}_h}\int_T \beta\nabla u\cdot\nabla v d\mathbf{x} - \sum_{e\in\mathcal{E}_h}\int_e \{\beta\nabla u\cdot\mathbf{n}\}[\![v]\!]\,ds + \epsilon\sum_{e\in\mathcal{E}_h}\int_e\{\beta\nabla v\cdot\mathbf{n}\}[\![u]\!]\,ds + \sum_{e\in\mathcal{E}_h}\int_e \frac{\sigma_e}{|e|}[\![u]\!]\,[\![v]\!]\,ds. \tag{23}$$

Here, ϵ can be chosen as -1, 0, or 1, which corresponds to symmetric, incomplete, or non-symmetric PPIFE methods, respectively. σ_e is a positive penalty parameter. As shown in [33], for non-symmetric PPIFE method, the scheme (22) is stable as long as $\sigma_e > 0$ on every edge e. For symmetric and incomplete PPIFE methods, (22) is stable when σ_e is large enough.

3.2. Charge-conservative particle interpolation

As shown in Section 2, the particle interpolation technique (18) is not accurate for particles in the interface elements. Also, the total charge is not conservative due to part of the nodes are inside the objects such as metal or ceramic materials. In this subsection, we introduce a two-step interpolation that can maintain the charge conservation.

3.2.1. Improved algorithm for charge deposit

We note that if a particle is in a noninterface element, it is suitable to use the standard charge distribution (18) with area weight method. If a particle P is in an interface element, we introduce a correction step to redistribute the charge assigned to nodes inside the conductor to other nodes. This step ensures the conservation of the charge in the simulation domain. The complete two-step interpolation is as follows:

Step 1: Initial distribution Interpolate the charge q_p of the particle to four nodes of the element using the standard area-weight approach:

$$q_i^{(1)} = w_i q_p, \quad i = 1, 2, 3, 4, \tag{24}$$

where

$$w_1 = \frac{A_4}{A}, \quad w_2 = \frac{A_3}{A}, \quad w_3 = \frac{A_2}{A}, \quad w_4 = \frac{A_1}{A}.$$

Here A_i, $i = 1, 2, 3, 4$, are the areas of small rectangles, illustrated in Fig. 4.

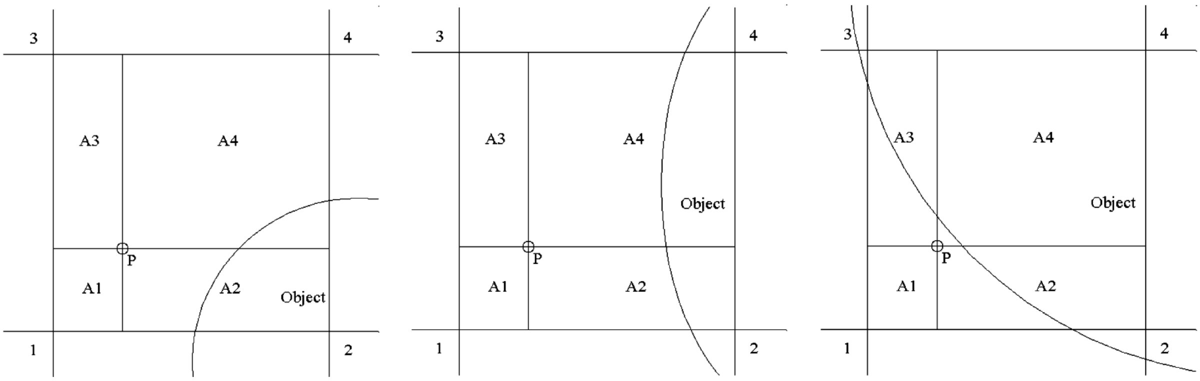
Step 2: Re-distribution Correct the charge distribution to nodes inside the object.

- **Case 1**: There is only one node (e.g. node #2) in the object, as shown in Fig. 4(a). The charge initially distributed to node #2 should be redistributed to the other three nodes. The correction procedure is the following:

$$\begin{aligned} q_1^{(2)} &= q_1^{(1)} + \frac{w_1}{w_1+w_3+w_4}q_2^{(1)},\\ q_2^{(2)} &= 0,\\ q_3^{(2)} &= q_3^{(1)} + \frac{w_3}{w_1+w_3+w_4}q_2^{(1)},\\ q_4^{(2)} &= q_4^{(1)} + \frac{w_4}{w_1+w_3+w_4}q_2^{(1)}. \end{aligned}$$

- **Case 2**: If two nodes are inside the object, as shown in Fig. 4(b), then the correction procedure becomes

$$\begin{aligned} q_1^{(2)} &= q_1^{(1)} + \frac{w_1}{w_1+w_3}(q_2^{(1)} + q_4^{(1)}),\\ q_2^{(2)} &= 0,\\ q_3^{(2)} &= q_3^{(1)} + \frac{w_3}{w_1+w_3}(q_2^{(1)} + q_4^{(1)}),\\ q_4^{(2)} &= 0. \end{aligned}$$



(a) One node inside the object.

(b) Two nodes inside the object.

(c) Three nodes inside the object.

Fig. 4. New deposit algorithm of particle charge.

- **Case 3**: If three nodes are inside the object, as shown in Fig. 4(c), then

$$\begin{aligned} q_1^{(2)} &= q_1^{(1)} + q_2^{(1)} + q_3^{(1)} + q_4^{(1)}, \\ q_2^{(2)} &= 0, \\ q_3^{(2)} &= 0, \\ q_4^{(2)} &= 0. \end{aligned}$$

We note that if we use the traditional PIC charge interpolation (without correction) on interface elements, the total charge of nodes is less than the charge of particle, that is

$$\sum_{i\in\Lambda} q_i^{(1)} \neq q_P, \tag{25}$$

where Λ denotes the indices of nodes outside the object. However, adding the correction, the interpolation satisfies

$$\sum_{i\in\Lambda} q_i^{(2)} = q_P. \tag{26}$$

This clearly shows that the new interpolation scheme preserves the charge-conservation.

Moreover, this new interpolation scheme is robust regarding the location of the particle. That is, if the particle touches the boundary of the interface element, the algorithm is still valid.

3.2.2. Improved algorithm for force deposit

The potential distribution at grid points is obtained by solving linear algebraic equations contained in the immersed finite element method, and then for the electric potential at point (x, y) in an element T can be approximated using the following equation:

$$\Phi_T(x, y) = \sum_{i=1}^{d} u_i \phi_{i,T}(x, y). \tag{27}$$

Here, u_i are the numerical solutions of the potential at the vertices of T, and $\phi_{i,T}$ are local IFE basis functions. The degree of freedom $d = 3$ for the linear IFE method, and $d = 4$ for the bilinear IFE method.

Suppose the electrical field is in Ω^+, and the conductor is in Ω^-. Then the electric field $\mathbf{E}$ at a particle $P(x, y)$ can be obtained by

$$\begin{aligned} E_x(x, y) &= -\left(u_1 \frac{\partial \phi_1^+(x, y)}{\partial x} + u_2 \frac{\partial \phi_2^+(x, y)}{\partial x} + \cdots + u_d \frac{\partial \phi_d^+(x, y)}{\partial x}\right), \\ E_y(x, y) &= -\left(u_1 \frac{\partial \phi_1^+(x, y)}{\partial y} + u_2 \frac{\partial \phi_2^+(x, y)}{\partial y} + \cdots + u_d \frac{\partial \phi_d^+(x, y)}{\partial y}\right). \end{aligned} \tag{28}$$

For points located in T^-, the electric field can be calculated in the same way. This method is more accurate than linear interpolation when the internal and external potential of objects are discontinuous. With the new PPIFE field solver, and the improved PIC interpolation technique on particles and force, the workflow of our improved IFE–PIC algorithm can be summarized in Fig. 5.

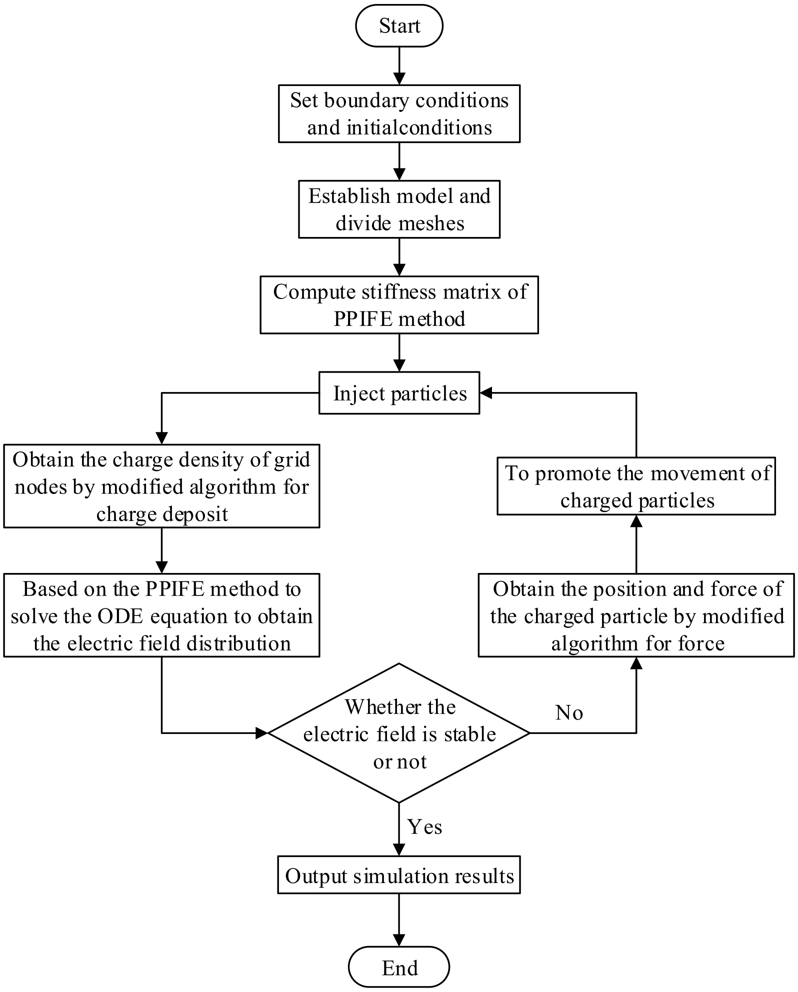

Start
Set boundary conditions and initialconditions
Establish model and divide meshes
Compute stiffness matrix of PPIFE method
Inject particles
Obtain the charge density of grid nodes by modified algorithm for charge deposit
To promote the movement of charged particles
Based on the PPIFE method to solve the ODE equation to obtain the electric field distribution
Obtain the position and force of the charged particle by modified algorithm for force
Whether the electric field is stable or not
No
Yes
Output simulation results
End

Fig. 5. Flow chart of the improved IFE-PIC method.

4. Numerical examples

In this section, we present some numerical examples to demonstrate the features of the improved IFE–PIC method.

We set up a test problem of conducting cylinder with background plasma particles. Let the two-dimensional simulation domain be $\Omega_s = [-1, 1] \times [-1, 1]$. The center and the radius of the cylinder are set to be $O = [0, 0]$ and $r_0 = \pi/12$, respectively, as shown in Fig. 6. For plasma particles, we load 1 635 841 ($= 1279^2$) uniformly distributed particles $P_{i,j}$ into the simulation domain and the positions of particles are given by

$$P_{i,j} = \left[-1+\frac{i}{1280}, -1+\frac{j}{1280}\right], \quad i,j = 1, 2, \ldots, 1279. \tag{29}$$

Assume that the loaded particles are electrons, and the charge density on the mesh point is $\rho = -4$, so we can get the charge of each particle. Then we remove all the particles inside the cylinder. We choose appropriate Dirichlet boundary conditions and source function so that the analytical solution of this simulation problem is

$$\Phi(x,y) = \begin{cases} \Phi^-(x,y) = \dfrac{1}{\beta^-}r^2 + \left(\dfrac{1}{\beta^+} - \dfrac{1}{\beta^-}\right)r_0^2, & \text{if } r \le r_0, \\ \Phi^+(x,y) = \dfrac{1}{\beta^+}r^2, & \text{if } r > r_0 \end{cases} \tag{30}$$

where $r = \sqrt{x^2+y^2}$, and $[\beta^-, \beta^+] = [1, 10]$. The analytical solution (30) of the potential in the simulation domain is shown in Fig. 6.

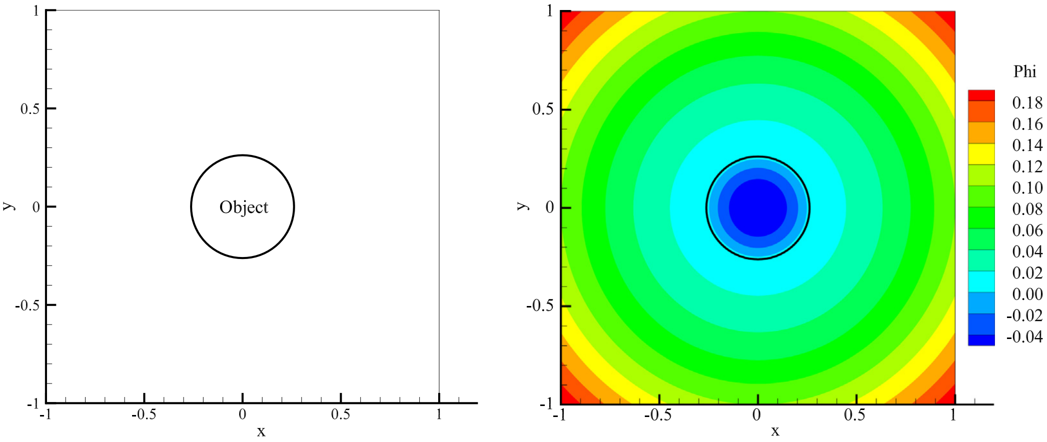


Fig. 6. Simulation domain Ω_s (left) and the analytical solution (right).

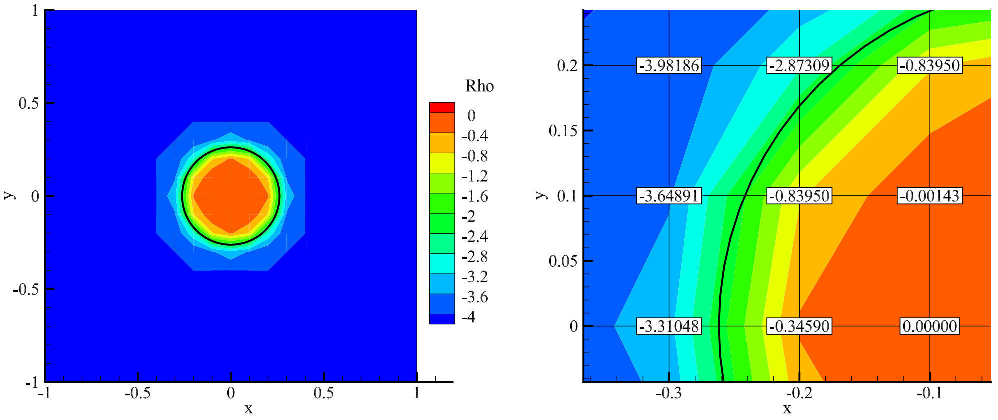


Fig. 7. Density distribution using standard PIC interpolation.

4.1. Comparison of PIC interpolations

We first compare the performance of traditional PIC charge interpolation with the new interpolation method. The charge densities on mesh points can be obtained by depositing the physical quantities (charge of electrons) from the particle locations onto mesh nodes. Fig. 7 shows the charge density distribution with standard charge deposit algorithm on a 20 × 20 Cartesian mesh. It can be shown that all the charge of interface element is less than the prescribed value $\rho = -4$, because the charge of particles are deposited to several mesh points inside the object.

Using the improved PIC algorithm introduced in Section 3.2, the charge density distribution in simulation domain Ω_s is shown in Fig. 8. On the right side of Fig. 8, we show a zoom-in plot around interface. It can be shown that the charge on the nodes of interface elements are not unanimously less than -4. On some nodes (e.g. $[-0.3, 0.1]$, and $[-0.3, 0.2]$), the change quantities are actually greater than -4. This is because of the redistribution step we added, so that the charge of particles will not be deposited to any mesh points inside the conductor. Moreover, this new interpolation algorithm ensures the conservation of the charge in the calculation domain.

4.2. Comparison of IFE–PIC performance

Next, we combine the new PIC algorithm with the new electrical field solver, the PPIFE scheme. We compare the performance of traditional and improved IFE–PIC schemes. In Figs. 9 and 10, we plot the numerical errors of electrical field by traditional and improved IFE–PIC methods, respectively. It can be easily observed that the new IFE–PIC method has much smaller error than the traditional IFE–PIC method especially around the interface.

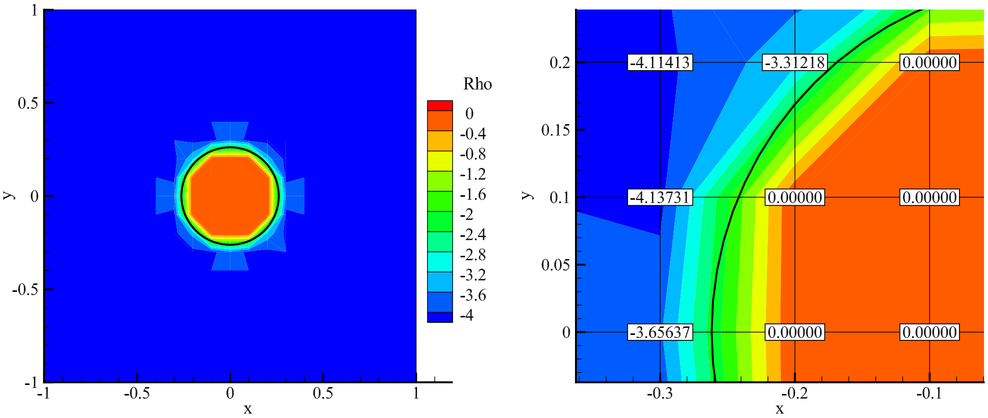


Fig. 8. Density distribution using improved PIC interpolation.

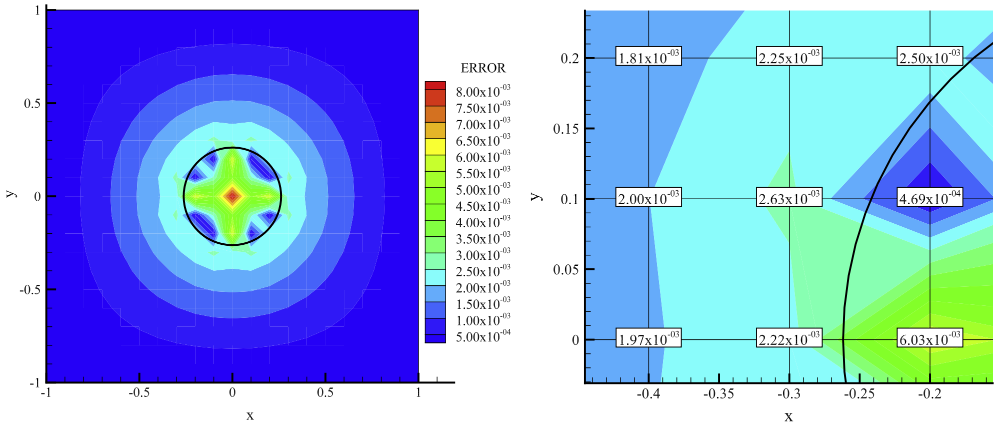


Fig. 9. Error of traditional IFE-PIC solution on a Cartesian mesh of Ω_s.

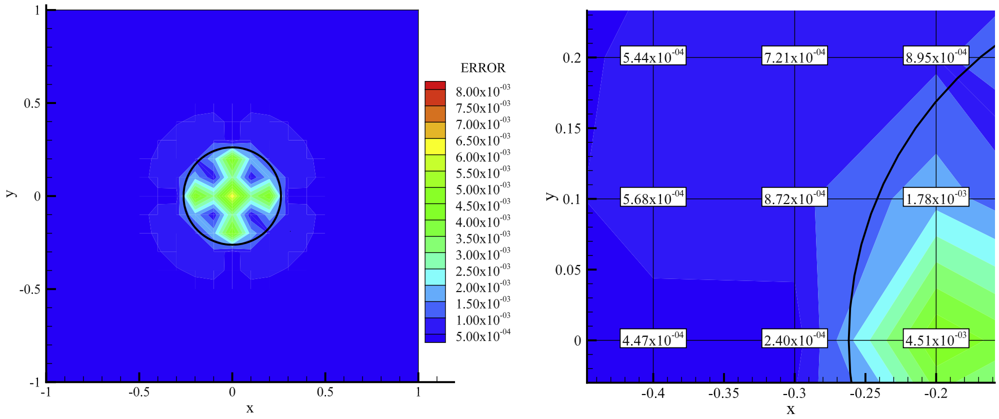


Fig. 10. Error of improved IFE-PIC solution on a Cartesian mesh of Ω_s.

4.3. Sensitivity of the number of particles

In this test, we focus on the sensitivity of our IFE–PIC method to the number of particles. Note that on non-interface elements, regardless of the number of particles within each element, the final charge of interpolation to nodes are same,

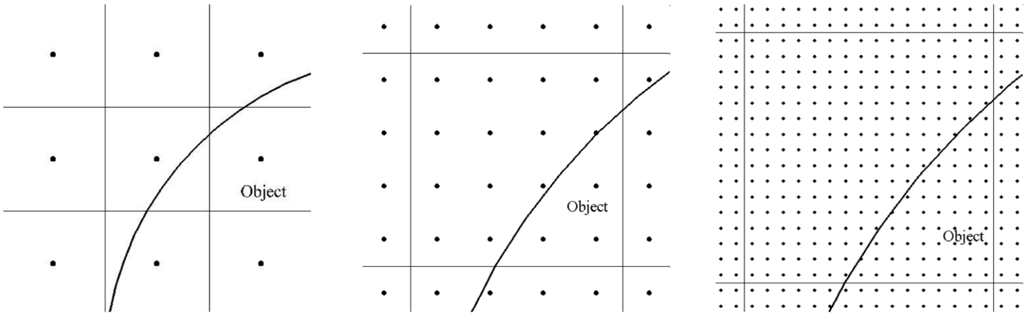


Fig. 11. Effect of different particle numbers on charge distribution in interface element.

Table 1
Comparison of the error of charge distribution.

N	Standard PIC		Improved PIC	
	$\overline{\rho}$	E_ρ	$\overline{\rho}$	E_ρ
1	−2.600000	35.00%	−4.083332	2.08%
4	−3.250000	18.75%	−4.145832	3.65%
16	−3.240625	18.98%	−4.286457	7.16%
64	−3.481812	12.95%	−4.238281	5.96%
256	−3.479996	13.00%	−4.236654	5.92%
1024	−3.482821	12.93%	−4.240478	6.01%

Table 2
Comparison of the L^2 norm error of electric potential.

N	IFE-PIC	Improved IFE-PIC
1	1.868830×10^{-3}	5.160630×10^{-4}
4	1.190797×10^{-3}	4.851096×10^{-4}
16	1.208151×10^{-3}	6.142589×10^{-4}
64	1.281327×10^{-3}	5.380597×10^{-4}
256	1.284765×10^{-3}	5.359054×10^{-4}
1024	1.280203×10^{-3}	5.403805×10^{-4}

i.e., $\rho = -4$. However, on interface elements, the element is split into two parts. The particles located in the conductor will not contribute to the charge distribution, as a result, the number of particles in an interface element should have greater influence on the charge distribution results.

Fig. 11 demonstrate an interface element with 1, or 16, or 256 equally distributed particles, respectively. As shown in Fig. 11, if the number of particles is small, the charge distribution will rely more on the random fall of individual particles. On the other hand, if the number of particles is large, the effect of a single particle could be neglected. This is consistent with the actual situation.

Next, we compute the average density for different number of particles with traditional and improved PIC interpolations. We use N to denote the number of particles in each interface element. Let $\mathbf{x}_i$, $i = 1, \ldots, N_I$ be the nodes that belong to at least one interface element. The mean value of density at the interface nodes is computed as follows:

$$\overline{\rho} = \frac{1}{N_I} \sum_{i=1}^{N_I} \rho_h(\mathbf{x}_i), \tag{31}$$

where the $\rho_h(\mathbf{x}_i)$ is the interpolated charging density at the point $\mathbf{x}_i$. The percentage error of the density is defined by

$$E_\rho = \frac{|\rho - \overline{\rho}|}{|\rho|}. \tag{32}$$

The L^2 norm error of the potential function is defined by

$$E_u = \|u - u_h\| = \left(\int_\Omega |u - u_h|^2 d\mathbf{x} \right)^{\frac{1}{2}}. \tag{33}$$

Tables 1 and 2 report the error of charging density and error of electric potential using the two IFE–PIC methods. These tests are conducted on a uniform 40 × 40 Cartesian mesh. Table 1 clearly indicates that as we increase the number of particles that belong to each element, the traditional PIC interpolation becomes more accurate, this indicates that the

Table 3
The L^2 norm errors of electric potential for different mesh.

Mesh	Traditional IFE-PIC	Improved IFE-PIC
10 × 10	1.112170×10^{-2}	8.131783×10^{-3}
20 × 20	3.953483×10^{-3}	2.520218×10^{-3}
40 × 40	1.280203×10^{-3}	5.403805×10^{-4}
80 × 80	5.275457×10^{-4}	1.784664×10^{-4}
160 × 160	3.553394×10^{-4}	1.313621×10^{-4}
320 × 320	1.372616×10^{-4}	1.103237×10^{-5}
Rate	1.240231	1.771787

traditional approach is very sensitive to the number of particles. On the other hand, the new interpolation technique is much more accurate than the traditional approach, and are more robust with respect to the number of particles. Table 2 shows that the improved IFE–PIC scheme is about three times more accurate than the classical scheme in the L^2 norm of the electric potential.

Remark 4.1. Based on our current numerical experiments, simply improving only one part of IFE solver or PIC interpolation is less significant, and the behavior on accuracy varies in different choices of interface geometry, conductivity coefficient, and the number of particles.

4.4. *Convergence of potential on a sequence of meshes*

In this example, we test convergence of potential on a family of uniform meshes with the same number of particles in each element. We consider that the number of particles is the same as in Section 4.1. Table 3 reports the L^2 norm error of the potential. From the comparison, we can see that our improved IFE–PIC scheme is much more accurate on every mesh level than the widely used scheme.

5. Conclusions

In this paper, we proposed a new IFE–PIC method for plasma simulation. The new method has improvement in both IFE solver and PIC interpolation. One prominent feature is that our new PIC interpolation has the charge-conservation property. Moreover, the improved IFE solver with partially penalty terms produces more accurate approximation around the interface.

References

[1] A. Vlasov, On vibration properties of electron gas, J. Exp. Theor. Phys. 8 (3) (1938) 291.

[2] J. Dawson, One-dimensional plasma model, Phys Fluids 5 (4) (1962) 445–459.

[3] O. Eldridge, M. Feix, One-dimensional plasma model at thermodynamic equilibrium, Phys. Fluids 5 (9) (1962) 1076–1080.

[4] R.W. Hockney, J.W. Eastwood, Computer Simulation using Particles, Taylor and Francis, 1981.

[5] P. Birch, S. Chapman, Two dimensional particle-in-cell simulations of the lunar wake, Phys. Plasma 9 (5) (2002) 1785–1789.

[6] S. Kimura, T. Nakagawa, Electromagnetic full particle simulation of the electric field structure around the moon and the lunar wake, Earth Planets Space 60 (6) (2008) 594–599.

[7] L. Schwager, C. Birdsall, Collector and source sheaths of a finite ion temperature plasma, Phys. Fluids B: Plasma Phys. 2 (5) (1990) 1057–1068.

[8] A. Langdon, C. Birdsall, Theory of plasma simulation using finite-size particles, Phys. Fluids 13 (8) (1970) 2115–2122.

[9] Z. Li, The immersed interface method using a finite element formulation, Appl. Numer. Math. 27 (3) (1998) 253–267.

[10] X. He, T. Lin, Y. Lin, X. Zhang, Immersed finite element methods for parabolic equations with moving interface, Numer. Methods Partial Differential Equations 29 (2) (2013) 619–646.

[11] T. Lin, Y. Lin, X. Zhang, Immersed finite element method of lines for moving interface problems with nonhomogeneous flux jump, in: Recent Advances in Scientific Computing and Applications, in: Contemp. Math., vol. 586, Amer. Math. Soc., Providence, RI, 2013, pp. 257–265.

[12] T. Lin, Y. Lin, X. Zhang, A method of lines based on immersed finite elements for parabolic moving interface problems, Adv. Appl. Math. Mech. 5 (4) (2013) 548–568.

[13] X. He, T. Lin, Y. Lin, Approximation capability of a bilinear immersed finite element space, Numer. Methods Partial Differential Equations 24 (5) (2008) 1265–1300.

[14] X. He, T. Lin, Y. Lin, Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions, Int. J. Numer. Anal. Model. 8 (2) (2011) 284–301.

[15] Z. Li, T. Lin, Y. Lin, R.C. Rogers, An immersed finite element space and its approximation capability, Numer. Methods Partial Differential Equations 20 (3) (2004) 338–367.

[16] Z. Li, T. Lin, X. Wu, New Cartesian grid methods for interface problems using the finite element formulation, Numer. Math. 96 (1) (2003) 61–98.

[17] T. Lin, Q. Yang, X. Zhang, A priori error estimates for some discontinuous Galerkin immersed finite element methods, J. Sci. Comput. 65 (3) (2015) 875–894.

[18] Y. Gong, Z. Li, Immersed interface finite element methods for elasticity interface problems with non-homogeneous jump conditions, Numer. Math. Theory Methods Appl. 3 (1) (2010) 23–39.

[19] T. Lin, D. Sheen, X. Zhang, A locking-free immersed finite element method for planar elasticity interface problems, J. Comput. Phys. 247 (2013) 228–247.

[20] T. Lin, X. Zhang, Linear and bilinear immersed finite elements for planar elasticity interface problems, J. Comput. Appl. Math. 236 (18) (2012) 4681–4699.

[21] S. Adjerid, N. Chaabane, T. Lin, An immersed discontinuous finite element method for Stokes interface problems, Comput. Methods Appl. Mech. Engrg. 293 (2015) 170–190.
[22] Y. Cao, Y. Chu, X. He, T. Lin, An iterative immersed finite element method for an electric potential interface problem based on given surface electric quantity, J. Comput. Phys. 281 (15) (2015) 82–95.
[23] Y. Chu, D. Han, Y. Cao, X. He, J. Wang, An immersed-finite-element particle-in-cell simulation tool for plasma surface interaction, Int. J. Numer. Anal. Model. 14 (2) (2017) 175–200.
[24] R. Kafafy, T. Lin, Y. Lin, J. Wang, Three-dimensional immersed finite element methods for electric field simulation in composite materials, Internat. J. Numer. Methods Engrg. 64 (7) (2005) 940–972.
[25] R. Kafafy, J. Wang, T. Lin, A hybrid-grid immersed-finite-element particle-in-cell simulation model of ion optics plasma dynamics, Dyn. Contin. Discrete Impuls. Syst. Ser. B Appl. Algorithms 12 (Suppl. Vol. 12b) (2005) 1–16.
[26] H. Cao, Y. Chu, E. Wang, Y. Cao, G. Xia, Z. Zhang, Numerical simulation study on barrel erosion of ion thruster accelerator grid, J. Propul. Power 31 (6) (2015) 1785–1792.
[27] H. Jian, Y. Chu, H. Cao, Y. Cao, X. He, G. Xia, Three-dimensional IFE-PIC numerical simulation of background pressure's effect on accelerator grid impingement current for ion optics, Vacuum 116 (2015) 130–138.
[28] H. Cao, Q. Li, K. Shan, Y. Cao, L. Zheng, Effect of preionization on the erosion of the discharge channel wall in a hall thruster using a kinetic simulation, IEEE Trans. Plasma Sci. 43 (1) (2015) 130–140.
[29] D. Han, J. Wang, X. He, a nonhomogeneous immersed finite element particle in cell method for modeling dielectric surface charging in plasmas, IEEE Trans. Plasma Sci. 44 (8) (2016) 1326–1332.
[30] D. Han, P. Wang, X. He, T. Lin, J. Wang, A 3D immersed finite element method with non-homogeneous interface flux jump for applications in particle-in-cell simulations of plasma-lunar surface interactions, J. Comput. Phys. 321 (2016) 965–980.
[31] Y. Chu, Y. Cao, X. He, M. Luo, Asymptotic boundary conditions with immersed finite elements for interface magnetostatic/electrostatic field problems with open boundary, Comput. Phys. Comm. 182 (11) (2011) 2331–2338.
[32] Y. Cao, Y. Chu, X. Zhang, X. Zhang, Immersed finite element methods for unbounded interface problems with periodic structures, J. Comput. Appl. Math. 307 (2016) 72–81.
[33] T. Lin, Y. Lin, X. Zhang, Partially penalized immersed finite element methods for elliptic interface problems, SIAM J. Numer. Anal. 53 (2) (2015) 1121–1144.