Computational Particle Mechanics (2022) 9:1279–1291
https://doi.org/10.1007/s40571-022-00470-0

Weak scaling of the parallel immersed-finite-element particle-in-cell (PIFE-PIC) framework with lunar plasma charging simulations

David Lund[1] · Xiaoming He[2] · Xu Zhang[3] · Daoru Han[1]

Received: 29 November 2021 / Revised: 8 February 2022 / Accepted: 12 February 2022 / Published online: 7 March 2022

Abstract
Weak scaling performance of a recently developed fully kinetic, 3-D parallel immersed-finite-element particle-in-cell framework, namely PIFE-PIC, was investigated. A nominal 1-D plasma charging problem, the lunar photoelectron sheath at a low Sun elevation angle, was chosen to validate PIFE-PIC against recently derived semi-analytic solutions of a 1-D photoelectron sheath. The weak scaling performance test shows that the overall efficiency of PIFE-PIC is insensitive to the number of macroparticles and, counterintuitively, more domain decomposition iterations in the field-solve of PIC may lead to faster computing due to better convergence of field solutions at early stages of PIC iteration. The PIFE-PIC framework was then applied to simulate plasma charging of a wavy lunar surface with 324,000 cells and 150 million macroparticles demonstrating the capability of PIFE-PIC in resolving local-scale plasma environment near the surface of the Moon.

Keywords Particle in cell · Weak scaling · Plasma charging · Immersed finite element

1 Introduction

With the renewed interest of surface exploration on the Moon, especially the goal of landing on the polar regions (also known as "the lunar terminator") as set by NASA's Artemis program, it is critical to understand the plasma–surface interactions at the local scale, i.e., near the lunar surface. Lacking an atmosphere and a global magnetic field, the Moon is directly exposed to solar radiation and various space plasma environments (mostly drifting protons and electrons in the solar wind). A direct consequence of such interactions is surface charging caused by bombardment of solar wind plasma and emission/collection of photoelectrons. The plasma sheath formed near the illuminated lunar surface is usually referred to as the "photoelectron sheath" because it is largely dominated by photoelectrons compared with average solar wind plasma. At the lunar polar regions, the near-surface photoelectron sheath and the charged surface are expected to have substantial influence on the charging of landers and rovers during surface missions. Since the lunar surface is covered by a regolith layer, which separates the solid bedrock from the plasma environment [1,2], a complete model of plasma charging on the lunar surface needs to explicitly take into account the properties of the regolith layer, such as the permittivity and layer thickness, as well as the lunar ground at the bedrock. Recently, Han et al. [3] presented a general approach of modeling plasma charging at the lunar surface including the regolith layer as well as the lunar bedrock below the regolith layer. This approach integrated particle-in-cell (PIC) with a non-homogeneous immersed-finite-element (IFE) field solver capable of resolving charging of dielectric materials [4,5]. The 3-D IFE-PIC model is capable of solving the electric field and charge deposition both inside and outside of irregularly shaped objects immersed in a plasma, which is unique among PIC-based charging models. The charging calculation from local charge

✉ Daoru Han
handao@mst.edu

David Lund
dclgzb@mst.edu

Xiaoming He
hex@mst.edu

Xu Zhang
xzhang@okstate.edu

[1] Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, 400 W. 13th Street, Rolla, MO 65409, USA

[2] Department of Mathematics and Statistics, Missouri University of Science and Technology, 400 W. 12th Street, Rolla, MO 65409, USA

[3] Department of Mathematics, Oklahoma State University, 401 Mathematical Sciences, Stillwater, OK 74078, USA

deposition in the PIC approach also enables time-varying modeling of the charging process.

The IFE-PIC code package in Ref. [3] was serial, which has limited its applications to relatively small-sized problems with respect to practical interests. Toward the goal of developing a massively scalable, fully kinetic, multi-scale, multi-species modeling framework for complex plasma–surface interactions, Han et al. [6,7] developed the parallel IFE-PIC (PIFE-PIC) framework with 3-D domain decomposition using Message Passing Interface (MPI) parallel computing architecture, where each subdomain is handled by an individual processor. Specifically, the details of 3-D domain decomposition for both *field-solve* and *particle-push* steps of PIC as well as strong scaling performance tests of PIFE-PIC are given in Ref. [7].

This paper investigates weak scaling of the PIFE-PIC framework. A nominal 1-D problem was chosen to first validate the PIFE-PIC framework against recently derived semi-analytic solutions of a 1-D photoelectron sheath [8] and then examine the weak scaling performance of PIFE-PIC. The rest of the paper is organized as follows. Section 2 briefly describes the IFE algorithm and its applications to kinetic plasma modeling, as well as the PIFE-PIC code suite. Section 3 presents a validation and baseline case for PIFE-PIC applied to 1-D lunar photoelectron sheath simulations. Section 4 investigates weak scaling performance of PIFE-PIC. Section 5 applies PIFE-PIC to simulate charging of a wavy lunar surface demonstrating the capability of PIFE-PIC. Finally, Sect. 6 gives a summary and conclusion.

2 IFE algorithm and the PIFE-PIC suite

The fundamental phenomenon of plasma charging on the surface of the Moon is dielectric surface charging, whereas the equilibrium surface potentials are determined by local current balance condition. Since the shape of the rugged lunar surface terrain is non-trivial, it is important to accurately resolve the interface conditions between the plasma region and the lunar regolith/bedrock region. For electrostatic problems such as lunar plasma charging, PIC methods are widely used to model the charged particles using macroparticles and track the motions of particles. The electric potential ϕ is governed by the following 3-D elliptic boundary value problem:

$$-\nabla\cdot\left(\varepsilon^{-}\nabla\phi^{-}\right)=0,\quad \text{in } \Omega^{-}, \tag{1}$$

$$-\nabla\cdot\left(\varepsilon^{+}\nabla\phi^{+}\right)=\rho(X),\quad \text{in } \Omega^{+}, \tag{2}$$

$$\phi=g(X),\quad \text{on } \Gamma_D, \tag{3}$$

$$\frac{\partial\phi}{\partial\mathbf{n}_{\Gamma_N}}=p(X),\quad \text{on } \Gamma_N, \tag{4}$$

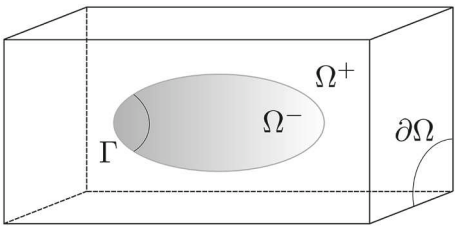


Fig. 1 Computational domain with an interface

where $X=(x,y,z)$, and ρ is the charge density function. The computational domain Ω is separated by an interface Γ into two subdomains Ω^+ and Ω^-. Γ_D and Γ_N denote the Dirichlet and the Neumann portion of the boundary $\partial\Omega$, with given boundary values g and p, respectively. $\mathbf{n}_{\Gamma_N}$ is the outward normal of Γ_N. See Fig. 1 for an illustration of the problem setup.

Across the interface Γ, the following interface jump conditions are enforced:

$$[\phi]_{\Gamma}=0, \tag{5}$$

$$\left[\varepsilon\frac{\partial\phi}{\partial\mathbf{n}_{\Gamma}}\right]_{\Gamma}=q, \tag{6}$$

where q is a given charge density function and $\mathbf{n}_{\Gamma}$ is the unit normal vector of Γ from Ω^- to Ω^+. The dielectric coefficient $\varepsilon(X)$ is discontinuous across the interface due to the material property change. Without loss of generality, we assume that $\varepsilon(X)$ is a piecewise constant function as follows:

$$\varepsilon(X)=\begin{cases}\varepsilon^{-}, & \text{in } \Omega^{-},\\ \varepsilon^{+}, & \text{in } \Omega^{+},\end{cases}$$

where $\min(\varepsilon^{+},\varepsilon^{-})>0$.

In general, there are two types of numerical methods for interface problems: fitted-mesh methods and unfitted-mesh methods. Conventional numerical methods, such as finite element method, require solution meshes to align with material interfaces. In general, these body-fitting meshes are unstructured if the interface geometry is nontrivial. Particle tracking performed on unstructured mesh is inefficient because a global search of elements is inevitable, which significantly increases the computational costs. See the left plot of Fig. 2. On the other hand, structured meshes, such as the Cartesian mesh, have inherent limitations of resolving complex interface geometries, such as the lunar surface. To overcome this type of difficulties, the immersed-finite-element (IFE) method among many other unfitted-mesh methods [9–13] was developed to solve interface problems based on interface-unfitted meshes while maintained sufficient approximation accuracy. These interface-unfitted numerical methods are particularly desirable in PIC sim-

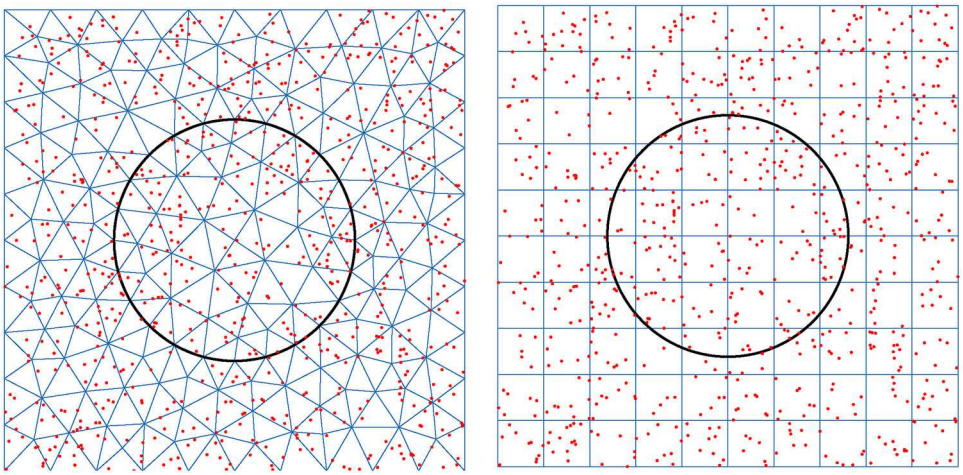


Fig. 2 A body-fitted mesh and an unfitted mesh for interface problems with particle tracking. (Colour figure online)

ulations because they enable efficient particle tracking as illustrated in the right plot of Fig. 2.

The main idea of IFE methods is to incorporate physical interface jump conditions in the design of local IFE functions. This idea was first introduced in Ref. [14] for one-dimensional elliptic interface problems with piecewise linear polynomial approximation. In the past two decades, the IFE methods have been extensively studied for elliptic interface problems [15–23], planar elasticity system [24–26], parabolic interface problems [27,28], hyperbolic interface problems [29–31], Stokes interface problems [32,33], etc. It has been shown that the IFE method can achieve optimal convergence on an interface-independent mesh with the number and location of the degrees-of-freedom isomorphic to the standard FEM on the same mesh [34–36]. These immersed-finite-element functions have been used in various numerical frameworks such as discontinuous Galerkin method [37–39], finite volume method [40–42], and nonconforming finite element method [43].

The IFE method has been successfully used together with PIC in plasma particle simulations [44–49]. Recently, a non-homogeneous IFE-PIC algorithm has been developed for particle simulations of plasma–material interactions with complex geometries while maintaining the computational speed of the Cartesian-mesh-based PIC [4,50–53]. In the past decades, the IFE-PIC method has matured to successfully model plasma dynamics problems arising from many space applications, such as ion thruster grid optics [54–57], ion propulsion plume-induced contamination [58–60], charging of lunar and asteroidal surfaces [3,5,61–64], and dust transport dynamics around small asteroids [65].

The PIFE-PIC package was developed based on the serial non-homogeneous IFE-PIC as presented in Refs. [4,5]. In PIFE-PIC, the computation domain is first decomposed into cubic blocks with the same PIC mesh size. Local (not necessarily uniform) IFE mesh is then generated for each subdomain. For the parallel electrostatic field solver, Dirichlet–Dirichlet domain decomposition with overlapping cells is used to distribute the subdomains among multiple MPI processes [66,67]. For each subdomain, the IFE solver is the same as the serial IFE method with Dirichlet boundary conditions [4,53,68–70]. These Dirichlet boundary conditions are imposed at the boundaries of the subdomains, which are also interior for the neighboring subdomains. Therefore, the field solution at respective neighboring subdomains are used as Dirichlet boundary conditions for each subdomain. Within the field-solve part of each step in the PIC loop, inner iterations among subdomains are performed such that the solutions of the overlapping cells are exchanged and updated as the new Dirichlet boundary conditions for the respective neighboring subdomains. Simulation particles belonging to a certain subdomain are stored together on the processor that solves the field of the same subdomain. In this sense, "particle quantities" and "field quantities" of each subdomain are handled by the same processor. Data communications are implemented at inner boundaries ("guard cell" regions) used to interchange field solutions and particle data for needed calculations. Algorithm 1 describes key steps of PIFE-PIC in the form of pseudocode showing three levels of iteration (loop): the main PIC loop, the domain decomposition method (DDM) loop, and the matrix solver preconditioned conjugate gradient (PCG) loop. More details of the PIFE-PIC framework used in this study are presented in Ref. [7], and the serial version of the field solver that handles dielectric charging was described in detail in Refs. [4,5].

3 Code validation and baseline simulation

In this section, we validate the PIFE-PIC code for the simulation of a nominal 1-D photoelectron sheath against recently derived semi-analytic solutions [8]. Since the upcoming Artemis Moon missions are targeted toward the lunar terminator region where the Sun elevation angle (SEA) is low,

Algorithm 1 Pseudocode of key steps of PIFE-PIC

```
Start
Read inputs: domain, geometry, and plasma data
Initialize: MPI directives, domain decomposition, and arrays
Load particles into domain
! Solve initial electrostatic field (it_PIC = 0) below:
while it_PIC = 0 do
    it_DDM = 0
    while it_DDM < it_DDMMaxInitial do
        Each subdomain solves its own field using PCG
        Exchange boundary conditions among subdomains (DDM loop)
        if DDM loop converges then
            Exit
        else
            Continue
        end if
        it_DDM = it_DDM + 1
    end while
    it_PIC = it_PIC + 1
end while
! Initial field solved
! Start main PIC loop
while it_PIC ≤ it_PICMax do
    Gather: calculate force on each particle
    Inject particles from global boundaries
    Move particles and handle inner boundaries among subdomains
    Handle particles going out of global domain
    Handle particles hitting objects: collect charges carried by particles
    Scatter: calculate space charge density for field-solve
    ! Field-solve below:
    ! Same as initial field solve but use it_DDMMaxPICloop
    ! Field solved for this PIC step
    it_PIC = it_PIC + 1
end while
! Main PIC loop finished
End
```

we chose a 5-degree SEA that would result in a monotonic electric potential profile near the surface [8,71–73]. Figure 3 shows a schematic of the simulation cases used for validation/baseline case (serial) and the weak scaling performance test (parallel, to be discussed in Sect. 4).

3.1 Problem description and simulation setup

This validation study contains a plasma species with thermal electrons and cold drifting ions impinging a flat surface. Other parameters of the charged species, including photoelectons, such as number density n, drifting velocity v_d, thermal velocity v_t, temperature T, and Debye length λ_D, are selected to represent average solar wind conditions at 1 AU, as shown in Table 1.

The validation case uses a computation domain of $1.5 \times 1.5 \times 100$ (normalized by Debye length of photoelectrons at 90° SEA) with a globally uniform PIC and IFE mesh, i.e., both meshes have a mesh size of $h = 0.5$. Thus, the entire simulation domain has $3 \times 3 \times 200$ PIC cells which totals to 1800 cubic PIC cells ($1800 \times 5 = 9000$ tetrahedral FE/IFE cells since each cuboid PIC cell is partitioned into five tetrahedral FE/IFE cells). The simulation domain contains two flat surfaces that physically correspond to the lunar bedrock interface and the lunar regolith surface. The lunar bedrock interface is located at $\hat{z} = 1.99$, and the lunar regolith surface is at $\hat{z} = 5.70$. This validation study utilizes the serial configuration of the PIFE-PIC code, such that the entire computation domain is partitioned into $1 \times 1 \times 1$ subdomain; therefore, the whole domain is computed by only one MPI process.

Particles were preloaded into the domain before the initial field solution and injected into the domain at $Z_{\max}$ within each PIC step. On average, 1728 macroparticles representing solar wind electrons ($12 \times 12 \times 12$) and 4096 macroparticles representing solar wind ions ($16 \times 16 \times 16$) were populated in one PIC cell. Particles hitting the $X_{\min}$, $X_{\max}$, $Y_{\min}$, and $Y_{\max}$ boundaries were treated with periodic conditions. Particles hitting the $Z_{\max}$ boundary were absorbed and removed from the domain. Particles hitting the lunar regolith surface were collected to calculate the non-homogeneous flux jump condition, which was then used to self-consistently solve for the electric field including the floating potential of the regolith surface.

At the $Z_{\min}$ and $Z_{\max}$ boundaries, the potentials are set to 0 as the reference potential. At $X_{\min}$, $X_{\max}$, $Y_{\min}$, and $Y_{\max}$ boundaries, zero-Neumann boundary conditions are applied due to the 1-D configuration. The relative permittivity of the regions for both the lunar bedrock and regolith layer was set to 4 [1]. The normalized time step size was set to be 0.05. The convergence criterion of the field-solve was set at a tolerance of 1×10^{-6} for the absolute residual of the preconditioned conjugate gradient (PCG) matrix solver with the maximum number of PCG iteration set to 1000. (For the serial case, the PCG solver took about 300 iterations to converge, more details in Sect. 4.) All the runs presented in this study were performed on AMD EPYC™ Rome (Ryzen 2) compute nodes provided by the Center for High Performance Computing Research at Missouri University of Science and Technology. This serial validation case took approximately 18.5 wall-clock hours for 40,000 PIC steps when the steady state was reached.

3.2 Comparison with semi-analytic solutions and performance profiling

Following the steps detailed in Ref. [8], semi-analytic solutions to the density and potential profiles of the 1-D photoelectron sheath were obtained to validate the PIFE-PIC solution. These comparisons are plotted in Fig. 4. Excellent agreement between two PIFE-PIC and semi-analytic solutions is achieved. This validation case shows the suitability of the setup and the fully kinetic PIFE-PIC framework to

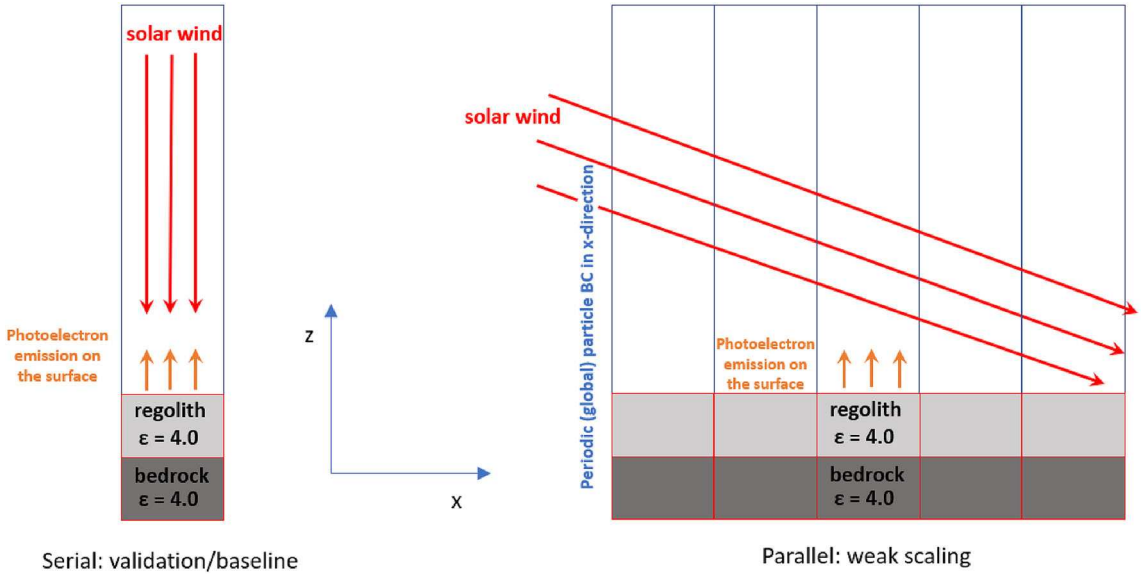


Fig. 3 Simulation cases for validation/baseline (serial, left) and weak scaling performance test (parallel, right). (Colour figure online)

Table 1 Average solar wind and photoelectron (at 90° SEA) parameters at 1 AU*

	Solar wind electron	Solar wind ion	Photoelectron
n, cm^{-3}	8.7	8.7	64
v_d, $\times 10^7$ cm/s	4.68	4.68	N/A
v_t, $\times 10^7$ cm/s	14.53	0.31	6.22
T, eV	12	10	2.2
λ_D, m	8.73	N/A	1.38

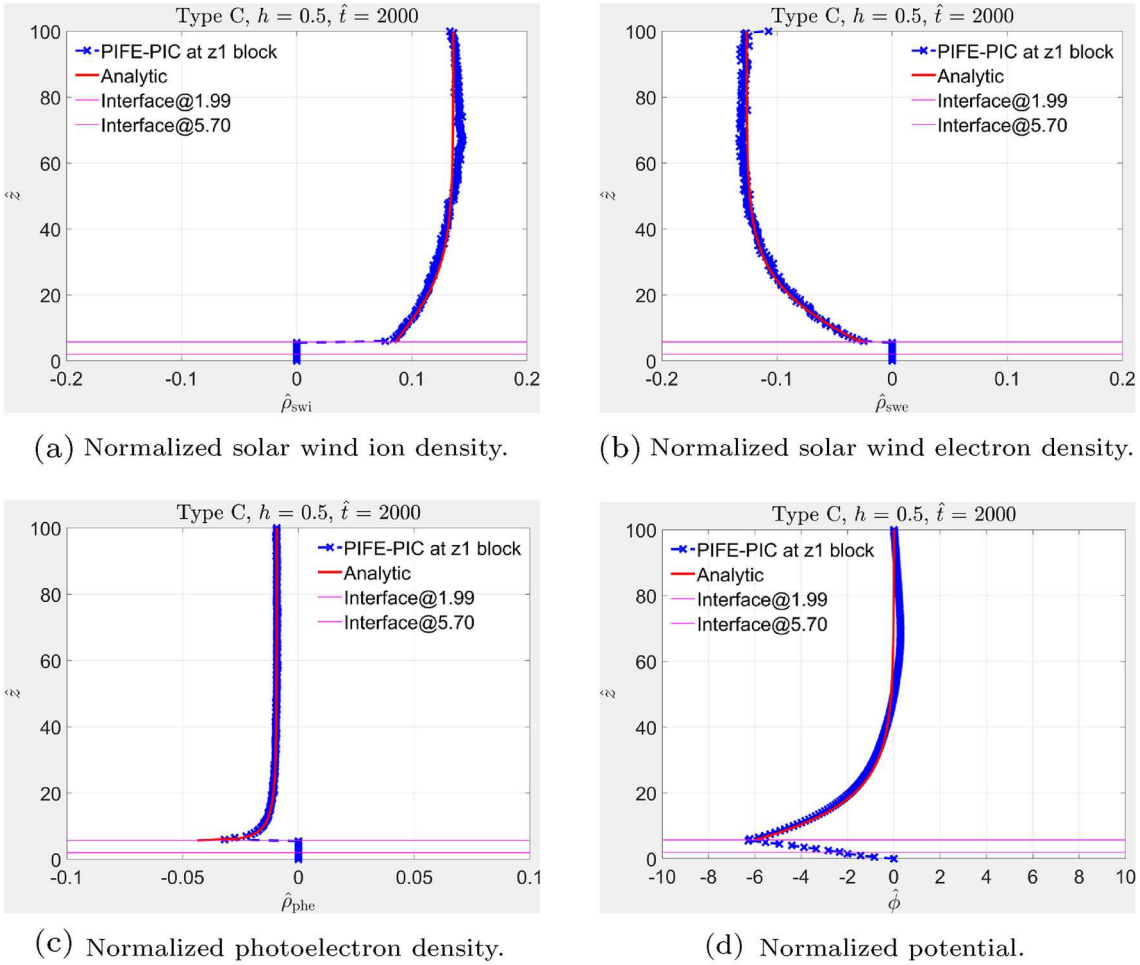
*N/A denotes "not applicable"

study the near-surface plasma environment and charging on the Moon.

Table 2 shows the detailed timer profile of PIFE-PIC on the validation case in terms of the percentage of total wall-clock time of key procedures in PIFE-PIC, namely "gather" (interpolate electric field at particle positions), "particle-push" (update particle velocities and positions), "particle-push-comm" (particle adjustment at local boundaries and communication among subdomains), "adjust-objects" (particle collection and charge deposition), "scatter" (deposit particle charge onto mesh grids), "field-solve" (solve for electric potential), "field-solve-phibc" (communication among subdomains and update of local potential boundary conditions), and "other" (including particle injection at global boundaries and input/output). These data show that for the validation case PIFE-PIC spent the majority of computing time on operations related to particles, including "gather," "particle-push," "adjust-objects," and "scatter." Since these operations scale linearly with the number of macroparticles in the simulation, we expect the weak scaling performance to depend partially on the number of macroparticles, which will be discussed in the next section.

4 Weak scaling parallel efficiency

For weak scaling, we keep the computing load (domain size and number of macroparticles) the same for all MPI processes while scaling up the problem size (both domain size and number of total macroparticles). It is noted here that for parallel cases, the solar wind travels toward the surface with an incidence angle of 10° representing the lunar terminator scenario (Fig. 3, right). Since the plasma charging problem studied here involves an interface in the domain, we only scale up the domain along x- and y-directions such that there is no domain decomposition along the z-direction and each process handles a computation domain with an interface. Five parallel cases are tested, along with the serial baseline case, with their domain decomposition configurations listed in Table 3. The domain size and number of PIC cells for the serial case are the same as the validation case above. As the problem size increases so does the number of processors; therefore, in the parallel versions of the weak scaling approach the problem size and number of PIC cells increase proportionally. For example, the $2 \times 2 \times 1$ domain decomposition (DD) configuration case uses four times the processors and has four times the number of PIC cells ($3 \times 3 \times 200 =$



(a) Normalized solar wind ion density.

(b) Normalized solar wind electron density.

(c) Normalized photoelectron density.

(d) Normalized potential.

Fig. 4 1-D photoelectron sheath validation case comparing the PIFE-PIC solutions against the semi-analytic solutions. (Colour figure online)

Table 2 Time percentage breakdown for all 40,000 PIC steps

Computing step	% of total wall-clock time (%)
Total wall-clock time	100.00
Initialization time	0.02
Main PIC Loop time	99.98
Total gather time	44.86
Total particle-push time	31.65
Total particle-push-comm time*	0.00
Total adjust-objects time	9.48
Total scatter time	7.66
Total field-solve time	6.03
Total field-solve-phibc time**	0.00
Total other time	0.31

*Included in the "particle-push time." 0% for the serial configuration
**Included in the "field-solve time." 0% for the serial configuration

Table 3 Domain decomposition configurations for weak scaling test cases

# of subdomains (MPI processes)	DD Configurations	Size of global domain (cells)
1 (serial)	$1 \times 1 \times 1$	$3 \times 3 \times 200$
4	$2 \times 2 \times 1$	$6 \times 6 \times 200$
9	$3 \times 3 \times 1$	$9 \times 9 \times 200$
16	$4 \times 4 \times 1$	$12 \times 12 \times 200$
25	$5 \times 5 \times 1$	$15 \times 15 \times 200$
36	$6 \times 6 \times 1$	$18 \times 18 \times 200$

1800 versus $6 \times 6 \times 200 = 7200$ PIC cells) as the serial version.

For all the runs in weak scaling, the convergence criteria of the field solution were kept the same as the validation case above (1×10^{-6} for the absolute residual of PCG solver) with the maximum number of PCG iteration set to 1000. It is noted here that the PCG solver of the serial case took about 300 iterations to converge, while for all the parallel cases, it took only about 40–50 iterations as shown in Fig. 5, which caused 100+% parallel efficiency for this set of weak scaling (more explanations in discussion of Tables 4, 5 and 6).

For the parallel configurations, the domain decomposition method (DDM) used in PIFE-PIC added one more level of iterations (DDM iterations) compared with the serial case. The DDM iteration numbers were set in a way that for the initial field solution, the maximum number of DDM iteration was 1000, while for each PIC step within the main PIC loop, the maximum number of DDM iteration was set to be either 10 or 5 for two different groups with same tolerance of 1×10^{-3} for the relative residual. It is noted here that to start with a better initial field for PIC steps, the 1000 DDM iterations were only needed for the initial field solution. In our observations, the field solution typically converges in less than 800 DDM iterations for the initial PIC step, about 50 DDM iterations for the second PIC step, and then stay around 10 DDM iterations starting from the third PIC step, while more macroparticles were injected to the domain at the boundary and caused perturbations. The normalized time step size was set to be 0.05, and all simulations ran for 40,000 PIC steps, which is the same as the validation case above.

The speedup is defined as $S = p \cdot T_s/T_p$, where T_s is the serial runtime and T_p is the parallel runtime on p MPI processes. The weak scaling parallel efficiency is then defined as $E = T_s/(T_p) \times 100\%$.

As shown in the validation/baseline case, the computing time largely depends on the number of macroparticles. Therefore, three sets of particle number were considered in the weak scaling test—each with (on average) 27 particles (3×3×3), 64 particles (4×4×4), and 125 particles (5×5×5) per species, per cell populated in the domain for two groups of DDM iterations:

- *Group I* at most 10 DDM iterations per main-loop PIC step (note the DDM may converge in less than 10 iterations);
- *Group II* at most 5 DDM iterations per main-loop PIC step (note the DDM may converge in less than 5 iterations).

Tables 4 to 6 list the total wall-clock time, speedup, and parallel efficiency of each case for both Group I and Group II for 27 particles (3×3×3), 64 particles (4×4×4), and 125 particles (5×5×5) per species, per cell, respectively. The timer data were taken over all 40,000 PIC steps.

A few trends are observed here:

1. It is surprising to see all parallel cases achieved 100+% parallel efficiency for weak scaling. Further investigations revealed that this was caused by the difference in the number of PCG iterations in the field-solve step (called in two loops—DDM loop and PIC loop) for the specific "unit domain" per MPI process chosen here, which is $3 \times 3 \times 200$ PIC cells. For the serial case, the PCG solver took more than 300 iterations to converge to the set criterion of 1×10^{-6} absolute residual, while for all the parallel cases, all processes took only about 40–50 PCG iterations to converge to the same criterion. Therefore, the serial case spent more iterations in the PCG solver, and thus slower, which eventually led to 100+% parallel efficiency for the parallel cases.
2. As the number of subdomains increases, the parallel efficiency does not change significantly except for the case of 4 subdomains, as a general trend for all cases of particle loading and DDM setup.
3. Overall the efficiency is insensitive to the number of particles in the domain. For all three levels of the number of particles, the efficiency trends are about the same, and they show a larger dependency on the number of maximum DDM iterations.
4. In general, for the parallel cases, the efficiency is higher for the Group I data (10 DDM iterations) compared to the Group II data (5 DDM iterations). This may seem counter-intuitive since typically fewer DDM iterations save more time in the "field-solve" step of PIFE-PIC, but in the

more-subdomain cases (with 10 DDM iterations) PIFE-PIC converges faster at the early stage because of the increased number of DDM iterations. Later, this allows the code to run fewer DDM iterations (in this case 2 DDM iterations) for the majority of the run time. On the other hand, the more-subdomain cases with 5 DDM iterations oscillate between 5 and 2 DDM iterations throughout the run time because there were not enough initial DDM iterations to converge at the early state of the run. An example of this trend can be seen in Fig. 5 on the cyan plot second from bottom labeled "# of DDM it." (only one example is shown for the case using 125 particles per species, per cell and 36 subdomains since all the other data follow the same trend). Therefore, using a larger "max DDM iterations" within the PIC step for this specific configuration led to a faster field solution. It is noted here that this trend may be limited to similar setups only - solving for steady state with preloaded particles in the domain. For other configurations, this trend may not hold and the choice of an optimal number of DDM iterations may need further investigations.

5 Application to plasma charging of a wavy lunar surface

In this section, the PIFE-PIC code is applied to simulate plasma charging of a wavy lunar surface under average solar wind conditions.

5.1 Problem description and simulation setup

The problem considered is solar wind plasma charging near the lunar surface, specifically, at the terminator region. The plasma environment is the same as the ones shown in Table 1. The geometry of the lunar surface is realized through an algebraic equation describing the surface terrain in the form of $z = z(x, y)$ where z denotes the local surface height. For the wavy lunar surface considered here, the shape is realized by the algebraic equation of (the "hat" denotes normalized)

$$\hat{z} = \hat{z}(\hat{x}, \hat{y}) = 2\cos\left(\frac{2\pi\hat{x}}{90}\right)\cos\left(\frac{2\pi\hat{y}}{60}\right) + 11.75 \tag{7}$$

as illustrated in Fig. 6.

The simulation domain has $90 \times 60 \times 60 = 324{,}000$ PIC cells (1.62 million tetrahedral FE/IFE cells). Each PIC cell is a $1.38 \times 1.38 \times 1.38$ cube (in m^3). In physical units, the domain size is approximately 124 m by 83 m by 83 m. At the $Z_{\min}$ boundary, the simulation domain includes a layer of the lunar bedrock with a thickness of $L_{\text{bedrock}} = 2.12 \times 1.38 = 2.9$ m. On top of the bedrock is a layer of dielectric regolith with a thickness of $L_{\text{regolith}} = (9.75 - 2.12) \times 1.38 = 10.5$ m. The relative permittivities of the lunar regolith layer and the bedrock are taken to be $\epsilon_{\text{regolith}} = 4$ and $\epsilon_{\text{bedrock}} = 10$, respectively [74]. 3-D domain decomposition of $6 \times 4 \times 4$ (total 96 MPI processes) is used to run the simulation (Fig. 6a).

Particles representing solar wind ions and electrons are preloaded and injected into the domain with an angle of 10° toward the surface in the X–Z plane. Particles representing photoelectrons are generated at the sunlit regions according to the local sunlight index (Fig. 6b). At the global $Z_{\max}$ domain boundary, ambient solar wind particles are injected. Particles hitting the global $Y_{\min}$ and $Y_{\max}$ boundaries are reflected due to symmetry. Since the solar wind is flowing in the positive x-direction, the $X_{\min}$ and $X_{\max}$ particle boundary conditions are periodic. Particles hitting the lunar surface are collected and their charges are accumulated to calculate surface charging.

The Dirichlet boundary condition of $\Phi = 0$ is applied at the $Z_{\max}$ boundary (the unperturbed solar wind), whereas Neumann boundary condition of zero electric field is applied on all other five domain boundaries. The maximum number for PCG iterations was set to 100 with a tolerance of 1×10^{-6} for absolute residual. (Indeed, all PCG solutions converged in about 80 iterations.) The maximum number of DDM iterations for initial field solution was set to 1000, and the maximum number of DDM iterations for each step within the main PIC loop was set to 200 with a tolerance of 1×10^{-3} for relative residual.

5.2 Surface charging results

The run took about 23 h to finish 20,000 PIC steps with the time step size of 0.05 (total simulation time till $\hat{t} = 1000$). At the steady state, the entire domain had about 150 million macroparticles. The results presented below are taken at $\hat{t} = 1000$.

Figure 7 illustrates the density contours of solar wind ions, solar wind electrons, photoelectrons, and total space charge near the surface. The solar wind ion and electron density contours show a differential density around the surface highlands. The photoelectron density contours clearly exhibit the lack of photoemission in the shadow region of the center highland. The total space charge density contours show the non-neutral regions associated with the differential density caused by the wavy surface terrain.

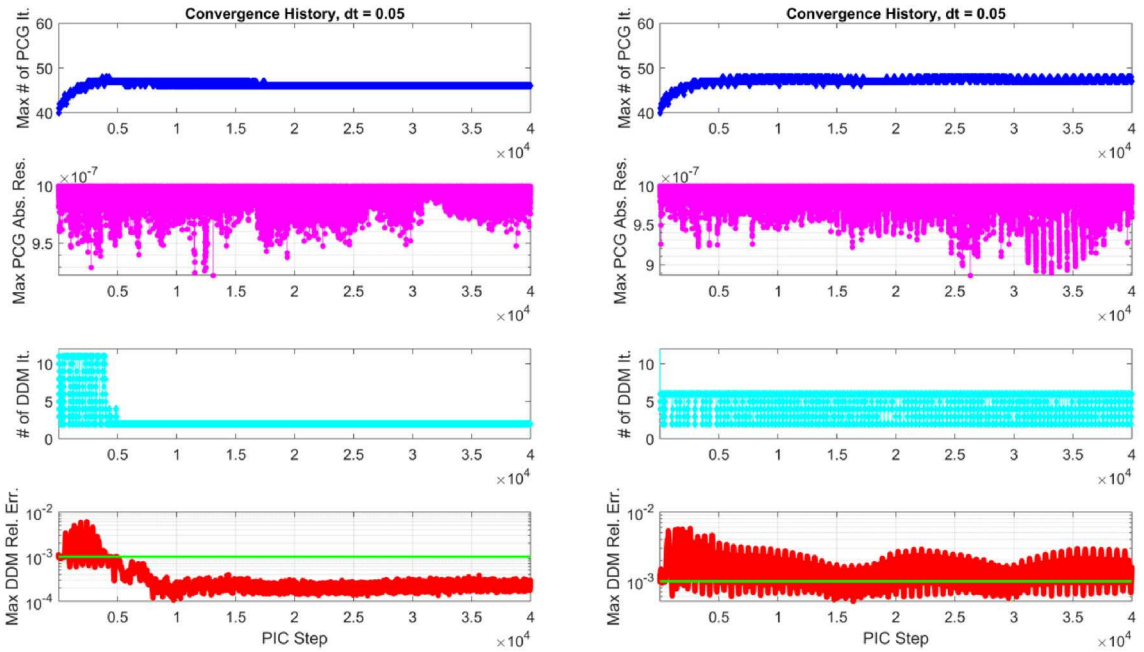
Figure 8 illustrates the potential contours of the domain and near the surface highlands. It is shown, for the average solar wind conditions considered here, the surface potential in the sunlit region of the center highland is charged to about $-2 \times 2.2 \simeq -4.4$ V, while the surface in the shadow region of the center hill is charged to about $-11 \times 2.2 \simeq -24.2$ V. It is noted as this length scale is on the order of tens of meters, the differential surface charging will affect the lunar surface

Table 4 Weak scaling test results for 27 particles per species, per cell

# of subdomains	Total time T_{I} (min)	Speedup S_{I}	Efficiency E_{I} (%)	Total time T_{II} (min)	Speedup S_{II}	Efficiency E_{II} (%)
1 (serial)	125.4	1.00	100.00	122.9	1.00	100.00
4	100.8	4.98	124.45	100.1	4.91	122.83
9	119.1	9.48	105.32	113.8	9.72	108.04
16	110.3	18.19	113.68	115.5	17.03	106.42
25	114.9	27.28	109.14	122.1	25.18	100.74
36	121.1	37.29	103.57	121.9	36.30	100.83

Table 5 Weak scaling test results for 64 particles per species, per cell

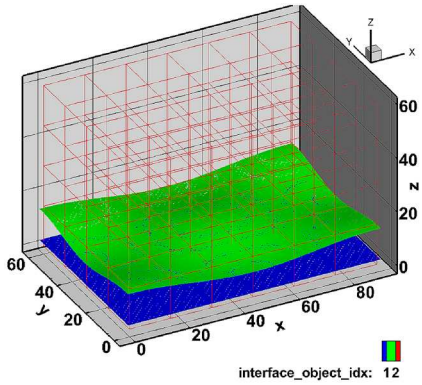
# of subdomains	Total time T_{I} (min)	Speedup S_{I}	Efficiency E_{I} (%)	Total time T_{II} (min)	Speedup S_{II}	Efficiency E_{II} (%)
1 (serial)	138.1	1.00	100.00	137.4	1.00	100.00
4	115.4	4.79	119.67	98.9	5.56	139.03
9	107.1	11.61	128.97	131.9	9.38	104.22
16	105.4	20.96	131.03	134.6	16.34	102.11
25	106.8	32.34	129.35	131.1	26.21	104.84
36	111.9	44.41	123.37	130.4	37.94	105.38



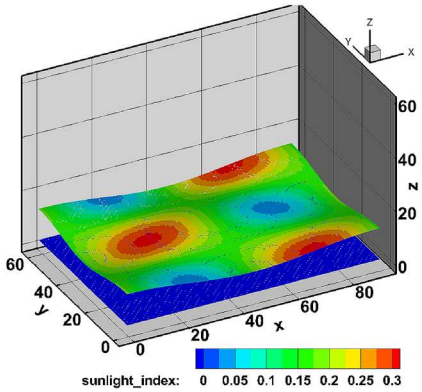
(a) Group I: 10 DDM iterations per main-loop PIC step.

(b) Group II: 5 DDM iterations per main-loop PIC step.

Fig. 5 Field convergence history of the 125 particles per species, per cell and 36 subdomain case, PCG absolute residual and DDM relative error. The green line on maximum DDM relative error plot is the DDM tolerance. (Colour figure online)

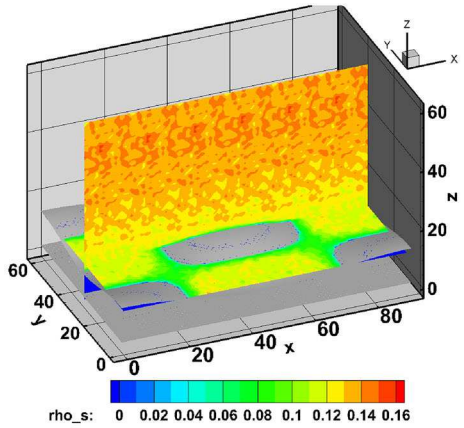


(a) The simulation domain including the lunar bedrock (below the blue layer) and the lunar regolith (between the green and blue layers). The light-red edges show the domain decomposition (6×4×4 = 96 MPI processes).

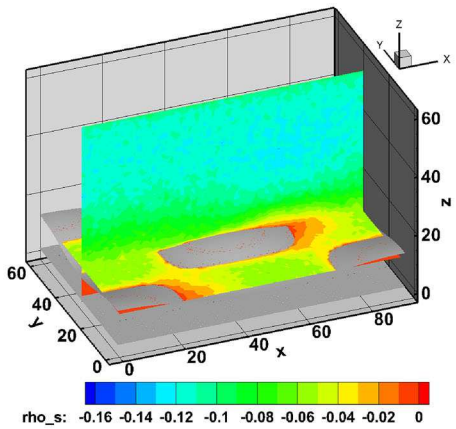


(b) The geometry of the lunar surface in PIFE-PIC. Color contours show the "sunlight index" indicating the inner product of Sun vector (10° above the ground in the X-Z plane) and local surface normal vector.

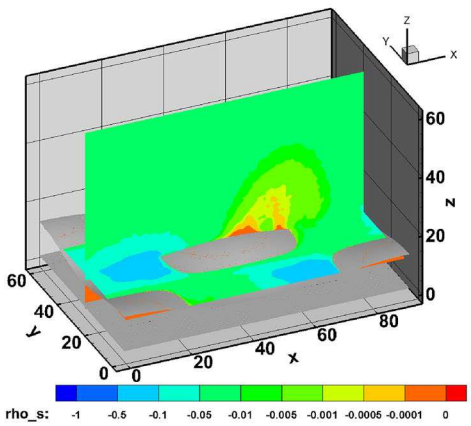
Fig. 6 The lunar surface geometry and simulation domain. (Colour figure online)



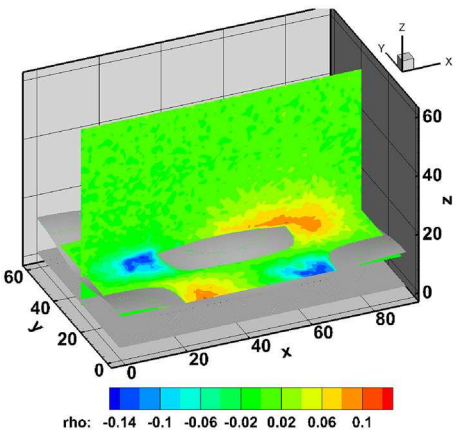
(a) Solar wind ion density contours.



(b) Solar wind electron density contours.



(c) Photoelectron density contours.



(d) Total charge density contours.

Fig. 7 Normalized density contours. For electrons, numerical values include a negative sign indicating the negative charges. The densities are normalized by 64 cm^{-3} and the spatial dimensions are normalized by 1.38 m. (Colour figure online)

Table 6 Weak scaling test results for 125 particles per species, per cell

# of subdomains	Total time T_{I} (min)	Speedup S_{I}	Efficiency E_{I} (%)	Total time T_{II} (min)	Speedup S_{II}	Efficiency E_{II} (%)
1 (serial)	160.1	1.00	100.00	159.1	1.00	100.00
4	121.0	5.30	132.38	117.7	5.41	135.13
9	122.1	11.80	131.11	146.6	9.77	108.51
16	126.6	20.24	126.49	149.2	17.07	106.67
25	130.3	30.73	122.91	145.2	27.39	109.56
36	129.4	44.53	123.70	154.9	36.98	102.73

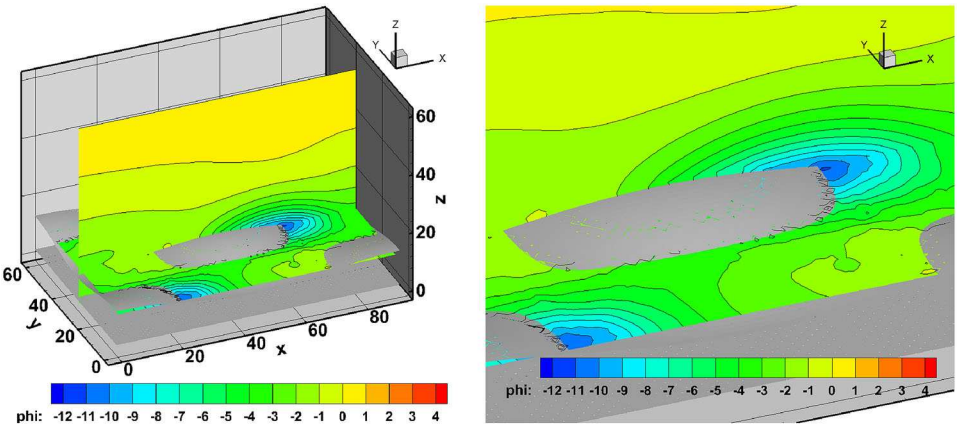


Fig. 8 Potential contours of lunar surface charging. The potential values are normalized by 2.2 V and the spatial dimensions are normalized by 1.38 m. (Colour figure online)

(a) Potential contours showing the differential charging near the lunar surface.

(b) Zoomed-in view of the potential contours near the lunar surface.

activities for exploration missions such that the risk of discharging/arcing and horizontal/vertical transport of levitated charged lunar dusts should be considered.

6 Summary and conclusion

In this study, weak scaling performance of a recently developed fully kinetic parallel immersed-finite-element particle-in-cell framework, namely PIFE-PIC, was investigated. A nominal 1-D plasma sheath problem of the vertical structure of the lunar photoelectron sheath at a low Sun elevation angle was chosen to validate PIFE-PIC against recently derived semi-analytic solutions of 1-D photoelectron sheath. The weak scaling performance test shows that the overall efficiency of PIFE-PIC is insensitive to the number of macroparticles and, counterintuitively, more domain decomposition iterations in the field-solve of PIC may lead to faster computing due to better convergence of field solutions at early stages of PIC iteration. The PIFE-PIC framework was then applied to simulate plasma charging of a wavy lunar surface with 324,000 cells and 150 million macroparticles demonstrating the capability of PIFE-PIC in resolving local-scale plasma environment near the surface of the Moon.

Acknowledgements This work was partially supported by a NASA Space Technology Graduate Research Opportunity (D.L.), NASA-Missouri Space Grant Consortium through NASA-EPSCoR-Missouri (X.H. and D.H.) and graduate scholarships (D.L.), as well as NSF through grants DMS-2005272 and DMS-2110833 (X.Z.), DMS-2111039 (X.H. and D.H.), and CBET-2132655 (D.H.). The simulations presented here were performed with computing resources provided by the Center for High Performance Computing Research at Missouri University of Science and Technology through an NSF Grant OAC-1919789.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. McKay DS, Heiken G, Basu A, Blanford G, Simon S, Reedy R, French BM, Papike J (1991) Chapter 7: The lunar regolith. In: Heiken GH, Vaniman DT, French BM (eds) Lunar sourcebook: a user's guide to the moon, pp 285–356. Cambridge University Press, Cambridge
2. Shkuratov YG, Bondarenko NV (2001) Regolith layer thickness mapping of the moon by radar and optical data. Icarus 149(2):329–338

3. Han D, Wang J, He X-M (2018) Immersed-finite-element particle-in-cell simulations of plasma charging at lunar terminator. J. Spacecr Rockets 55(6):1490–1497
4. Han D, Wang P, He X-M, Lin T, Wang J (2016) 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 321:965–980
5. Han D, Wang J, He X-M (2016) A non-homogeneous immersed-finite-element particle-in-cell method for modeling dielectric surface charging in plasmas. IEEE Trans Plasma Sci 44(8):1326–1332
6. Han D, He X, Wang JJ (2018) PIFE-PIC: a 3-D parallel immersed finite element particle-in-cell framework for plasma simulations. In: AIAA SciTech forum 2018. AIAA 2018-2196, Kissimmee, Florida
7. Han D, He X, Lund D, Zhang X (2021) PIFE-PIC: parallel immersed finite element particle-in-cell for 3-D kinetic simulations of plasma–material interactions. SIAM J Sci Comput 43(3):235–257
8. Zhao J, Wei X, Du X, He X, Han D (2021) Photoelectron sheath and plasma charging on the lunar surface: semianalytic solutions and fully-kinetic particle-in-cell simulations. IEEE Trans Plasma Sci 49(10):3036–3050
9. Burman E, Claus S, Hansbo P, Larson MG, Massing A (2015) CutFEM: discretizing geometry and partial differential equations. Int J Numer Methods Eng 104(7):472–501
10. LeVeque RJ, Li ZL (1994) The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J Numer Anal 31(4):1019–1044. https://doi.org/10.1137/0731054
11. Fries T-P, Belytschko T (2010) The extended/generalized finite element method: an overview of the method and its applications. Int J Numer Methods Eng 84(3):253–304. https://doi.org/10.1002/nme.2914
12. Zhou YC, Zhao S, Feig M, Wei GW (2006) High order matched interface and boundary method for elliptic equations with discontinuous coefficients and singular sources. J Comput Phys 213(1):1–30. https://doi.org/10.1016/j.jcp.2005.07.022
13. Babuška I, Banerjee U (2012) Stable generalized finite element method (SGFEM). Comput Methods Appl Mech Eng 201(204):91–111. https://doi.org/10.1016/j.cma.2011.09.012
14. Li Z (1998) The immersed interface method using a finite element formulation. Appl Numer Math 27(3):253–267
15. Gong Y, Li B, Li Z (2008) Immersed-interface finite-element methods for elliptic interface problems with non-homogeneous jump conditions. SIAM J Numer Anal 46:472–495
16. Guzmán J, Sánchez MA, Sarkis M (2016) Higher-order finite element methods for elliptic problems with interfaces. ESAIM Math Model Numer Anal 50(5):1561–1583
17. He X-M (2009) Bilinear immersed finite elements for interface problems. Ph.D. dissertation, Virginia Polytechnic Institute and State University
18. He X-M, Lin T, Lin Y (2008) Approximation capability of a bilinear immersed finite element space. Numer Methods Partial Differ Equ 24(5):1265–1300
19. Li Z, Lin T, Wu X (2003) New Cartesian grid methods for interface problems using the finite element formulation. Numer Math 96(1):61–98
20. Preusser T, Rumpf M, Sauter S, Schwen LO (2011) 3D composite finite elements for elliptic boundary value problems with discontinuous coefficients. SIAM J Sci Comput 33(5):2115–2143
21. Vallaghè S, Papadopoulo T (2010) A trilinear immersed finite element method for solving the electroencephalography forward problem. SIAM J Sci Comput 32(4):2379–2394
22. Guo R (2019) Design, analysis, and application of immersed finite element methods. Ph.D. dissertation, Virginia Polytechnic Institute and State University
23. Zhang X (2013) Nonconforming immersed finite element methods for interface problems. Ph.D. dissertation, Virginia Polytechnic Institute and State University
24. Lin T, Zhang X (2012) Linear and bilinear immersed finite elements for planar elasticity interface problems. J Comput Appl Math 236(18):4681–4699
25. Lin T, Sheen D, Zhang X (2013) A locking-free immersed finite element method for planar elasticity interface problems. J Comput Phys 247:228–247
26. Guo R, Lin T, Lin Y (2020) Error estimates for a partially penalized immersed finite element method for elasticity interface problems. ESAIM Math Model Numer Anal 54(1):1–24
27. Feng W, He X-M, Lin Y, Zhang X (2014) Immersed finite element method for interface problems with algebraic multigrid solver. Commun Comput Phys 15(4):1045–1067
28. He X-M, Lin T, Lin Y, Zhang X (2013) Immersed finite element methods for parabolic equations with moving interface. Numer Methods Partial Differ Equ 29(2):619–646
29. Adjerid S, Moon K (2019) An immersed discontinuous Galerkin method for acoustic wave propagation in inhomogeneous media. SIAM J Sci Comput 41(1):139–162
30. Adjerid S, Lin T, Zhuang Q (2020) Error estimates for an immersed finite element method for second order hyperbolic equations in inhomogeneous media. J Sci Comput 84(2):35
31. Bai J, Cao Y, He X-M, Liu H, Yang X (2018) Modeling and an immersed finite element method for an interface wave equation. Comput Math Appl 76(7):1625–1638
32. Adjerid S, Chaabane N, Lin T, Yue P (2019) An immersed discontinuous finite element method for the Stokes problem with a moving interface. J Comput Appl Math 362:540–559
33. Jones D, Zhang X (2021) A class of nonconforming immersed finite element methods for Stokes interface problems. J Comput Appl Math 392:113493
34. Guo R, Lin T (2019) A higher degree immersed finite element method based on a Cauchy extension for elliptic interface problems. SIAM J Numer Anal 57(4):1545–1573
35. He X-M, Lin T, Lin Y (2012) The convergence of the bilinear and linear immersed finite element solutions to interface problems. Numer Methods Partial Differ Equ 28(1):312–330
36. Lin T, Lin Y, Zhang X (2015) Partially penalized immersed finite element methods for elliptic interface problems. SIAM J Numer Anal 53(2):1121–1144
37. He X-M, Lin T, Lin Y (2010) Interior penalty bilinear IFE discontinuous Galerkin methods for elliptic equations with discontinuous coefficient, dedicated to David Russell's 70th birthday. J Syst Sci Complex 23(3):467–483
38. He X-M, Lin T, Lin Y (2014) A selective immersed discontinuous Galerkin method for elliptic interface problems. Math Methods Appl Sci 37(7):983–1002
39. Lin T, Yang Q, Zhang X (2015) A priori error estimates for some discontinuous Galerkin immersed finite element methods. J Sci Comput 65(3):875–894
40. Ewing RE, Li Z, Lin T, Lin Y (1999) The immersed finite volume element methods for the elliptic interface problems. Modelling '98 (Prague). Math Comput Simul 50(1–4):63–76
41. Cao W, Zhang X, Zhang Z, Zou Q (2017) Superconvergence of immersed finite volume methods for one-dimensional interface problems. J Sci Comput 73(2–3):543–565
42. He X-M, Lin T, Lin Y (2009) A bilinear immersed finite volume element method for the diffusion equation with discontinuous coefficients, dedicated to Richard E. Ewing on the occasion of his 60th birthday. Commun Comput Phys 6(1):185–202
43. Lin T, Sheen D, Zhang X (2019) A nonconforming immersed finite element method for elliptic interface problems. J Sci Comput 79(1):442–463

44. Bai J, Cao Y, Chu Y, Zhang X (2018) An improved immersed finite element particle-in-cell method for plasma simulation. Comput Math Appl 75(6):1887–1899
45. Bai J, Cao Y, He X-M, Peng E (2021) An implicit particle-in-cell model based on anisotropic immersed-finite-element method. Comput Phys Commun 261:107655
46. Cao Y, Chu Y, Zhang X, Zhang X (2016) Immersed finite element methods for unbounded interface problems with periodic structures. J Comput Appl Math 307:72–81
47. Chu Y, Cao Y, He X-M, Luo M (2011) Asymptotic boundary conditions with immersed finite elements for interface magnetostatic/electrostatic field problems with open boundary. Comput Phys Commun 182(11):2331–2338
48. Chu Y, Han D, Cao Y, He X-M, Wang J (2017) An immersed-finite-element particle-in-cell simulation tool for plasma surface interaction. Int J Numer Anal Model 14(2):175–200
49. Wang J, He X-M, Cao Y (2007) Modeling spacecraft charging and charged dust particle interactions on lunar surface. In: Proceedings of the 10th spacecraft charging technology conference, Biarritz, France
50. Cao Y, Chu Y, He X-M, Lin T (2015) An iterative immersed finite element method for an electric potential interface problem based on given surface electric quantity. J Comput Phys 281:82–95
51. He X-M, Lin T, Lin Y (2011) Immersed finite element methods for elliptic interface problems with non-homogeneous jump conditions. Int J Numer Anal Model 8(2):284–301
52. Lu C, Wan J, Cao Y, He X-M (2020) A fully decoupled iterative method with three-dimensional anisotropic immersed finite elements for Kaufman-type discharge problems. Comput Methods Appl Mech Eng 372:113345
53. Lu C, Yang Z, Bai J, Cao Y, He X-M (2020) Three-dimensional immersed finite element method for anisotropic magnetostatic/electrostatic interface problems with non-homogeneous flux jump. Int J Numer Methods Eng 121(10):2107–2127
54. Cao H, Cao Y, Chu Y, He X-M, Lin T (2018) A Huygens immersed-finite-element particle-in-cell method for modeling plasma-surface interactions with moving interface. Commun Nonlinear Sci Numer Simul 59:132–148
55. Jian H, Chu Y, Cao H, Cao Y, He X-M, Xia G (2015) Three-dimensional IFE-PIC numerical simulation of background pressure's effect on accelerator grid impingement current for ion optics. Vacuum 116:130–138
56. Kafafy R, Wang J (2005) Whole subscale ion optics simulation: direct ion impingement and electron backstreaming. In: 41st AIAA/ASME/SAE/ASEE joint propulsion conference and exhibit. AIAA 2005-3691, Tucson, Arizona
57. Kafafy R, Wang J (2007) Whole ion optics gridlet simulations using a hybrid-grid immersed-finite-element particle-in-cell code. J Propuls Power 23(1):59–68
58. Kafafy RI, Wang J (2006) A hybrid grid immersed finite element particle-in-cell algorithm for modeling spacecraft–plasma interactions. IEEE Trans Plasma Sci 34(5):2114–2124
59. Wang J, Cao Y, Kafafy R, Pierru J, Decyk VK (2006) Simulations of ion thruster plume–spacecraft interactions on parallel supercomputer. IEEE Trans Plasma Sci 34(5):2148–2158
60. Han D, Wang JJ (2013) Simulations of ion thruster plume contamination with a whole grid sputtered Mo source model. In: 49th AIAA/ASME/SAE/ASEE joint propulsion conference and exhibit. AIAA 2013-3888, San Jose, California
61. Depew D, Han D, Wang J, He X-M, Lin T (2014) Immersed-finite-element particle-in-cell simulations of lunar surface charging. In: Proceedings of the 13th spacecraft charging technology conference, Pasadena, California, June 23–27
62. Han D (2015) Particle-in-cell simulations of plasma interactions with asteroidal and lunar surfaces. Ph.D. thesis, University of Southern California
63. Han D, Wang J (2019) 3-D fully-kinetic particle-in-cell simulations of small asteroid charging in the solar wind. IEEE Trans Plasma Sci 47(8):3682–3688
64. Yu W, Han D, Wang J (2019) Numerical simulations of dust dynamics around small asteroids. IEEE Trans Plasma Sci 47(8):3724–3730
65. Yu W, Wang JJ, Han D (2016) Numerical modeling of dust dynamics around small asteroids. In: AIAA SPACE forum 2016. AIAA 2016-5447, Long Beach, California
66. Quarteroni A, Valli A (1999) Domain decomposition methods for partial differential equations. Numerical mathematics and scientific computation. Oxford Science Publications, New York
67. Smith B, Bjørstad P, Gropp W (1996) Domain decomposition: parallel multilevel methods for elliptic partial differential equations. Cambridge University Press, Cambridge
68. Guo R, Lin T (2020) An immersed finite element method for elliptic interface problems in three dimensions. J Comput Phys 414(1):109478
69. Guo R, Zhang X (2021) Solving three-dimensional interface problems with immersed finite elements: a-priori error analysis. J Comput Phys 441:110445
70. Kafafy R, Lin T, Lin Y, Wang J (2005) Three-dimensional immersed finite element methods for electric field simulation in composite materials. Int J Numer Methods Eng 64(7):940–972
71. Fu JHM (1971) Surface potential of a photoemitting plate. J Geophys Res (1896–1977) 76(10):2506–2509
72. Nitter T, Havnes O, Melandsø F (1998) Levitation and dynamics of charged dust in the photoelectron sheath above surfaces in space. J Geophys Res Sp Phys 103(A4):6605–6620
73. Poppe AR (2011) Modeling, theoretical and observational studies of the lunar photoelectron sheath. Ph.D. thesis, University of Colorado, Boulder
74. Heiken GH, Vaniman DT, French BM (1991) Lunar sourcebook: a user's guide to the moon. Cambridge University Press, Cambridge