Journal of Computational Physics 363 (2018) 178–199

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A parallel electrostatic Particle-in-Cell method on unstructured tetrahedral grids for large-scale bounded collisionless plasma simulations

Sergey N. Averkin, Nikolaos A. Gatsonis *

Aerospace Engineering Program, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609, USA

ARTICLE INFO

Article history:
Received 16 May 2017
Received in revised form 4 February 2018
Accepted 7 February 2018
Available online 21 February 2018

Keywords:
Bounded collisionless plasma simulation
Parallel Particle-In-Cell (PIC)
Unstructured PIC
Finite volume Multi-Point Flux
Approximation Poisson solver
Parallel GMRES

ABSTRACT

An unstructured electrostatic Particle-In-Cell (EUPIC) method is developed on arbitrary tetrahedral grids for simulation of plasmas bounded by arbitrary geometries. The electric potential in EUPIC is obtained on cell vertices from a finite volume Multi-Point Flux Approximation of Gauss' law using the indirect dual cell with Dirichlet, Neumann and external circuit boundary conditions. The resulting matrix equation for the nodal potential is solved with a restarted generalized minimal residual method (GMRES) and an ILU(0) preconditioner algorithm, parallelized using a combination of node coloring and level scheduling approaches. The electric field on vertices is obtained using the gradient theorem applied to the indirect dual cell. The algorithms for injection, particle loading, particle motion, and particle tracking are parallelized for unstructured tetrahedral grids. The algorithms for the potential solver, electric field evaluation, loading, scatter-gather algorithms are verified using analytic solutions for test cases subject to Laplace and Poisson equations. Grid sensitivity analysis examines the L^2 and L^∞ norms of the relative error in potential, field, and charge density as a function of edge-averaged and volume-averaged cell size. Analysis shows second order of convergence for the potential and first order of convergence for the electric field and charge density. Temporal sensitivity analysis is performed and the momentum and energy conservation properties of the particle integrators in EUPIC are examined. The effects of cell size and timestep on heating, slowing-down and the deflection times are quantified. The heating, slowing-down and the deflection times are found to be almost linearly dependent on number of particles per cell. EUPIC simulations of current collection by cylindrical Langmuir probes in collisionless plasmas show good comparison with previous experimentally validated numerical results. These simulations were also used in a parallelization efficiency investigation. Results show that the EUPIC has efficiency of more than 80% when the simulation is performed on a single CPU from a non-uniform memory access node and the efficiency is decreasing as the number of threads further increases. The EUPIC is applied to the simulation of the multi-species plasma flow over a geometrically complex CubeSat in Low Earth Orbit. The EUPIC potential and flowfield distribution around the CubeSat exhibit features that are consistent with previous simulations over simpler geometrical bodies.

* Corresponding author.
E-mail address: gatsonis@wpi.edu (N.A. Gatsonis).

https://doi.org/10.1016/j.jcp.2018.02.011
0021-9991/

1. Introduction

Particle-in-Cell (PIC) methods using unstructured grids and parallelization offer flexibility to perform large-scale computations of collisionless plasmas in domains with complex geometries. The algorithmic and mathematical issues for parallelized unstructured PIC implementations involve all aspects of a PIC cycle: gather/scatter, particle search and motion, potential and electric field evaluation, boundary conditions in bounded and unbounded plasmas, and quality of simulation due to artificial collisions and heating. There have been few implementations of electrostatic and electromagnetic PIC on unstructured grids and an even smaller number of parallelized ones. A three-dimensional parallel, electromagnetic PIC method on non-uniform hexahedral grids was developed by Wang et al. [1,2]. The method is based on a finite volume formulation with hexahedral cells that are connected with cubic cells, distorted to fit the complex geometries. Each hexahedral is mapped one to one to a unit cube in the logical Cartesian space. The gather/scatter procedures are performed in the Cartesian space using a charge conserving weighting scheme by Villasenor and Buneman [3]. The code has been applied to ion beam neutralization [4]. Wu et al. [5] developed a 3d finite element, parallelized, electrostatic code using unstructured tetrahedral grids with dynamic domain decomposition. Petillo et al. [6] developed a finite element electromagnetic PIC code on structured and unstructured grids and applied to electron guns. A three-dimensional PIC code on unstructured tetrahedral grids coupled with a finite element electrostatic solver and a frequency-domain electromagnetic solver was developed by Pavarin et al. [7] and applied to the simulation of a cylindrical cusped-plasma accelerator [8]. Alternative approaches addressing complex geometries without using unstructured meshes include conformal mapping [9] and adaptive mesh refinement [10]. Unstructured PIC simulations present a potential for increased artificial collisions and numerical heating due to the exerted self-force on particles [11]. Bettencourt [12] suggested an algorithm that allows controlling the amount of the self-force. Gatsonis and Spirkin [13] presented the mathematical formulation and implementation of an electrostatic PIC method on unstructured 3d Delaunay–Voronoi tetrahedral grids (UPIC3dE). The duality of the Delaunay–Voronoi grid was used in [13] effectively in the gather/scatter, potential solver, particle mover, and sampling step of the UPIC3dE cycle but imposed restrictions due to the required quality of the Delaunay discretization in 3D.

To take advantage of available general-purpose tetrahedral grid generators and multi-platform shared-memory multiprocessing computers including GPUs, this work presents a new mathematical formulation of a parallelized electrostatic PIC method on unstructured tetrahedral grids (EUPIC). All algorithms of the EUPIC are parallelized and implemented using OpenMP methodology allowing large-scale plasma computations with complex geometrical domains on multiprocessors. The grid structure used in EUPIC involves the tetrahedral cells, which scale with the local Debye length and the indirect dual cells formed by connecting the centroids of each adjoining face to the midpoints of the edges shared and then connecting the centroids of the faces to the centroids of the tetrahedra to which these faces belong. Charge assignment to the vertices of the tetrahedra (or nodes) and electric-field weighting to the particle follows [13]. The evaluation of the electric potential follows the finite volume formulation of the integral Gauss law in [13] but is performed using the indirect dual cell as the Gaussian surface. The potential is assumed to vary linearly within a cell, which makes our formulation consistent with the Multi-Point Flux Approximation (MPFA) family of methods [14]. The potential on conductors driven by external circuits is evaluated by a finite volume MPFA of the integral Gauss law, the charge conservation law and Kirchhoff's lumped circuit law. Unlike previous approaches [13,15,16] during an EUPIC iteration a single extended system of algebraic equations is solved providing the potential on all nodes of the domain, including externally driven, Dirichlet and Neumann nodes. The solution is obtained by the restarted generalized minimal residual method (GMRES) solver with an incomplete LU preconditioner with zero fill-ins (ILU(0)) following [17]. GMRES is parallelized in EUPIC with OpenMP using a combination of node-coloring and level-scheduling approaches. The nodal electric field is evaluated by a finite volume MPFA of the integral definition for the electric field [13]. For a Dirichlet or driven-circuit boundary node the electric field is corrected by use of the nodal surface charge, which is evaluated by a finite volume MPFA of Gauss' Law. The particle integrator in EUPIC follows Buneman's time-centered leapfrog formulation [18]. The particle search-locate algorithm is performed using an optimized version of an algorithm developed for Direct Simulation Monte Carlo method on tetrahedra [19]. Particle injection from surfaces in EUPIC follows [13]. Periodic boundary conditions are also implemented in EUPIC for a number of pairs of periodic surfaces by mirroring the grids and translating particles between two periodic surfaces. Nodal macroscopic properties are evaluated using a super-cell that consist of all tetrahedral cells surrounding a node taking into account particles with different weights. The methods and algorithms in EUPIC are validated and verified with an extensive set of test cases. Grid sensitivity analysis is performed to identify the order of accuracy for the potential and electric field evaluation. The effects of particles/cell, grid scaling, and timestep on the numerical heating, the slowing-down time, and the deflection time are investigated by performing simulations of fully ionized electron–positive ion plasmas in a grounded box with periodic boundary conditions. Laframboise's results [20] on current collection by cylindrical Langmuir probes in collisionless plasmas are used for verification of EUPIC simulations in the orbital motion limited and thin sheath regimes. Finally, EUPIC is used for a simulation of the plasma flow around a CubeSat in Low Earth Orbit.

2. Electrostatic unstructured particle in cell in 3D (EUPIC) methodology

The EUPIC method solves the electrostatic Vlasov–Poisson system for a multi-species fully ionized plasma [21,22]. The single-particle distribution function for species s particles, gives the average number of particles in a volume $d^3rd^3v \equiv$

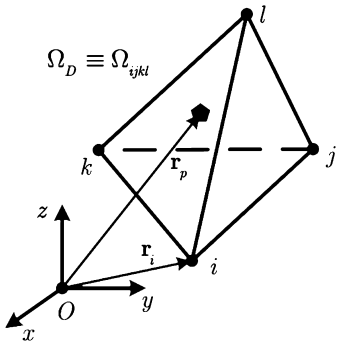


Fig. 1. The tetrahedron Ω_D with global indexing in a domain Ω.

$d\mathbf{r}d\mathbf{v}$ of the phase-space centered at a point $(\mathbf{r}, \mathbf{v})$ as $f_s(\mathbf{r}, \mathbf{v}, t)d^3rd^3v = d^6N_s(t)$ and the local number density as $n_s(r,t) = \int f_s(\mathbf{r}, \mathbf{v}, t)d^3v$. The equation for the distribution function is

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \frac{\partial f_s}{\partial \mathbf{r}} + \frac{q_s}{m_s}\left[\mathbf{E}(\mathbf{r}, t) + \mathbf{v} \times \mathbf{B}_{\text{ext}}(\mathbf{r}, t)\right] \cdot \frac{\partial f_s}{\partial \mathbf{v}} = 0. \tag{1}$$

In the electrostatic formulation induced magnetic fields due to the particle motion are neglected, and Faraday's Law implies that electric fields are irrotational given by the electrostatic potential $\phi(\mathbf{r}, t)$

$$\mathbf{E}(\mathbf{r}, t) = -\nabla\phi(\mathbf{r}, t). \tag{2}$$

The self-consistent electric field (and potential) is due to the smoothed distribution (internal) $\rho_{\text{int}}(t)$ and external distribution of charges $\rho_{\text{ext}}(t)$, and is given by Maxwell's equations that reduce to the integral Gauss's law,

$$\oiint_A \mathbf{E} \cdot d\mathbf{A} = -\oiint_A \nabla\phi \cdot d\mathbf{A} = \frac{1}{\varepsilon_0}\iiint_V \rho dV \tag{3}$$

or to the differential Poisson's equation

$$\varepsilon_0 \nabla \cdot \mathbf{E}(\mathbf{r}, t) = \varepsilon_0 \nabla \cdot \left(\mathbf{E}_{\text{ext}}(\mathbf{r}, t) + \mathbf{E}_{\text{int}}(\mathbf{r}, t)\right) = \sum_s q_s \int f_s(\mathbf{r}, \mathbf{v}, t)d^3v - \rho_{ext}(\mathbf{r}, t). \tag{4}$$

In the Vlasov–Poisson system the single-particle trajectories in the presence of an electrostatic and an external magnetic field are

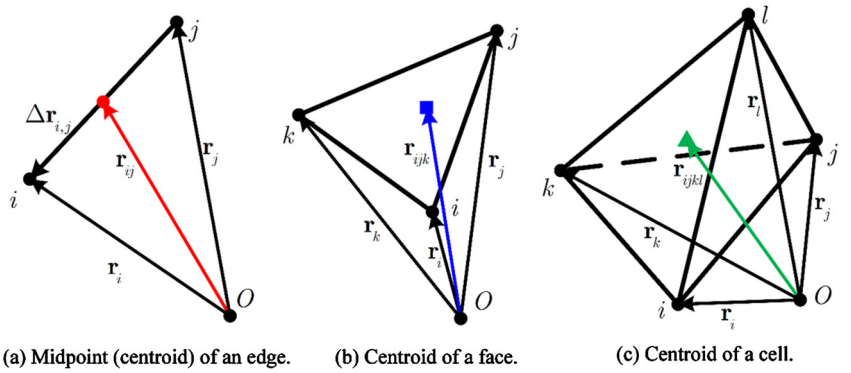
$$\begin{aligned} &\frac{d\mathbf{r}_s(t)}{dt} = \mathbf{v}_s(t), \\ &m_s\frac{d\mathbf{v}_s(t)}{dt} = \frac{q_s}{m_s}\left[\mathbf{E}\big(\mathbf{r}_s(t), t\big) + \mathbf{v}_s \times \mathbf{B}_{\text{ext}}\big(\mathbf{r}_s(t), t\big)\right]. \end{aligned} \tag{5}$$

2.1. Grid generation and data structure

The EUPIC method uses an unstructured three-dimensional mesh with tetrahedral cells obtained through a grid generator such as GMSH [23]. The EUPIC method requires local data structures for efficient implementation of particle motion, electric field evaluation and the sampling of macroscopic properties. Grid parameters are stored in data structures that maintain node position, node connectivity, face sharing and cell nearest-neighbor information similar to [13]. A tetrahedral cell is illustrated in Fig. 1 and assigned to a global cell index $D = 1, \ldots, G_D$, where G_D is the number of cells in the domain. The vertices of the cells are designated as the nodes of the domain, each assigned to a global node index $d = 1, \ldots, G_d$, where G_d is the number of nodes in the domain. The triangular faces A_σ of the tetrahedral cells with area $|A_\sigma|$ are assigned to a global face index $\sigma = 1, \ldots, G_\sigma$, where G_σ is the number of faces in the domain. The four nodes associated with a cell D form a unique index set $\{i, j, k, l\}$ through an association $D \rightarrow \{i, j, k, l\}$, where the ordered set $\{i, j, k, l\} \in \{1, \ldots, G_d\}$. The numbering of the nodes corresponds to the right orientation of the tetrahedron. A node $\{i\}$ has coordinates $\mathbf{r}_i = (x_i, y_i, z_i)$ and the volume of the cell Ω_D denoted by $|\Omega_D|$ is

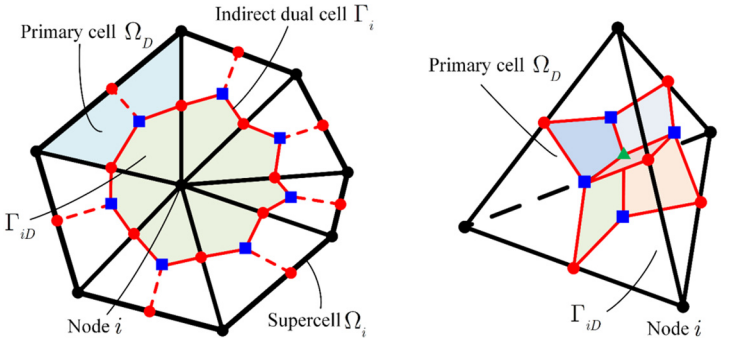
$$|\Omega_D| \equiv |\Omega_{ijkl}| = \Delta\mathbf{r}_{l,i} \cdot (\Delta\mathbf{r}_{i,j} \times \Delta\mathbf{r}_{i,k})/6, \tag{6}$$

where $\Delta\mathbf{r}_{i,j} = \mathbf{r}_i - \mathbf{r}_j$.



(a) Midpoint (centroid) of an edge. (b) Centroid of a face. (c) Centroid of a cell.

Fig. 2. Notation used for the centroids in the tetrahedral mesh.



(a) Primary Ω_D and corresponding indirect dual cell Γ_i^D and supercell Ω_i in 2D

(b) Primary Ω_D and corresponding part of indirect dual Γ_{iD} cells in 3D

Fig. 3. Dual cells for a given primary cell Ω_D in 2D and 3D.

Centroids used in various aspects of an EUPIC cycle are depicted in Fig. 2. The centroid (or midpoint) of the edge connecting nodes i and j, $\mathbf{r}_{ij}$, the centroid of the face surrounded by nodes i, j, k, $\mathbf{r}_{ijk}$, and the centroid of the tetrahedron Ω_{ijkl}, $\mathbf{r}_{ijkl}$, are defined respectively as

$$\begin{aligned}\mathbf{r}_{ij} &= \frac{1}{2}(\mathbf{r}_i + \mathbf{r}_j),\\ \mathbf{r}_{ijk} &= \frac{1}{3}(\mathbf{r}_i + \mathbf{r}_j + \mathbf{r}_k),\\ \mathbf{r}_{ijkl} &= \frac{1}{4}(\mathbf{r}_i + \mathbf{r}_j + \mathbf{r}_k + \mathbf{r}_l).\end{aligned} \tag{7}$$

A triangular face of the cell $\Omega_{ijkl}(\Omega_D)$ with nodes i, j, k opposite to the node l is denoted by A_{Dl}. The unit outward normal to this face with respect to the cell Ω_D is $\mathbf{n}_{Dl}$.

The indirect dual cell used in Poisson's solver is shown in Fig. 3(a) for a 2D configuration. For a node i it is formed by connecting the centroids of each adjoining face to the midpoints of the edges shared by i. The partial indirect dual cell for a 3D configuration around node i is shown in Fig. 3b. It is formed by connecting the centroids of each adjoining face to the midpoints of the edges shared by i, then connecting the centroids of the faces to the centroids of the tetrahedra to which these faces belong. This construction splits each tetrahedron $\Omega_D(\Omega_{ijkl})$ into four hexahedra. The hexahedron containing a node i is denoted by Γ_{iD}. The dual cell for a node i containing all such hexahedra is denoted by Γ_i and is given by

$$\Gamma_i = \bigcup_{\substack{D\\ \Omega_D \ni i}} \Gamma_{iD}. \tag{8}$$

The face of the dual cell Γ_{iD} adjacent to the edge connecting the nodes i and k is denoted by $\hat{A}_{Dik}$ and can be written as

$$\hat{A}_{Dik} = \Gamma_{iD} \cap \Gamma_{kD}. \tag{9}$$

The area of the face $\hat{A}_{Dik}$ is denoted by $|\hat{A}_{Dik}|$. The outward unit normal to this face for the cell Γ_{iD} is denoted by $\hat{\mathbf{n}}_{Dik}$. The relation between unit outward normals for two adjacent cells is

$$\hat{\mathbf{n}}_{Dik} = -\hat{\mathbf{n}}_{Dki}, \tag{10}$$

where $\hat{\mathbf{n}}_{Dki}$ is the outward normal for the face $\hat{A}_{Dki} \equiv \hat{A}_{Dik}$ of the cell Γ_{kD}.

The data structure implemented in EUPIC allows evaluation of information on neighboring cells during particle motion without any additional operations as discussed in Sec. 2.6.

2.2. Charge assignment, nodal number density and nodal charge density

The particle loading and injection procedures in EUPIC follow those in [13]. The charge assignment functions used in EUPIC follow also [13] and satisfy charge conservation and first-order constraint. With the Nearest Grid Point (NGP) scheme, the charge from the particle at position $\mathbf{r}_p \in \Omega_D(\Omega_{ijkl})$ is assigned to the closest node from the cell Ω_D. Thus, the weight function $W_{mD}(\mathbf{r}_p)$ assigned to the node $m \in \Omega_D$ for a particle at position $\mathbf{r}_p \in \Omega_D(\Omega_{ijkl})$ is

$$W_{mD}(\mathbf{r}_p) = \begin{cases} 1, & \min_{n\in\Omega_D} |\mathbf{r}_n - \mathbf{r}_p| = |\mathbf{r}_m - \mathbf{r}_p| \\ 0, & \min_{n\in\Omega_D} |\mathbf{r}_n - \mathbf{r}_p| \neq |\mathbf{r}_m - \mathbf{r}_p| \end{cases} \quad m = i, j, k, l. \tag{11}$$

With the Cloud In Cell (CIC) scheme, the weight functions are

$$W_{mD}(\mathbf{r}_p) = (1/3)\Delta\mathbf{r}_{np} \cdot \mathbf{n}_{Dm}|A_{Dm}|/\Omega_D \quad m = i, j, k, l;\ n \in A_{Dm}. \tag{12}$$

Using weights given by Eqs. (11) or (12) the species s charge at a node i is

$$Q_i^s = q_s \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} W_{iD}(\mathbf{r}_p) F_{Wp}, \tag{13}$$

where F_{Wp} is the particle weight of particle p, i.e. the number of real molecules represented by a single computational particle p, N_p^s is the number of particles of species s available in the computation domain. With the nodal charges known the species s charges density at a node i is obtained using the volume of the indirect dual cell Γ_i shown in Fig. 3(a), as

$$\rho_s(x_i, y_i, z_i) \equiv \rho_s^i = Q_s^i/|\Gamma_i|. \tag{14}$$

2.3. Electric potential at nodes

The integral Gauss law given by Eq. (3) using as the Gaussian surface the dual cell Γ_i surrounding a node $i = 1, \ldots, G_d$ shown in Fig. 3a for 2D configuration yields

$$\sum_{D=1}^{G_D(i)} \sum_{\substack{k\in\Omega_D \\ k\neq i}} \iint_{\hat{A}_{Dik}} \nabla\phi \cdot d\mathbf{A} = -\frac{Q_i}{\varepsilon_0}, \tag{15}$$

where $G_D(i)$ denotes the number of cells which share the same node i. Assuming furthermore a linear potential variation inside each tetrahedron $\Omega_D \cap \Gamma_i \neq \emptyset$, the electric field (or gradient) in Eq. (15) is constant in every cell Ω_D and can be found from the following system of linear equations

$$\begin{aligned} (\nabla\phi)_D \cdot \Delta\mathbf{r}_{i,j} &= \phi_i - \phi_j, \\ (\nabla\phi)_D \cdot \Delta\mathbf{r}_{i,k} &= \phi_i - \phi_k, \\ (\nabla\phi)_D \cdot \Delta\mathbf{r}_{i,l} &= \phi_i - \phi_l, \end{aligned} \tag{16}$$

whose solution for the electric field for a positively oriented tetrahedron is written as

$$\mathbf{E}_D = -(\nabla\phi)_D = \frac{1}{3|\Omega_D|}\left(\phi_i \mathbf{n}_{Di}|A_{Di}| + \phi_j \mathbf{n}_{Dj}|A_{Dj}| + \phi_k \mathbf{n}_{Dk}|A_{Dk}| + \phi_l \mathbf{n}_{Dl}|A_{Dl}|\right). \tag{17}$$

The discrete form of Eq. (15) can be written as

$$\sum_{D=1}^{G_D(i)} (\nabla\phi)_D \cdot \sum_{\substack{k\in\Omega_D \\ k\neq i}} \hat{\mathbf{n}}_{Dik}|\hat{A}_{Dik}| = -\frac{Q_i}{\varepsilon_0}. \tag{18}$$

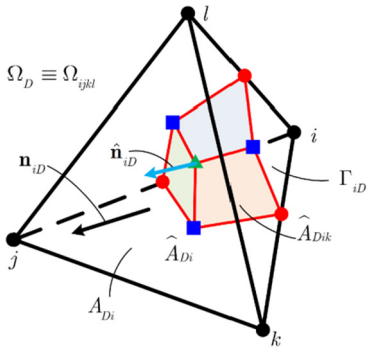


Fig. 4. Γ_{iD} is the part Gaussian surface Γ_i used in the evaluation of the electric potential.

The areas and corresponding normals in Eq. (18) as shown in Fig. 4 are

$$\hat{\mathbf{n}}_{Di}|\hat{A}_{Di}| = \sum_{\substack{k\in\Omega_D \\ k\neq i}} \hat{\mathbf{n}}_{Dik}|\hat{A}_{Dik}| = \hat{\mathbf{n}}_{Dij}|\hat{A}_{Dij}| + \hat{\mathbf{n}}_{Dik}|\hat{A}_{Dik}| + \hat{\mathbf{n}}_{Dil}|\hat{A}_{Dil}| \tag{19}$$

and can be simplified to

$$\hat{\mathbf{n}}_{Di}|\hat{A}_{Di}| = \frac{1}{3}\mathbf{n}_{Di}|A_{Di}|. \tag{20}$$

Substituting Eq. (17) and Eq. (20) into Eq. (18), it becomes

$$\sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk}|A_{Dk}| \cdot \mathbf{n}_{Di}|A_{Di}| = \frac{Q_i}{\varepsilon_0}. \tag{21}$$

Applying Eq. (21) to all the interior nodes G_d in the domain, a system of $G_d \times G_d$ linear equations for the unknown nodal potentials is obtained

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1G_d} \\ R_{21} & R_{22} & R_{23} & \cdots & R_{2G_d} \\ R_{31} & R_{32} & R_{33} & \cdots & R_{3G_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{G_d1} & R_{G_d2} & R_{G_d3} & \cdots & R_{G_dG_d} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{G_d} \end{pmatrix} = \begin{pmatrix} Q_1/\varepsilon_0 \\ Q_2/\varepsilon_0 \\ Q_3/\varepsilon_0 \\ \vdots \\ Q_{G_d}/\varepsilon_0 \end{pmatrix}, \tag{22}$$

where

$$R_{ij} = \begin{cases} \sum_{D=1}^{G_D(i)} \frac{|A_{Di}|^2}{9|\Omega_D|}, & i = j, \\ \sum_{D=1}^{G_D(i)} \frac{\mathbf{n}_{Dk}\cdot\mathbf{n}_{Di}|A_{Dk}||A_{Di}|}{9|\Omega_D|}, & i \text{ is adjacent to } j, \\ 0, & \text{otherwise.} \end{cases}$$

It should be noted that the assumption of the linear variation of the potential inside each tetrahedron is consistent with the MPFA methods used in the numerical solution of the diffusion equations on arbitrary unstructured grids [14].

2.3.1. Plasma with surfaces of imposed potential and/or electric field

Consider next a plasma bounded by a surface with imposed potential (Dirichlet) as shown in Fig. 5(a). The potential at the Dirichlet node $i \in A_{Di}$ of a surface A_{Di} is

$$\phi_i = \Phi_0. \tag{23}$$

In the case of Neumann boundary conditions the normal field at node $j \in A_{\mathrm{Ne}}$ of a surface A_{Ne} is

$$(\mathbf{E}_j)_n = (\mathbf{E}_0)_n. \tag{24}$$

Using the Gaussian surface corresponding to the dual cell Γ_j surrounding a node with index $j \in A_{\mathrm{Ne}}$ as shown in Fig. 5(b), Gauss's law given by Eq. (3) becomes

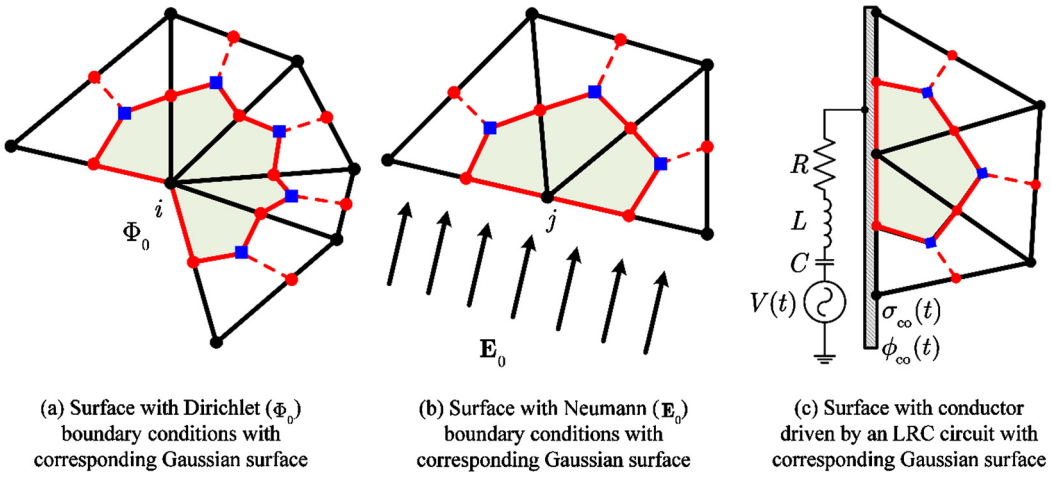


Fig. 5. Dirichlet node (a), Neumann node (b), and conductor node driven by LRC circuit boundary conditions showing the corresponding Gaussian surfaces.

$$\sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk}|A_{Dk}| \cdot \mathbf{n}_{Dj}|A_{Dj}| + \frac{1}{3}\mathbf{E}_0 \cdot \sum_{\substack{\sigma\in A_{\mathrm{Ne}} \\ A_\sigma \ni j}} \mathbf{n}_\sigma |A_\sigma| = \frac{Q_j}{\varepsilon_0}, \tag{25}$$

where $\mathbf{n}_\sigma$ is the outward unit normal to the boundary triangular face A_σ.

Any arbitrary combination of Dirichlet and Neumann surfaces in contact with the plasma can be considered by replacing equations in the linear system (22) with corresponding equations given by Eqs. (23) or (25). The actual structure of the resulting system of equations is shown in the next section.

2.3.2. Plasma with a conductor driven by an LRC circuit

Consider next a bounded plasma that includes a driven conductor as shown in Fig. 5(c). The conductor of surface A_{co} is driven by a voltage source $V(t)$ connected in series with a resistance, R, inductance, L, and capacitance, C, as shown in Fig. 5(c). The evaluation of the potential generalizes the approach in [13] following [15,16]. The integral Gauss' law Eq. (3) applied to the dual cell associated with a node $i \in A_{\mathrm{co}}$ shown in Fig. 5(c) becomes

$$\varepsilon_0 \sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk}|A_{Dk}| \cdot \mathbf{n}_{Di}|A_{Di}| - \sigma_i \sum_{\substack{A_\sigma\in A_{\mathrm{co}} \\ A_\sigma \ni i}} |A_\sigma| = Q_i, \tag{26}$$

where $\sigma_i(t)$ is the surface charge density and Q_i is the volume charge of node i. Assuming a perfect conductor with constant potential ϕ_{co}, the total surface charge density $\sigma_{\mathrm{co}}(t)$ on the driven conductor with total area $|A_{\mathrm{co}}|$ is obtained by summing all surface charge densities as

$$\begin{aligned} \sigma_{\mathrm{co}}|A_{\mathrm{co}}| &\equiv \sum_{i\in A_{\mathrm{co}}} \sigma_i \sum_{\substack{A_\sigma\in A_{\mathrm{co}} \\ A_\sigma\ni i}} |A_\sigma| \\ &= \varepsilon_0\phi_{\mathrm{co}} \sum_{i\in A_{\mathrm{co}}} \sum_{D=1}^{G_D(i)} \frac{|A_{Di}|^2}{9|\Omega_D|} + \varepsilon_0 \sum_{i\in A_{\mathrm{co}}} \sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{\substack{k\in\Omega_D \\ k\neq i}} \phi_k \mathbf{n}_{Dk}|A_{Dk}| \cdot \mathbf{n}_{Di}|A_{Di}| - \sum_{i\in A_{\mathrm{co}}} Q_i, \end{aligned} \tag{27}$$

where $\sum_{i\in A_{\mathrm{co}}} Q_i$ is the volume charge associated with the plasma in the dual cells of the conductor. In Eq. (27) $\phi_{\mathrm{co}}(t)$ and $\sigma_{\mathrm{co}}(t)$ are unknown variables and are evaluated from the boundary conditions from circuit laws. The charge conservation law at the conductor (or Kirchhoff's current law) becomes

$$|A_{\mathrm{co}}|\frac{d\sigma_{\mathrm{co}}}{dt} = \frac{dQ_{\mathrm{ci}}}{dt} + \frac{dQ_{\mathrm{cv}}}{dt}, \tag{28}$$

where, $Q_{\mathrm{cv}}(t)$ is the convective charge collected by the conductor from the plasma and $Q_{\mathrm{ci}}(t)$ is the charge on the conductor due to the circuit. Kirchhoff's voltage law for the lumped circuit shown in Fig. 5(c) is

$$L\frac{d^2Q_{\mathrm{ci}}}{dt^2} + R\frac{dQ_{\mathrm{ci}}}{dt} + C\,Q_{\mathrm{ci}}(t) = V(t) + \phi_{\mathrm{co}}(t). \tag{29}$$

Substituting a second-order, backward Euler finite-difference approximation of Eq. (29) [15] into a first-order, backward Euler finite-difference approximation of Eq. (28) it becomes

$$|A_{\rm co}|\sigma_{\rm co}(t) = |A_{\rm co}|\sigma_{\rm co}(t-\Delta t) + \Delta Q_{\rm cv}(t) + \frac{\Delta Q_{\rm ci}(t-\Delta t) + V(t) + \phi_{\rm co}(t)}{\frac{9L}{4\Delta t^2} + \frac{3R}{2\Delta t} + C}, \tag{30}$$

where

$$\begin{aligned}\Delta Q_{\rm ci}(t-\Delta t) = &-\left(-\frac{15L}{4\Delta t^2} - \frac{R}{2\Delta t} + C\right) Q_{\rm ci}(t-\Delta t) \\ &- \left(\frac{11L}{2\Delta t^2} + \frac{R}{2\Delta t}\right) Q_{\rm ci}(t-2\Delta t) + 8\frac{L}{4\Delta t^2} Q_{\rm ci}(t-3\Delta t) - \frac{L}{4\Delta t^2} Q_{\rm ci}(t-4\Delta t),\end{aligned} \tag{31}$$

and $\Delta Q_{\rm cv}(t) = Q_{\rm cv}(t) - Q_{\rm cv}(t-\Delta t)$ is the convective charge collected by the conductor from the time $t-\Delta t$ to t from the plasma. Substituting Eq. (27) into Eq. (30) it becomes

$$\begin{aligned}&\left[\varepsilon_0 \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{|A_{Di}|^2}{9|\Omega_D|} - \left(\frac{9L}{4\Delta t^2} + \frac{3R}{2\Delta t} + C\right)^{-1}\right]\phi_{\rm co}(t) + \varepsilon_0 \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{\substack{k\in\Omega_D \\ k\neq i}} \phi_k \mathbf{n}_{Dk}|A_{Dk}| \cdot \mathbf{n}_{Di}|A_{Di}| \\ &= |A_{\rm co}|\sigma_{\rm co}(t-\Delta t) + \Delta Q_{\rm cv}(t) + \sum_{i\in A_{\rm co}} Q_i + \left(\Delta Q_{\rm ci}(t-\Delta t) + V(t)\right)\left(\frac{9L}{4\Delta t^2} + \frac{3R}{2\Delta t} + C\right)^{-1}.\end{aligned} \tag{32}$$

For completeness we also assume that node j is at fixed potential Φ_0 (it can be, for example, grounded) and the node k is a part of the surface $A_{\rm Ne}$ with an applied electric field $\mathbf{E}_0$ then Eq. (32) coupled with the system (22) for the interior nodes results in

$$\begin{pmatrix} R_{11} & \cdots & R_{1i} & \cdots & R_{1j} & \cdots & R_{1G_d} & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & -1 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ R_{k1} & \cdots & R_{ki} & \cdots & R_{kj} & \cdots & R_{kG_d} & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ R_{G_d1} & \cdots & R_{G_di} & \cdots & R_{G_dj} & \cdots & R_{G_dG_d} & 0 \\ \tilde{R}_1 & \cdots & \tilde{R}_i & \cdots & \tilde{R}_j & \cdots & \tilde{R}_{G_d} & 1 \end{pmatrix} \begin{pmatrix} \phi_1 \\ \vdots \\ \phi_i \\ \vdots \\ \phi_j \\ \vdots \\ \phi_k \\ \vdots \\ \phi_{G_d} \\ \phi_{\rm co} \end{pmatrix} = \frac{1}{\varepsilon_0} \begin{pmatrix} Q_1 \\ \vdots \\ 0 \\ \vdots \\ \varepsilon_0\Phi_0 \\ \vdots \\ Q_k - \frac{1}{3}\varepsilon_0 \mathbf{E}_0 \cdot \sum_{\substack{\sigma\in A_{\rm Ne} \\ A_\sigma \ni k}} \mathbf{n}_\sigma |A_\sigma| \\ \vdots \\ Q_{G_d} \\ \tilde{Q} \end{pmatrix}, \tag{33}$$

where the ith row corresponds to the nodes of the driven conductor $A_{\rm co}$, the row j is for the conductor at fixed potential Φ_0, the kth row is for the Neumann boundary $A_{\rm Ne}$ with applied electric field $\mathbf{E}_0$, R_{ij} are defined by Eq. (22) and $\tilde{R}_i$ and $\tilde{Q}$ are given as

$$\tilde{R}_i = \begin{cases} \left[\varepsilon_0 \sum_{j\in A_{\rm co}} \sum_{D=1}^{G_D(j)} \frac{|A_{Dj}|^2}{9|\Omega_D|} - \left(\frac{9L}{4\Delta t^2} + \frac{3R}{2\Delta t} + C\right)^{-1}\right]^{-1} \left(\varepsilon_0 \sum_{\substack{D=1 \\ j\in\Omega_D \\ j\neq i}}^{G_D(i)} \frac{1}{9|\Omega_D|} \mathbf{n}_{Di}|A_{Di}| \cdot \mathbf{n}_{Dj}|A_{Dj}|\right), & i \in A_{\rm co}, \\ 0, & i \notin A_{\rm co}, \end{cases}$$

$$\tilde{Q} = \frac{\varepsilon_0|A_{\rm co}|\sigma_{\rm co}(t-\Delta t) + \varepsilon_0 \Delta Q_{\rm cv}(t) + \varepsilon_0 \sum_{i\in A_{\rm co}} Q_i(t) + \varepsilon_0 \frac{\Delta Q_{\rm ci}(t-\Delta t) + V(t)}{\frac{9L}{4\Delta t^2} + \frac{3R}{2\Delta t} + C}}{\varepsilon_0 \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{|A_{Di}|^2}{9|\Omega_D|} - \left(\frac{9L}{4\Delta t^2} + \frac{3R}{2\Delta t} + C\right)^{-1}}.$$

The system of sparse linear equations given by Eq. (33) is solved using restarted GMRES solver with the ILU(0) preconditioner following [17]. This derivation can be extended to a plasma with additional driven electrodes and, thus, allows the representation of a specific plasma device.

In the case of a floating conductor the potential is determined by the surface charge collected by it. This case is equivalent to a driven conductor shown in Fig. 5(c) connected to an open circuit with $C \to 0$. Gauss's law for the floating conductor is expressed as in Eq. (27) and charge conservation law given by Eq. (28) in discrete form becomes

$$|A_{\rm co}|\sigma_{\rm co}(t) = |A_{\rm co}|\sigma_{\rm co}(t-\Delta t) + \Delta Q_{\rm cv}(t), \tag{34}$$

where we took into account that there is no charge on the conductor due to the circuit.

Substituting Eq. (34) into Eq. (27), it becomes

$$\begin{aligned}&\varepsilon_0 \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{|A_{Di}|^2}{9|\Omega_D|}\phi_{\rm co}(t) + \varepsilon_0 \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{\substack{k\in\Omega_D\\ k\neq i}} \phi_k \mathbf{n}_{Dk}|A_{Dk}|\cdot\mathbf{n}_{Di}|A_{Di}| \\ &= |A_{\rm co}|\sigma_{\rm co}(t-\Delta t) + \Delta Q_{\rm cv}(t) + \sum_{i\in A_{\rm co}} Q_i.\end{aligned} \tag{35}$$

The coefficients $\tilde{R}_i$ and $\tilde{Q}$ from Eq. (33) are defined for floating conductor based on Eq. (35) as

$$\tilde{R}_i = \begin{cases} \left(\sum_{j\in A_{\rm co}} \sum_{D=1}^{G_D(j)} \frac{|A_{Dj}|^2}{9|\Omega_D|}\right)^{-1} \sum_{\substack{D=1\\ j\in\Omega_D\\ j\neq i}}^{G_D(i)} \frac{1}{9|\Omega_D|}\mathbf{n}_{Di}|A_{Di}|\cdot\mathbf{n}_{Dj}|A_{Dj}|, & i\in A_{\rm co},\\ 0, & i\notin A_{\rm co},\end{cases}$$

$$\tilde{Q} = \left(\sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{|A_{Di}|^2}{9|\Omega_D|}\right)^{-1} \left(|A_{\rm co}|\sigma_{\rm co}(t-\Delta t) + \Delta Q_{\rm cv}(t) + \sum_{i\in A_{\rm co}} Q_i(t)\right).$$

2.4. Particle injection from surfaces and periodic boundary conditions on unstructured meshes

Particle injection from surfaces follows [13,19]. Periodic boundary conditions are also implemented in EUPIC for a number of periodic surfaces pairs obtained by translation by a vector **R**, which, in general, is different for different pairs. The main difficulty in implementing periodic boundary conditions on unstructured meshes is the necessity to find a corresponding periodic face from which particle is reinjected from the periodic surface once it left the domain from the other. In order to circumvent this computationally expensive procedure we use mirror meshes on the periodic surfaces pairs. The list of the periodic faces of the two periodic boundaries separated by **R** is stored. When a particle crosses a periodic face, its position is shifted by **R** and reinjected from the other periodic face. This implementation is both computationally efficient and allows using arbitrary shaped periodic boundaries. For the periodic potential we use Dirichlet boundary conditions described in Sec. 2.3.1.

2.5. Nodal electric field and force interpolation

The electric field at node i can be found from

$$\mathbf{E} = -\lim_{V\to 0} \frac{1}{V} \oiint_A \phi d\mathbf{A} \tag{36}$$

applied to the dual cell Γ_i in Fig. 3

$$\mathbf{E}_i = -\frac{1}{|\Gamma_i|} \sum_{D=1}^{G_D(i)} \sum_{k\in\Omega_D} \int_{\hat{A}_{Dik}} \phi d\mathbf{A} \tag{37}$$

or after carrying out all integrations in fully discrete form

$$\mathbf{E}_i = -\left(\sum_{D=1}^{G_D(i)} |\Omega_D|\right)^{-1} \sum_{\substack{\sigma=1\\ i,A_\sigma\in\Omega_D\\ l\in\Omega_D\setminus A_\sigma}}^{G_\sigma(i)} \mathbf{n}_{Dl}|A_\sigma|\frac{1}{3}\sum_{j\in A_\sigma}\phi_j, \tag{38}$$

where $G_\sigma(i)$ is the number of faces surrounding the node i including the boundary faces for each node i is a part of the faces. Equation (38) can be used to evaluate electric field on both interior and boundary nodes. However, for a Neumann boundary node the field is evaluated directly from the imposed one following Eq. (24). For a Dirichlet node or driven-circuit node $i\in A_{Di}$ the electric field is given by

$$\mathbf{E}_i = \frac{\sigma_i}{\varepsilon_0}\mathbf{n}_i, \tag{39}$$

where σ_i is the surface charge density at node i and $\mathbf{n}_i$ is the unit outward normal at node i. Equation (39) in discrete form becomes

$$\mathbf{E}_i = -\frac{1}{3}\frac{\sigma_i}{\varepsilon_0} \sum_{\substack{\sigma\in A_{Di}\\ A_\sigma\ni j}} \mathbf{n}_\sigma|A_\sigma|. \tag{40}$$

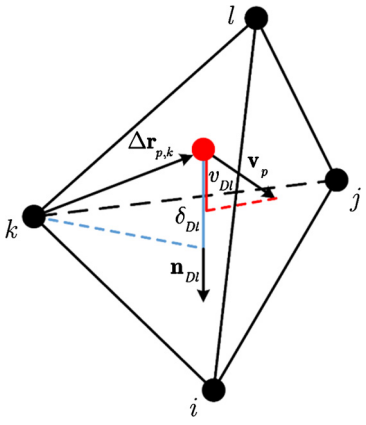


Fig. 6. Particle tracking algorithm.

The surface charge density, σ_i, is calculated from a finite volume MPFA of the Gauss' law applied to the Gaussian surfaces shown in Fig. 5(b, c) and is given by

$$\sigma_i = \frac{\varepsilon_0 \sum_{D=1}^{G_D(i)} \frac{1}{9|\Omega_D|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk}|A_{Dk}| \cdot \mathbf{n}_{Di}|A_{Di}| - Q_i}{\sum_{\substack{A_\sigma \in A_{Di} \\ A_\sigma \ni i}} |A_\sigma|}. \tag{41}$$

When a node is shared between two or more surfaces with different boundary conditions (for example, corners) an area-weighted average of the electric field is evaluated from each of these surfaces calculated following Eqs. (24) and (40).

The electric field at the position of a particle $\mathbf{r}_p \in \Omega_D$ is obtained through interpolation from the four nodes of the tetrahedral cell using the same weights as in Sec. 2.2 as

$$\mathbf{E}(x_p, y_p, z_p) = \sum_{m\in\Omega_D} \mathbf{E}_m W_{mD}(\mathbf{r}_p). \tag{42}$$

2.6. Integration of particle motion and particle tracer

The trajectory of a particle in the presence of an electrostatic field, $\mathbf{E}$, and an external magnetic field, $\mathbf{B}_{\text{ext}}$, is integrated as in [13] following Buneman's time-centered leapfrog formulation [18]

$$\begin{aligned} &\frac{\mathbf{v}_p^{t+\Delta t/2} - \mathbf{v}_p^{t-\Delta t/2}}{\Delta t} = \frac{q_p}{m_p}\left[\mathbf{E}^t(\mathbf{r}_p) + \frac{\mathbf{v}_p^{t+\Delta t/2} - \mathbf{v}_p^{t-\Delta t/2}}{2} \times \mathbf{B}_{\text{ext}}^t(\mathbf{r}_p)\right], \\ &\mathbf{r}_p^{t+\Delta t} = \mathbf{v}_p^{t+\Delta t/2}\Delta t + \mathbf{r}_p^t \end{aligned} \tag{43}$$

combined with Boris's algorithm [24] for the implementation of the $\mathbf{v}\times\mathbf{B}_{\text{ext}}$ rotation in case of an applied external magnetic field.

The particle search-locate algorithm in EUPIC is carried out using an improved version of the successive-neighbor algorithm developed in [19] for Direct Simulation Monte Carlo method on unstructured three-dimensional meshes. Compared to [19] the current algorithm requires less number of floating point operations. The steps for particles tracking in tetrahedra shown in Fig. 6 are outlined below:

(S.1) For each tetrahedral cell D using the outward normal $\mathbf{n}_{Dl}$ for each face l in the cell for each particle p calculate the projection of velocity $\mathbf{v}_p$ onto $\mathbf{n}_{Dl}$, $v_{Dl} = \mathbf{v}_p \cdot \mathbf{n}_{Dl}$ that gives the velocity at which the particle is approaching the plane of the corresponding face A_{Dl}.

(S.1.1) If $v_{Dl} < 0$ the particle is moving in the opposite direction of the face and intersection is not possible; set $\delta t_{Dl} = 1 + \Delta t$.

(S.1.2) If $v_{Dl} = 0$ the particle is moving parallel to the face and intersection with this face is not possible, set $\delta t_{Dl} = 1 + \Delta t$.

(S.1.3) If $v_{Dl} > 0$ the particle can cross the plane of the face and intersection is possible. In this case calculate the distance from the face to the particle in the cell, as $\delta_{Dl} = -\Delta\mathbf{r}_{p,k} \cdot \mathbf{n}_{Dl}$, where $\Delta\mathbf{r}_{p,k} = \mathbf{r}_p(t) - \mathbf{r}_k$, and $\mathbf{r}_k$ is the position of a node which lies into the face as shown in Fig. 6. Using the particle-face distance δ_{Dl} and the normal velocity v_{Dl} calculate the time needed for a particle to reach the face as, $\delta t_{Dl} = \delta_{Dl}/v_{Dl}$.

(S.2) Calculate $\delta t_{\min} = \min_{l \in \Omega_D}(\delta t_{Dl})$.

(S.2.1) If $\delta t_{\min} \le \Delta t$ then the particle crosses the corresponding face. Once a face which the particle crosses is determined, the neighbor cell is also found. It is achieved by storing a list of the global faces indices, a list of the global nodes indices opposite to the faces and a list of the global indices of neighbor cells in the same order for every cell in the grid, allowing efficient next cell determination during particle crossing. According to the type of the crossed face the following cases are possible:

(S.2.1.1) If the particle crosses a boundary face, the boundary conditions associated with this face are imposed. The new location of the particle in the next cell (or the same cell in case of a reflection) is $\mathbf{r}_p(t + \delta t_{\min}) = \mathbf{r}_p(t) + \mathbf{v}_p \delta t_{\min}$ and the new time step for this particle is $\Delta t^* = \Delta t - \delta t_{\min}$. The velocity of the particle in the next cell depends on the boundary condition. Then this process is repeated with new time step until the particle remains in a new cell or leaves the domain.

(S.2.1.2) If particle crosses a free boundary, it is deleted from the computational domain.

(S.2.2) If $\delta t_{\min} > \Delta t$ the particle remains in the current cell and its final position is $\mathbf{r}_p(t + \Delta t) = \mathbf{r}_p(t) + \mathbf{v}_p \Delta t$.

2.7. Nodal macroscopic properties

Evaluation of nodal macroscopic properties for multiple particle weights is based on output samples from EUPIC. The species s number density at a node i is obtained using the volume of the supercell Ω_i shown in Fig. 3, as

$$n_s(x_i, y_i, z_i) \equiv n_i^s = \mathrm{N}_s(i,t)/|\Omega_i|, \tag{44}$$

where the number of real particles of species s inside a supercell Ω_i shown in Fig. 3(a) is

$$\mathrm{N}_s(i,t) \equiv \mathrm{N}_i^s(t) = \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} F_{Wp}, \tag{45}$$

where N_p^s is the number of computational particles of species s in the domain. The species s mean velocity at a node i is

$$\mathbf{V}_s(i,t) \equiv \left\{V_{sx}(i,t), V_{sy}(i,t), V_{sz}(i,t)\right\} = \frac{1}{\mathrm{N}_s(i,t)} \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} F_{Wp} \mathbf{v}_{sp}. \tag{46}$$

The mass-average velocity $\mathbf{V}(i,t)$ at a node i is

$$\mathbf{V}(i,t) \equiv \left\{V_x(i,t), V_y(i,t), V_z(i,t)\right\} = \frac{1}{\sum_s n_s(i,t) m_s} \sum_s n_s(i,t) m_s \mathbf{V}_s. \tag{47}$$

The thermal (or random) velocity for the p-th particle of species s at time t is given with respect to the node species-average velocity

$$\mathbf{C}_{sp} = \mathbf{v}_{sp} - \mathbf{V}_s(i,t) \tag{48}$$

or with respect to the cell mass-average velocity as

$$\mathbf{C}_{sp}^* = \mathbf{v}_{sp} - \mathbf{V}(i,t) = \mathbf{C}_{sp} + \mathbf{W}_{sp}. \tag{49}$$

The species s translational temperature, scalar pressure, pressure tensor, heat flux vector, and current density in the species-average system or the mass-average system (designated by *) are obtained as

$$\begin{aligned}
&\frac{3}{2} k T_s^{(*)}(i,t) = \frac{1}{\mathrm{N}_s(i,t)} \frac{1}{2} m_s \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} F_{Wp} C_{sp}^{(*)2}, \\
&p_s^{(*)}(i,t) = \frac{1}{3} n_s(i,t) \frac{1}{\mathrm{N}_s(i,t)} m_s \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} F_{Wp} C_{sp}^{(*)2} = n_s(i,t) k_B T_s^{(*)}(i,t), \\
&\mathbf{P}_s^{(*)}(i,t) = n_s(i,t) \frac{1}{\mathrm{N}_s(i,t)} m_s \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} F_{Wp} \mathbf{C}_{sp}^{(*)} \mathbf{C}_{sp}^{(*)},
\end{aligned} \tag{50}$$

$$\mathbf{q}_s^{(*)}(i,t) = \frac{1}{2}\frac{1}{\mathrm{N}_s(i,t)} m_s \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} F_{Wp} \mathbf{C}_{sp}^{(*)} \mathbf{C}_{sp}^{(*)},$$

$$\mathbf{j}_s = \frac{1}{|\Omega_i|} q_s \sum_{\substack{D=1 \\ \Omega_D \ni i}}^{N_D} \sum_{\substack{p=1 \\ \mathbf{r}_p \in \Omega_D}}^{N_p^s} W_{iD}(\mathbf{r}_p) F_{Wp} \mathbf{v}_{sp}.$$

3. Parallelization implementation

The EUPIC is parallelized using OpenMP in order to take advantage of multi-platform shared-memory multiprocessing computers including GPUs. The parallelization of the algorithms for charge assignment, force interpolation, and particle motion is achieved by distributing clusters of tetrahedral cells to distinct parallel threads that compute independently from each other. The dynamic load balancing based on the number of particle processed by each thread is used to achieve better parallel efficiency. It is achieved by counting the number of particles in all cells every 100 iterations and distributing cell clusters that contains a nearly equal amount of particles for parallel processing to all threads in gather/scatter and motion algorithms. It allows to avoid using OpenMP's intrinsic load balancing algorithms that introduce additional overhead for every iteration. For the evaluation of macroscopic properties the corresponding moments are first calculated for cells in parallel. Then for each node the moments are summed for all cells surrounding a node in parallel and independently of other nodes. These sums are then used to calculate the macroscopic properties.

Parallelization of the GMRES computational cycle involves the dot product calculations that are computationally expensive. These calculations are performed by splitting the vectors and matrices in stripes and offloading calculations to different threads. In case of matrix–vector products the matrix is split by rows taking into account the number of non-zero elements achieving better load balancing. In addition, during the ILU(0) preconditioning it is necessary to solve an LU sparse system. The main problem with such solutions is data dependencies between unknown values. In order to reduce such data dependencies we use a combination of nodes reordering based on the coloring of the data dependency directed acyclic graph [25,26] and calculation ordering based on the level-scheduling scheme [27,26]. First, we create a directed acyclic graph of data dependencies of the L and U matrices and use a graph-coloring scheme to color this graph. The nodes are reordered in a way that those with the same color are sitting next to each other. Then the level-scheduling scheme is used to reordered system to extract independent calculation steps from the solution of the LU system. These reordering steps are performed only once in the beginning of the simulation and then used during the entire EUPIC simulation. For the main stopping criteria of the restarted GMRES method we use the relative residual e^n_{GMRES} at outer GMRES iteration n defined as

$$e^n_{\mathrm{GMRES}} = \frac{\|r_n\|}{\max(\varepsilon_0, \|\tilde{Q}\|)} \leq \varepsilon. \tag{51}$$

In the above $r_n = \tilde{R}\phi_n - \tilde{Q}$ is the residual and ϕ_n is the potential at the outer GMRES iteration n, $\|\ldots\|$ is the either L^2 or L^∞ norm, ε_0 is the constant that determines the minimum possible value of $\|\tilde{Q}\|$ to avoid division by zero, ε is the specified relative error. The calculation of the actual residual is computationally expensive, therefore, during the inner GMRES iterations the following estimate for the preconditioned relative residual is used

$$\tilde{e}^m_{\mathrm{GMRES}} = \min\left(\frac{\|M^{-1}r_m\|}{\max(\varepsilon_0, \|M^{-1}\tilde{Q}\|)}, \frac{\|r_0\|}{\|M^{-1}r_0\|}\frac{\|M^{-1}r_m\|}{\max(\varepsilon_0, \|\tilde{Q}\|)}\right) \leq \varepsilon, \tag{52}$$

where m is the inner GMRES iteration, M is ILU(0) preconditioner matrix, r_0 is the residual at the beginning of inner iterations, r_m is the residual at the m inner iteration. Since the satisfaction of the condition given by Eq. (52) doesn't guarantee the satisfaction of Eq. (51), the condition in Eq. (51) is also evaluated a posteriori. In the simulations presented in this work the rather conservative values for stopping criteria of $\varepsilon = 10^{-11}$, $\varepsilon_0 = 10^{-13}$ were used accompanied by a Krylov subspace dimension of 100. The GMRES solver usually converged in less than 200 of iterations depending on the particular problem and the relative residual defined by Eq. (51) was usually less than 5×10^{-13}.

The parallelization of the electric field calculation algorithm is achieved by first rewriting Eq. (38) explicitly as a product between a sparse matrix and a vector as

$$\mathbf{E}_i = \sum_{j=1}^{G_d} \tilde{\mathbf{E}}_{ij}\phi_j, \tag{53}$$

where $\tilde{\mathbf{E}}_{ij}$ is the vector-valued coefficient showing the contribution to the electric field at node i from the potential at node j. The vector-valued matrix $\tilde{\mathbf{E}}$ has the same fill-in structure as the matrix R in Eq. (22). Then parallelization follows the method outlined above for matrix–vector multiplication. The advantage of explicit matrix–vector product is the increased efficiency due to better CPU cache utilization. Once electric field at nodes is calculated it is corrected on the boundary nodes following Eq. (40).

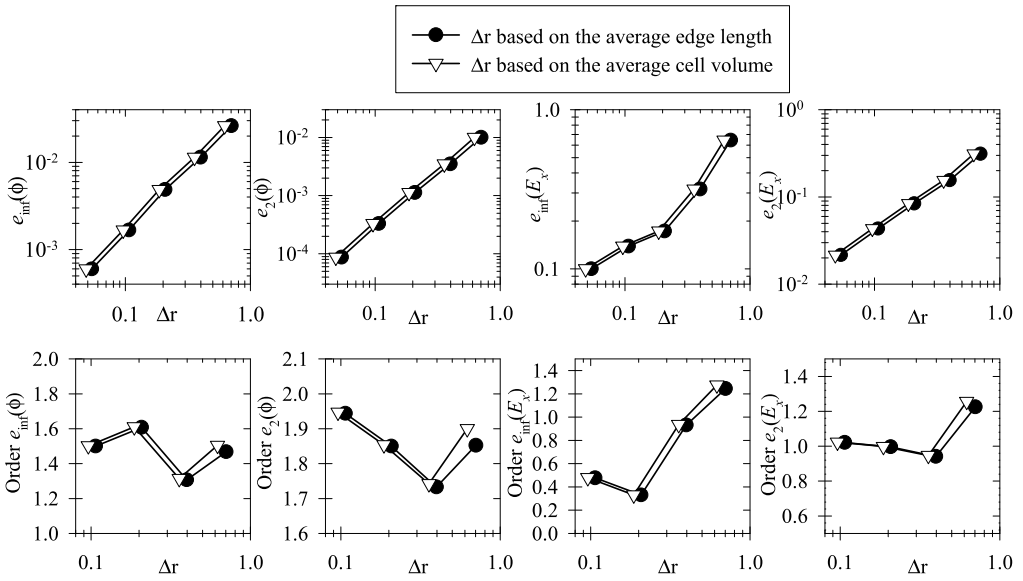


Fig. 7. Effects of edge-averaged and volume-averaged cell size on L^2 and L^∞ norms of the relative error in potential and x-component of electric field (top) and corresponding orders of convergence (bottom).

4. Verification, validation and error analysis

The first test case provides verification of the electrostatic solver, nodal field evaluation, and a grid sensitivity analysis. It involves a grounded conducting sphere of radius R in vacuum with a uniform applied electric field $\mathbf{E}_0$ along z-axis and Neumann boundary conditions $\mathbf{E}(x, y, z = 5R) = \mathbf{E}_0$. The analytic solution [28] is

$$\begin{aligned} \phi(x, y, z) &= -E_0\left(1 - \frac{R^3}{r^3}\right)z, \\ \mathbf{E}(x, y, z) &= E_0\left[3\frac{zR^3}{r^5}\mathbf{r} + \left(1 - \frac{R^3}{r^3}\right)\mathbf{k}\right], \end{aligned} \tag{54}$$

where $r = \sqrt{x^2 + y^2 + z^2}$, $\mathbf{k}$ is the unit vector along z. Fig. 7 presents the effects of the cell size on the L^2 norm of the relative error in $\phi(x, y, z)$ and $E_x(x, y, z)$ defined as $e_2(f_h) = \|f_h - f_e\|_2/\|f_e\|_2$ and the L^∞ norm of the relative error defined as $e_\infty(f) = \|f_h - f_e\|_\infty/\|f_e\|_\infty$, where f_h is a numerical solution obtained with the average cell size Δr, f_e is an exact solution given by Eq. (54). The discrete L^2 norm is $\|f\|_2 = \sqrt{\sum_{i=1}^{G_d} f_i^2}$ and the discrete L^∞ norm is $\|f\|_\infty = \max_i |f_i|$. Fig. 7 shows also the order of convergence given by $(\log e_2(f_{2h}) - \log e_2(f_h))/\log 2$ for different cell sizes. We use two definitions to access the impact of the cell size, Δr, an edge-averaged with $\Delta r = \sum_{i=1}^{G_d} l_i/G_d$ and a volume-averaged with $\Delta r = (6\sqrt{2}/|\bar{\Omega}|)^{1/3}$ and $|\bar{\Omega}| = \sum_{i=1}^{G_d} |\Omega_i|/G_d$. Fig. 7 shows that both definitions lead to similar results, with $e_2(\phi) \sim O(\Delta r^2)$ and $e_2(E_x) \sim O(\Delta r)$ that is consistent with theoretical estimations [14].

The second test case expands with the verification of loading, gather and scatter procedures. It involves a ground sphere of radius R enclosing a cloud of stationary positive ions with the linearly varying number density $n = n_0(1 - r/R)$ for $r \le R$, $n = 0$ for $r > R$ and Dirichlet boundary conditions $\phi(r = R) = 0$, where $r = \sqrt{x^2 + y^2 + z^2}$. The potential and electric field are spherically symmetric and are given by

$$\begin{aligned} \phi(r) &= \phi_0\left[1 - 2\left(\frac{r}{R}\right)^2 + \left(\frac{r}{R}\right)^3\right] \\ \mathbf{E}(r) &= \frac{\phi_0}{R}\left(4 - 3\frac{r}{R}\right)\frac{\mathbf{r}}{R} \end{aligned} \tag{55}$$

where $\phi_0 = en_0R^2/(12\varepsilon_0)$ is a potential in the center with the maximum electric field $E_{max} = 4\phi_0/3R$ at $r = 2R/3$. Spatial discretization is dictated by the need to resolve the maximum gradient $\Delta\phi/E_{max}$ and $\Delta r/R \simeq 1.5\alpha$ where $\alpha \le 0.1$ is a grid sensitivity parameter. The simulations were performed with $R = 0.1$ m, $n_0 = 10^{13}$ m^{-3}, $\alpha = 2 \times 10^{-2}$; 4×10^{-2}; 8×10^{-2} and with 300 computational particles per cell.

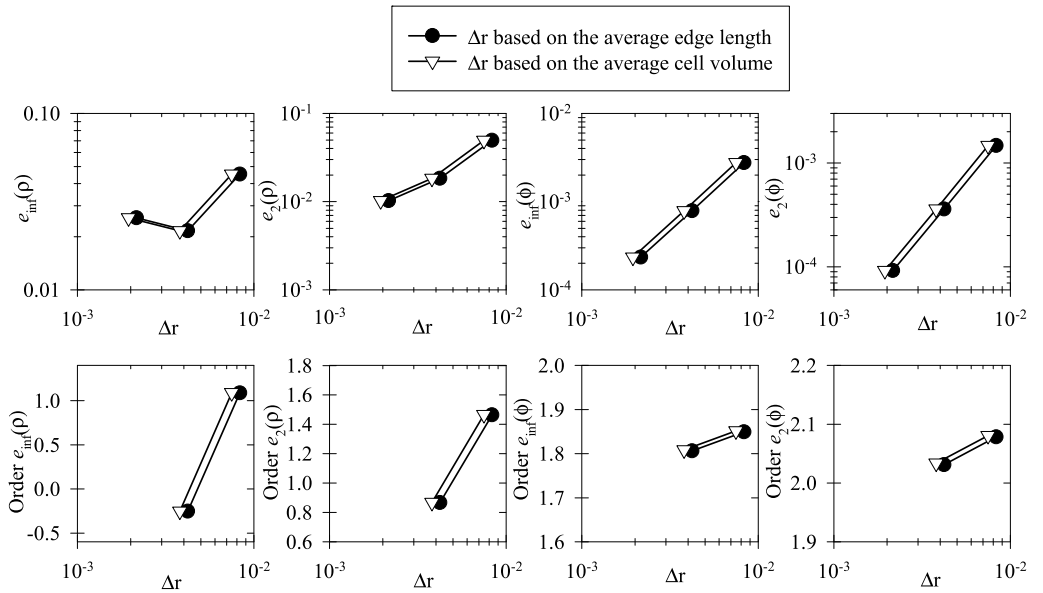


Fig. 8. Effects of edge-averaged and volume-averaged cell size on L^2 and L^∞ norms of the relative error in charge density and potential (top) and corresponding orders of convergence (bottom).

Fig. 8 presents the effect of grid size on the L^2 norm and L^∞ norm of the relative errors in charge density $\rho(x, y, z)$ and potential $\phi(x, y, z)$, and the order of convergence for different cell sizes. The results show that $e_2(\rho) \sim O(\Delta r)$ and $e_2(\phi) \sim O(\Delta r^2)$.

The third test case examines the momentum and energy conservation properties of the electrostatic solver and leapfrog time integrator along with temporal sensitivity analysis. It involves an electron initially at rest placed inside the grounded sphere with the linearly-varying stationary ion density distribution as in the previous case. With immobile ions and initial conditions the electron is trapped in the potential well and its total energy must be conserved for all times. In addition, the translational energy and momentum variations must be periodic functions with constant amplitudes. The time step for integration of the electron trajectory is bounded for stability by $\Delta t < \Delta r/v_e = 3\sqrt{6}\alpha/(4\omega_e)$, where $\omega_e = \sqrt{(e^2 n_0)/(m_e \varepsilon_0)}$. The simulations were performed with $R = 0.1$ m, $n_0 = 10^{14}$ m^{-3}, $\alpha = 3 \times 10^{-2}$ and variable timesteps $\Delta t = 5 \times 10^{-11}$; 5×10^{-10}; 5×10^{-9} s that satisfy stability criterion. The grid consists of about 1,500,000 cells and 200,000 nodes and is loaded with about 300,000,000 singly-charged massive ions. The L^2 norm of the error in the electric field was found to be below 3% for the most part of the domain. The electron kinetic energy evolution in Fig. 9(a) and x-component of momentum in Fig. 9(b) show that they vary harmonically with almost constant amplitudes for up to 2e–7 s that corresponds to 113 ω_e^{-1} that confirms the conservation of energy and momentum. In addition, it can be seen that the effects of varying time-step are minimal for up to 2e–7 s that corresponds to 113 ω_e^{-1} or three periods of electron oscillation.

The fourth test case provides analysis of heating, slowing down and deflection times of EUPIC following [11] and [13]. The heating time τ_{Hs} is defined as the time for the average kinetic energy per particle of species s to increase its energy by $k_B T_s/2$,

$$\left\langle \Delta E(\tau_{Hs}) \right\rangle = k_B T_s(0)/2. \tag{56}$$

The numerical slowing-down time τ_S is the time for the average parallel velocity component of species s, $\langle v_{s\parallel}(\tau_S) \rangle$ to become

$$\left\langle v_{s\parallel}(\tau_S) \right\rangle = \frac{1}{N_s} \sum_{p=1}^{N_s} F_{Wp} v_{p\parallel}(\tau_S) = \left\langle v_{s\parallel}(0) \right\rangle / \exp(1), \tag{57}$$

where $\mathbf{v}_{p\parallel}(0) = \mathbf{v}_p(0)$ is the initial velocity of the particle p, $v_{p\parallel}(t)$ is the projection of the particle p velocity vector at time t onto the vector $\mathbf{v}_p(0)/|\mathbf{v}_p(0)|$. For simulation times less than τ_S the plasma can be assumed collisionless. The deflection time, τ_d, is also related to numerical collisions [29] and is defined as the time for the root-mean-square average deflection angle to reach 90 degrees,

$$\sqrt{\left\langle \varphi_s^2(\tau_d) \right\rangle} = \sqrt{\frac{1}{N_s} \sum_{p=1}^{N_s} F_{Wp} \varphi_p^2(\tau_d)} = \pi/2, \tag{58}$$

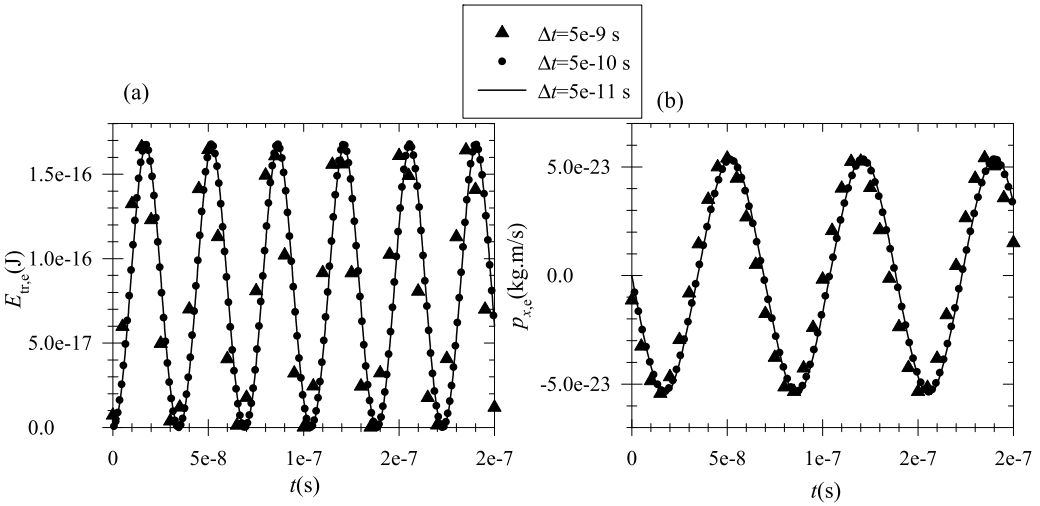


Fig. 9. Effects of time step on translational energy (a) and x-component of momentum (b) of an electron in a potential well.

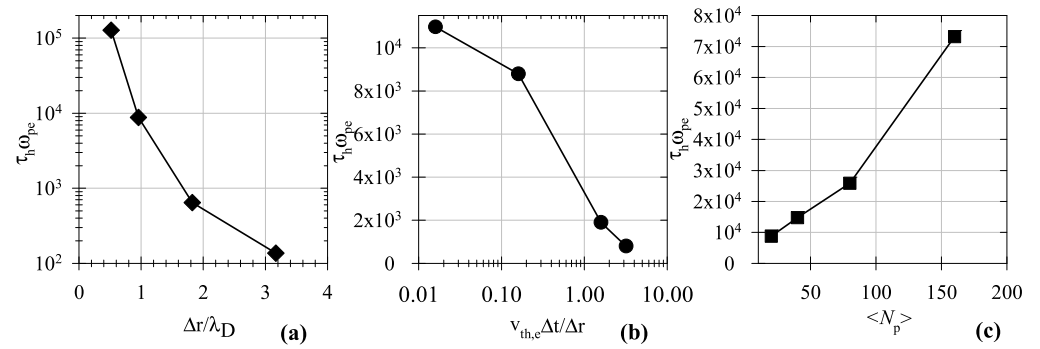


Fig. 10. Heating time in EUPIC simulations showing the effects of grid size (a), time step (b) and the average number of computational particles per cell (c).

where the deflection angle for a particle p at time t is defined as

$$\varphi_p(t) = \cos^{-1}\left[\left(\mathbf{v}_p(0)\cdot\mathbf{v}_p(t)\right)/\left(|\mathbf{v}_p(0)||\mathbf{v}_p(t)|\right)\right]. \tag{59}$$

The simulations are performed in a 3D cubic domain with periodic boundary conditions at all sides. The potential at the outer boundaries is fixed at 0 V to eliminate boundary effects. The plasma is loaded initially following a quasi-equilibrium Maxwellian distribution with $n_e = n_i = 10^{16}$ m^{-3}, $T_e = T_i = 2$ eV and zero drift velocity. Previous investigations [11,13] have shown that the CIC scheme leads to much larger heating times compared to the NGP scheme. From Eqs. (11) and (12) it can be seen that on unstructured tetrahedral meshes both schemes require calculation of 4 dot products and as such have comparable computational cost. Therefore, on unstructured meshes the NGP charge weighting scheme does not have any advantage even from computational costs point of view. For this reason, all simulations in the paper are performed using the CIC charge weighting scheme.

Fig. 10a shows the variation of the heating time as the function of the grid spacing at $\Delta t = 0.1\omega_{pe}^{-1}$ and $\langle N_p\rangle = 20$. As the grid spacing increases the heating time is decreasing. Doubling the grid spacing from $\Delta r = 0.5\lambda_{De}$ to $\Delta r = \lambda_{De}$ results in an almost order of magnitude decrease in the heating time. Further increase in the grid spacing lowers the heating time to $10^3\omega_{pe}^{-1}$ at $\Delta r = 3.17\lambda_{De}$. The effect of the time step at fixed grid spacing $\Delta r = \lambda_{De}$ and average number of computational particles per cell $\langle N_p\rangle = 20$ on the heating time is shown in Fig. 10b. The general trend is that increasing time step leads to decreasing the heating time. At time steps below $\Delta t = 0.1595(\Delta r/v_{\mathrm{th},e}) = 0.1\omega_{pe}^{-1}$ the further decrease in time step by an order of magnitude from $\Delta t = 0.1595(\Delta r/v_{\mathrm{th},e}) = 0.1\omega_{pe}^{-1}$ to $\Delta t = 0.01595(\Delta r/v_{\mathrm{th},e}) = 0.01\omega_{pe}^{-1}$ only slightly increases the heating time from $8.8\times10^3\omega_{pe}^{-1}$ to $1.1\times10^4\omega_{pe}^{-1}$. Similar behavior was found in the earlier simulations of unstructured PIC on Delaunay–Voronoi meshes [13]. The dependence of the heating time on the average number of particles per cell at constant grid spacing $\Delta r = \lambda_{De}$ and time step $\Delta t = 0.1\omega_{pe}^{-1}$ is depicted in Fig. 10c. The increase in the average number of particles per cell in the range from 20 to 80 results in the almost linear increase in the heating time from $8.8\times10^3\omega_{pe}^{-1}$ to $2.6\times10^4\omega_{pe}^{-1}$. The further increase in the average number of computational particles per cell from 80 to 160 particles per cell bumps the heating time to $7.3\times10^4\omega_{pe}^{-1}$. The results from Fig. 10 are used as a guide in the determining of the EUPIC

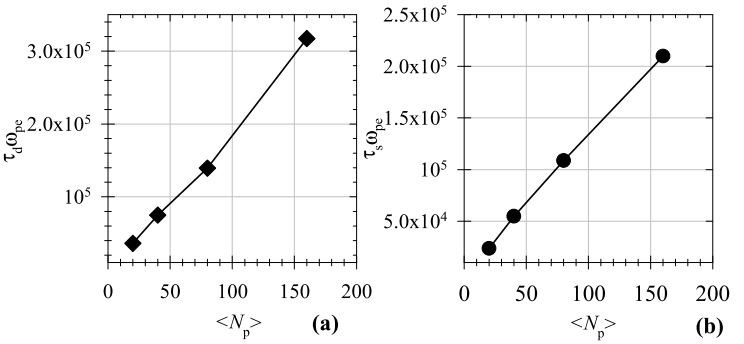


Fig. 11. Deflection (a) and slowing down (b) times in EUPIC simulations as a function of the average number of computational particles per cell with $\Delta r = \lambda_{De}$ and $\Delta t = 0.1\omega_{pe}^{-1}$.

Table 1
Input conditions for EUPIC simulations of cylindrical Langmuir probes.

Case	Thin sheath	OML
R_D (m)	1.2×10^{-2}	5×10^{-4}
r_p (m)	10^{-3}	10^{-5}
l_p (m)	10^{-2}	10^{-4}
$n_e = n_i$ (m^{-3})	10^{16}	10^{16}
$T_e = T_i$ (eV)	2	2
r_p/λ_D	10	0.1
$e\Phi_p/kT_e$	−25; −2.5; 0; 2.5; 25	1; 3; 5; 7; 9

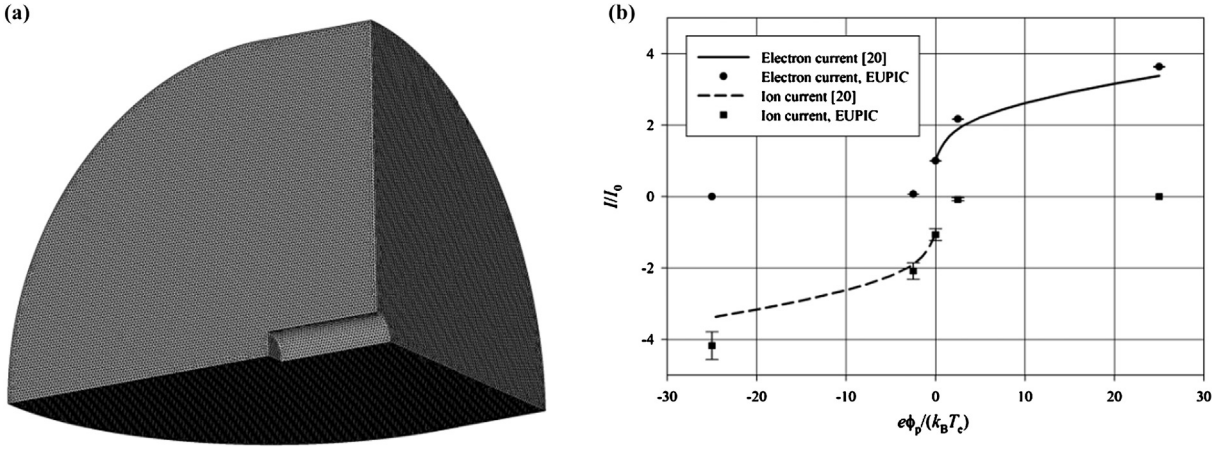


Fig. 12. EUPIC simulation of current collection by a cylindrical probe in thin sheath regime ($r_p \gg \lambda_D$): (a) Computational domain; (b) the normalized collected electron and ion currents.

simulation parameters in the subsequent simulations. Fig. 11 shows deflection (a) and slowing down times (b) as a function of the average number of computational particles per cell at constant grid spacing $\Delta r = \lambda_{De}$ and time step $\Delta t = 0.1\omega_{pe}^{-1}$. Both times increase almost linearly as the average number of computational particles increases. The deflection and slowing down times are increased from $\tau_d = 3.6 \times 10^4 \omega_{pe}^{-1}$ and $\tau_S = 2.4 \times 10^4 \omega_{pe}^{-1}$ respectively at $\langle N_p \rangle = 20$ to $\tau_d = 3.2 \times 10^5 \omega_{pe}^{-1}$ and $\tau_S = 2.1 \times 10^5 \omega_{pe}^{-1}$ respectively at $\langle N_p \rangle = 160$. It should be noted that deflection and slowing down times are at least three times larger than the corresponding heating time.

The fifth test case involves the simulation of current collection by a cylindrical Langmuir probe of length L_p and radius r_p in a collisionless plasma. The input parameters are shown in Table 1 and cover the regime of operation from thin sheath ($r_p \gg \lambda_D$) to orbital-motion limited (OML) ($r_p \ll \lambda_D$). Fig. 12a shows the simulation domain for the thin sheath cases as an ellipsoidal region with distances from the probe in azimuthal and radial directions equal to $12r_p$. Due to the symmetry of the problem one-eighth of the actual domain is used. The computational domain consists of about 1.7×10^6 tetrahedral cells. Boundary conditions include fixed potential at the probe Φ_p and zero potential at the outer ellipsoid-like surface corresponding to the unperturbed plasma. On the sidewalls zero normal component of the electric field is imposed to represent the planes of symmetry. The particles are injected through the outer surface following Maxwellian distribution function with the parameters from Table 1. Computational particles reaching the probe and the outer surface are removed from the

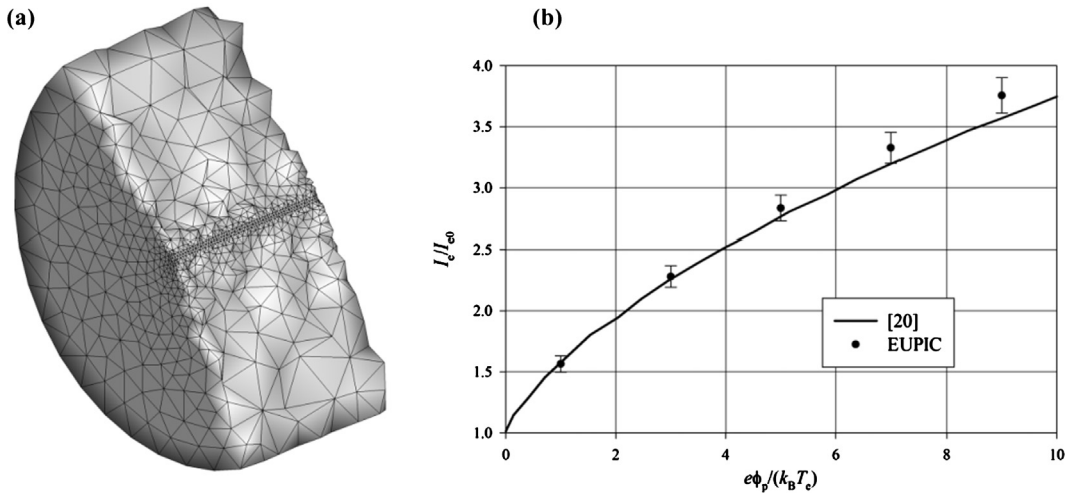


Fig. 13. EUPIC simulation of current collection by a cylindrical probe in the OML regime ($r_p \ll \lambda_D$): (a) Computational domain; (b) the normalized collected electron current as a function of normalized probe potential.

simulations. On the symmetry planes the particles are specularly reflected. The electron current to the probe (retarded current) is considered as positive and the ion current (accelerated current) as negative. Fig. 12b plots the electron and ion currents normalized by their respective thermal currents ($I_{e,i0} = n_{e,i} r_p l_p \sqrt{2\pi k_B T_{e,i}/m_{e,i}}$) as a function of the probe potential normalized by the electron temperature. The results of the EUPIC simulations compare favorably to the experimentally validated numerical predictions by Laframboise [20].

Fig. 13a show the cut of the computational domain for the EUPIC simulation of current collection by a cylindrical probe in the OML regime ($r_p \ll \lambda_D$). The Langmuir probe is modeled as a cylinder surrounded by a cylindrical domain. Fixed probe potential Φ_p is set at the inner cylinder while zero potential is specified at the outer cylinder corresponding to the unperturbed plasma. On the sidewalls zero normal component of the electric field boundary condition is used to represent the planes of symmetry. The particles are injected through the surface of the outer cylinder following Maxwellian distribution function with the parameters from Table 1 corresponding to the OML case. Fig. 13a shows the comparison of the electron current normalized by the random electron current calculated by EUPIC method with the Laframboise simulations [20] as a function of the probe potential normalized by the electron temperature. The results of these simulations compare well.

5. Parallelization efficiency

In this section the parallelization efficiency of the OpenMP implementation in EUPIC is discussed. For this purpose, simulations of current collection by a cylindrical Langmuir probe in the thin sheath regime were performed following the parameters listed in Table 1 for a case of $e\Phi_p/kT_e = 25$. A total of 2.285×10^6 tetrahedral cells were used and initially 1.2×10^8 of computational particles were loaded. The simulations were performed on the dual socket node with two Intel Xeon E5-2690 CPUs with the total number of 16 physical cores and enabled hyper-threading, a technology that allows to execute two streams of operations on the same core. The EUPIC code was compiled using the Intel Fortran Compiler 17.0.1 on Red Hat Enterprise Linux Server 6.8. The simulations have shown that the performance strongly depends on how calculations are spread across different cores. The best performance was achieved by using the following environment variables: OMP_PLACES = cores and OMP_PROC_BIND = close that were introduced in the OpenMP 4 standard. With these options the operating system binds OpenMP threads to physical cores as close as possible to the parent thread and don't allow their migration to other cores. Compared to the default options these settings give performance boost around 30% on up to 8 threads while for more threads the performance boost is less than 5%. Fig. 14 presents the speedup and parallel efficiency (the ratio of the actual speedup to the theoretical speedup) as a function of the number of OpenMP threads. It can be seen that up to 8 threads the parallel efficiency is quite high and monotonically decreases from 96% for two threads to 82% for 8 threads as the number of threads increases. Up to 8 threads due to the choice of the environment variables the simulation was running on a single CPU of the dual socket configuration. Once the number of threads exceeds 8, as can be seen from Fig. 14, the performance drops due to the non-uniform memory access (NUMA) architecture and particle data structure in the EUPIC code. The particle data in EUPIC are represented by global arrays and linked lists so that while a particle moves from one cell to another it still occupies the same position in the global array and only its pointer is updated. Since particle data are stored in the memory of one CPU, the access to this data from the other CPU is much slower due to the NUMA memory design architecture. With the choice of OpenMP environment variables in our simulations, if the number of threads exceeds 8 but below 16, then 8 threads are assigned to 8 physical cores of one CPU and the remaining threads are assigned to the physical cores of another CPU. In this case 8 threads have fast access to the data and finish their simulations earlier than remaining threads from another CPU. As a result this 8 threads are waiting for most of the

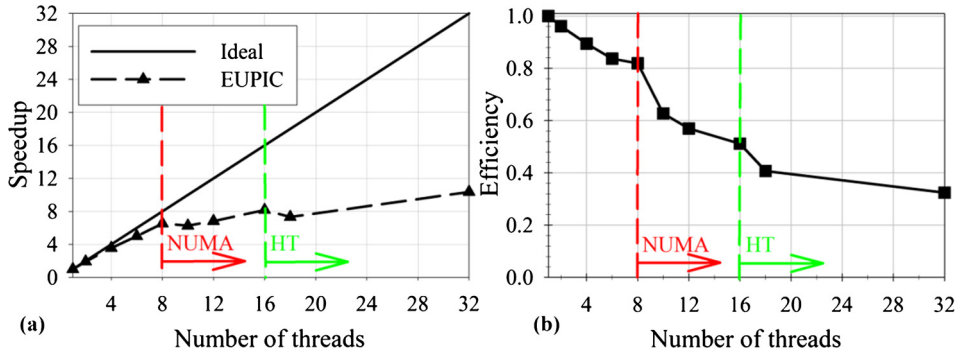


Fig. 14. Speedup (a) and parallel efficiency (b) of the EUPIC simulation of the current collection by a cylindrical Langmuir probe in the thin sheath regime as a function of the number of OpenMP threads. The red line shows the number of threads after which NUMA effects start playing the role. The green line shows the number of threads after which the hyper-threading technology is used. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Table 2
The average iteration time in the EUPIC simulation of the current collection by a cylindrical Langmuir probe and the percentage of time spend in each step of the EUPIC cycle.

Number of threads	Average time per iteration (s)	Percentage of the average time spent in each algorithm in PIC cycle				
		Charge assignment	Particle motion	Potential solver	Electric field interpolation and particle velocity update	Other
1	64.9434	19.0564	45.3219	12.1982	23.1375	0.2860
2	33.7756	18.0534	45.3070	13.1228	23.1334	0.3835
4	18.1497	17.4091	44.6110	14.4511	22.9895	0.5393
6	12.9278	17.0111	43.8054	15.9242	22.5314	0.7279
8	9.9075	16.8027	42.5648	17.8535	21.8960	0.8830
10	10.3511	17.1590	38.5253	22.9041	20.5811	0.8306
12	9.4986	16.6318	39.2598	23.1179	20.1226	0.8680
16	7.9301	15.6215	37.5267	26.3841	19.4789	0.9888
18	8.8555	14.5536	39.8162	25.0151	19.5062	1.1089
32	6.2594	13.5222	37.3569	27.3490	20.2428	1.5291

simulation time for the remaining threads of another CPU to complete their part of the computations. For the case of up to 16 threads, the parallel efficiency remains rather low because the simulation is limited by the memory access. At 16 threads all cores were loaded and each thread had much less work to do and, thus, much less data to transfer. Therefore, some speedup at 16 threads compared to 8 threads can be observed as seen in Fig. 14. Tests with more than 16 threads result in some of the cores executing two streams of operations due to the usage of hyper-threading and leads to the slowdown shown in Fig. 14. When the node is loaded fully with 32 threads by using all physical and virtual cores, another speedup in performance is realized as shown in Fig. 14. In this case the full load on all cores mitigates effects of memory access due to the NUMA architecture by taking an advantage of the fact that if one thread of a core is waiting data to perform calculations the other thread can perform its own computations resulting in more uniform load of the cores. The speedup of 10.3753 for 32 threads corresponds to the effective parallel efficiency of 0.65 based on the number of physical cores. One way to reduce the negative performance issues due to the NUMA architecture is to use message passing interface (MPI) for each CPU of the node. This strategy will also allow to run simulations on multiple nodes.

To better evaluate the impact of memory access on operations Table 2 presents the averaged time per iteration and the percentage of time spend in each step of the EUPIC cycle. It can be seen that the most time is spent in the particle motion algorithm. The remaining time is spent in charge assignment, potential solver using GMRES and electric field interpolation, and particle velocity update. As the number of threads increases the percentage of time spent in charge assignment, particle motion and electric field interpolation decreases while the percentage of time spent in the potential solver increases. This indicates worse parallel efficiency of the GMRES solver. This is partially due to the usage of the ILU(0) preconditioner which considerably reduces the number of operations but has inherit data dependencies that lead to low parallel efficiency as has been reported in the literature [25]. Another reason is the aforementioned NUMA memory design effects since GMRES data correspond to one CPU. It should be noted that our choice of the relative tolerance in Eqs. (51) and (52) corresponds to the machine precision error at most of the nodes in the domain. Using lower tolerance will decrease the number of operations required to converge and speedup the simulations even more.

6. EUPIC simulation of the plasma flow over a CubeSat in LEO

In the final example, EUPIC is used for the simulation of the plasma flow over a CubeSat shown in Fig. 15 [30,31]. The CubeSat features an instrument for in-situ measurements in the front unit and an electric propulsion system in the

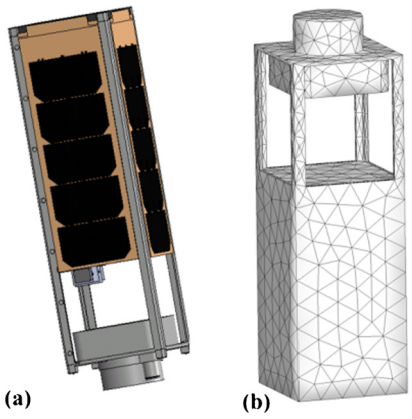


Fig. 15. CubeSat used in EUPIC simulation (a) and EUPIC surface grid (b).

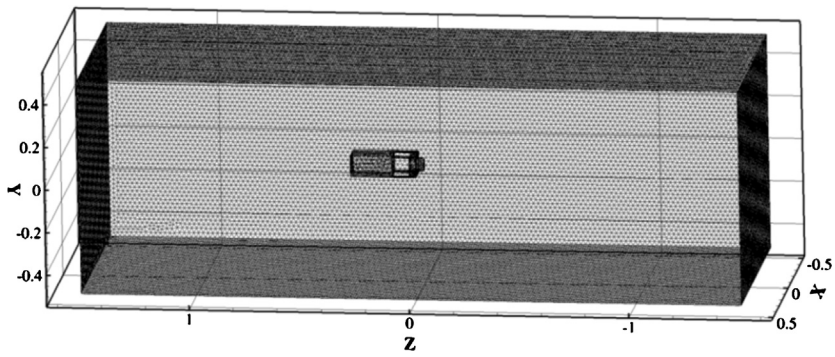


Fig. 16. Computational domain used in the EUPIC simulation of the flow over a CubeSat.

last unit used for attitude and orbital control. In order to evaluate the operation of the instrument it is required to obtain preflight predictions of the incoming fluxes of O^+, H^+, and electrons as well as the distribution of the electric potential around the CubeSat. To accomplish this we simulate the CubeSat with surface mounted solar panel as shown in Fig. 15. In this configuration the geometrical features of the CubeSat include closed surfaces formed by the solar arrays and the aft section which is open and houses the propulsion system. The EUPIC computational domain is a rectangular region of 1 m × 1 m × 3 m shown in Fig. 16 and is discretized with 1.25×10^6 cells which scale with the local Debye length. The plasma flow corresponds to 800-km conditions with $n_{O^+} = 9.06 \times 10^9$ m^{-3}, $n_{H^+} = 2.44 \times 10^9$ m^{-3}, $n_e = 1.23 \times 10^{10}$ m^{-3}, $T_e = 2243.1$ K, $T_{O^+} = T_{H^+} = 1238.5$ K [32]. The particles are injected with velocity $V_x = 0$, $V_y = 0$, $V_z = 7455$ m/s from the left as shown in Fig. 16. These conditions result in a mesothermal ion flow where the thermal speed of electrons $V_{th,e} \gg V_z > V_{th,i}$. The potential boundary conditions are Dirichlet with $\phi = 0$ V on the far field and floating at the CubeSat surface. The simulation was performed with $\Delta t = 10^{-8}$ s and a total of 1.5×10^8 computational particles. The simulation was run until steady state, established by the total masses of the plasma components in the domain. After reaching steady state 200 samples were collected to generate the sampled-averaged fields.

Fig. 17 shows the potential distribution with the formation of a sheath region surrounding the CubeSat. The conducting surfaces reach a floating potential of −0.39 V with respect to unperturbed plasma consistent with simulation results of a 1.5 Unit CubeSat performed with the PIC Spacecraft Plasma Interaction Software (SPIS) [33]. The total charge density in Fig. 18a and Fig. 19a is positive in the vicinity of the CubeSat and shows a negative region in its wake. The electron density in Fig. 18b and Fig. 19b shows a depletion region in the sheath region of the CubeSat and in its near-wake region. These features are qualitatively similar to those from SPIS simulations [33]. Fig. 18c–d and Fig. 19c–d show that both H^+ and O^+ ions show the formation of a wake region and an ion-focusing region which is more prominent for the heavier O^+ ions. These results are consistent with those from 2d and 3d PIC simulations [33,34].

7. Conclusions

This paper presents the mathematical formulation and a parallelized implementation of an electrostatic unstructured Particle-in-Cell method on arbitrary tetrahedral grids (EUPIC). This implementation provides the ability to simulate plasmas in arbitrary geometries using readily available grid generators.

The electric potential in EUPIC is evaluated on cell vertices using a finite volume Multi-Point Flux Approximation of Gauss' law applied to the indirect dual cell. The Dirichlet, Neumann and external circuit boundary conditions

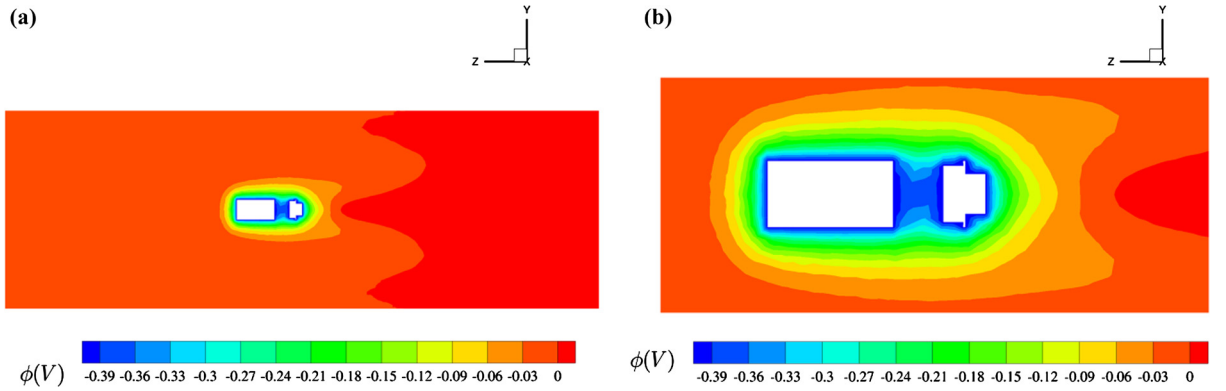


Fig. 17. Potential (V) on the $(X = 0, Y, Z)$ plane from the EUPIC simulation of the plasma flow around a CubeSat in LEO (a) and near the CubeSat surface (b).

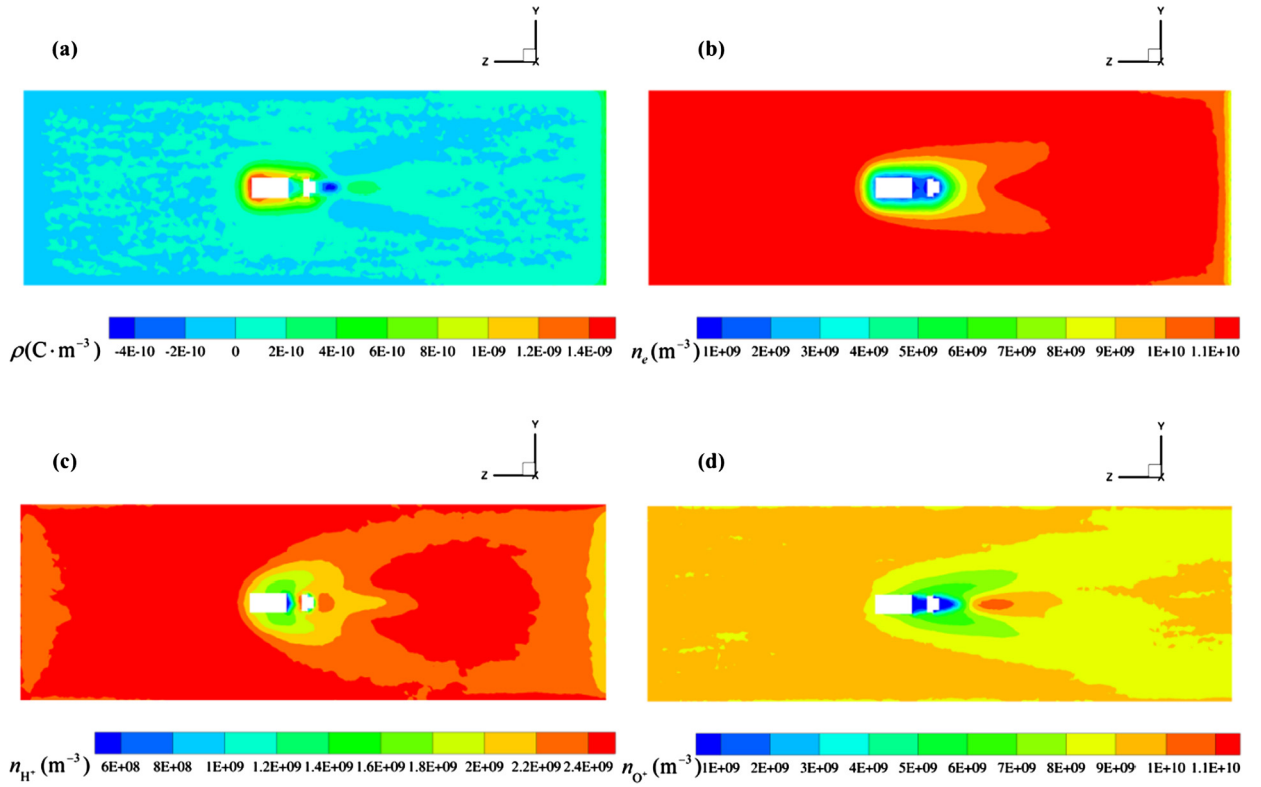


Fig. 18. Flowfield properties on the $(X = 0, Y, Z)$ plane from EUPIC simulation of the plasma flow around a CubeSat in LEO. Total charge density (a), electron number density (b), H^+ number density (c), O^+ number density (d).

are derived by applying Gauss' law to the surface tetrahedra. The resulting non-symmetric system of the sparse linear equations for the nodal potential is solved with a restarted GMRES with ILU(0) preconditioner algorithm. This algorithm is parallelized using a combination of node coloring and level scheduling approaches. The electric field on vertices is obtained using the gradient theorem applied to the indirect dual cell. Boundary conditions and the algorithms for injection, particle loading, particle motion, and particle tracking are implemented for unstructured tetrahedral grids.

The verification and error analysis of EUPIC was performed using analytic solutions of Laplace's equation for a grounded sphere in vacuum with an applied uniform electric field and of Poisson's equation for a stationary ion cloud with linearly varying number density. Grid sensitivity analysis quantifies the L^2 and L^∞ norms of the relative error in potential, field, and charge density as a function of edge-averaged and volume-averaged cell size. Analysis shows second order of convergence for the potential and first order of convergence for the electric field and charge density. The simulation of an electron motion in a stationary ion cloud with linearly varying charge density was used to investigate the energy and mo-

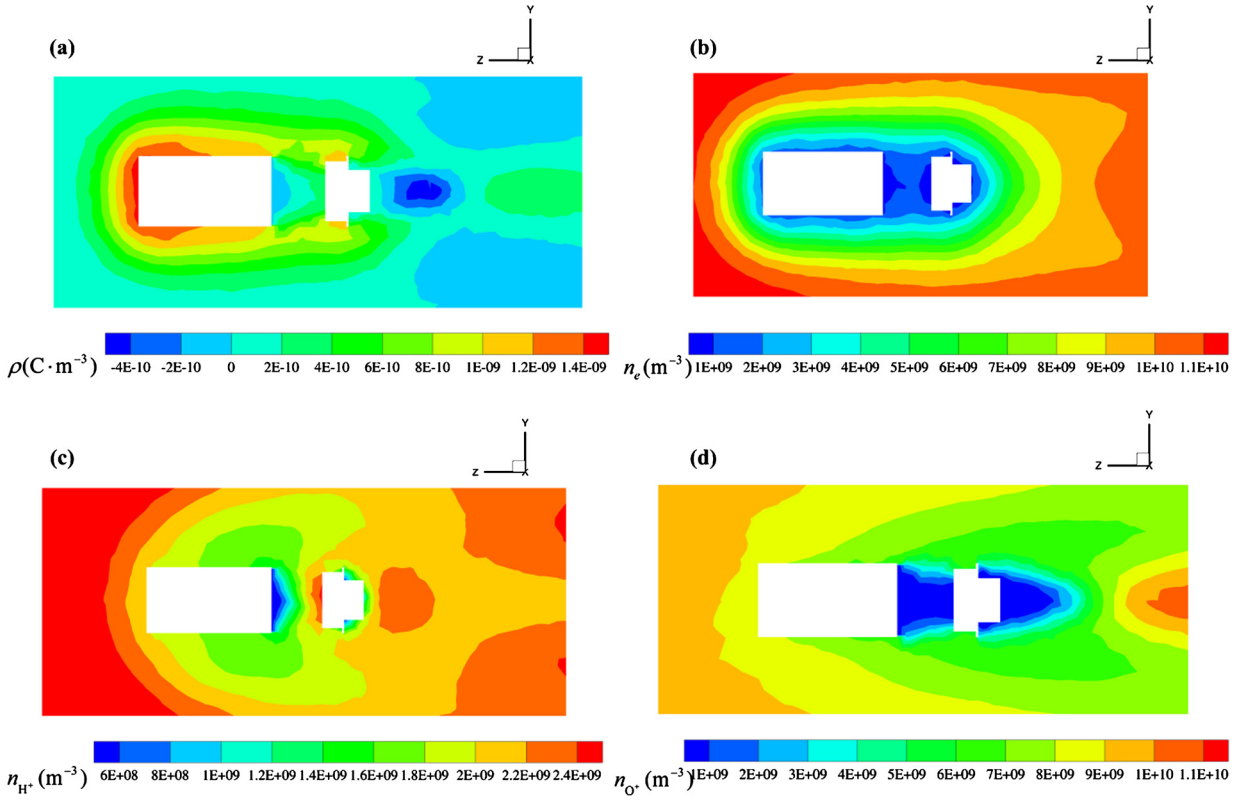


Fig. 19. Flowfield properties on the $(X = 0, Y, Z)$ plane near the CubeSat surface from EUPIC simulation of the plasma flow around a CubeSat in LEO. Total charge density (a), electron number density (b), H^+ number density (c), O^+ number density (d).

mentum conservation properties of the EUPIC and to perform temporal sensitivity analysis. Results show that the motion of an electron is harmonic with almost constant amplitudes of the translational energy and x-component of momentum with minimal dependence on the time step for times up to 113 ω_e^{-1} corresponding to three periods of oscillation. The effects of the average number of particles per cell, grid scaling, and timestep on the numerical heating, the slowing-down time, and the deflection times were investigated by considering the evolution of a fully ionized two-component plasma in a cube with periodic conditions for particles at all sides and zero potential at the boundaries. The results show that heating, slowing down, and deflection times increase almost linearly with increasing average number of computational particles per cell. The cell size as observed by EUPIC simulations has a strong effect on the heating time. Halving a cell size from $\Delta r = \lambda_{De}$ to $\Delta r = 0.5\lambda_{De}$ leads to an order of magnitude increase in the heating time. It is also found that decreasing the time step increases the heating time in a non monotonic fashion. The results show that both slowing down and deflection times are at least three times lower than the heating time. EUPIC simulations of current collection by Langmuir probes are used for validation with Laframboise numerical results which have been validated experimentally [20]. The EUPIC currents are found to be in very good agreement for a wide range of probe to Debye length rations covering the thin sheath to the OML regimes and normalized potential to temperature ratios in the range from -25 to 25. Current collection by a cylindrical Langmuir probe in the thin sheath regime was used to study parallelization efficiency of the EUPIC code. It was found that the performance strongly depends on how threads are bound to CPU cores. For up to 8 cores the parallelization efficiency is more than 80% and it monotonically decreases as the number of threads increases due to NUMA memory design and hyper-threading technology effects. The final test case demonstrates the ability of EUPIC to simulate complex geometrical bodies embedded in flowing multi-species plasmas. The test case involves the EUPIC simulation of the plasma flow over a CubeSat under LEO conditions. EUPIC results show the structure of the ion and electron wakes, the formation of ion enhancement regions in the wake, and the potential structure. The EUPIC results are quantitatively and qualitatively consistent with previous 2D and 3D PIC investigations over simpler geometries.

Acknowledgements

This work was partially funded by AFOSR's Computational Mathematics Program under grants F49620-00-1-0278, F49620-03-1-0219, and FA9550-06-1-0236.

References

[1] J. Wang, P. Liewer, V. Decyk, 3D electromagnetic plasma particle simulations on a MIMD parallel computer, Comput. Phys. Commun. 87 (1) (1995) 35–53.
[2] J. Wang, D. Kondrashov, P.C. Liewer, S.R. Karmesin, Three-dimensional deformable-grid electromagnetic particle-in-cell for parallel computers, J. Plasma Phys. 61 (3) (1999) 367–389.
[3] J. Villasenor, O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, Comput. Phys. Commun. 69 (2–3) (1992) 306–316.
[4] J. Wang, O. Chang, Y. Cao, Electron–ion coupling in mesothermal plasma beam emission: full particle PIC simulations, IEEE Trans. Plasma Sci. 40 (2) (2012) 230–236.
[5] J.-S. Wu, K.-H. Hsu, F.-L. Li, C.-T. Hung, S.-Y. Jou, Development of a parallelized 3D electrostatic PIC-FEM code and its applications, Comput. Phys. Commun. 177 (1–2) (2007) 98–101.
[6] J.J. Petillo, E.M. Nelson, J.F. DeFord, N.J. Dionne, B. Levush, Recent developments to the MICHELLE 2-D/3-D electron gun and collector modeling code, IEEE Trans. Electron Devices 52 (5) (2005) 742–748.
[7] D. Pavarin, F. Ferri, M. Manente, D. Curreli, D. Melazzi, D. Rondini, A. Cardinali, Development of plasma codes for the design of mini-helicon thrusters, in: 32nd International Electric Propulsion Conference, Wiesbaden, Germany, 2011.
[8] A.L. Fabris, C.V. Young, M. Manente, D. Pavarin, M.A. Cappelli, Ion velocimetry measurements and particle-in-cell simulation of a cylindrical cusped plasma accelerator, IEEE Trans. Plasma Sci. 43 (1) (2015) 54–63.
[9] C.S. Meierbachtol, A.D. Greenwood, J.P. Verboncoeur, B. Shanker, Conformal electromagnetic particle in cell: a review, IEEE Trans. Plasma Sci. 43 (11) (2015) 3778–3793.
[10] J.-.L. Vay, P. Colella, J.W. Kwan, P. McCorquodale, D.B. Serafini, A. Friedman, D.P. Grote, G. Westenskow, J.-.C. Adam, A. Héron, I. Haber, Application of adaptive mesh refinement to particle-in-cell simulations of plasmas and beams, Phys. Plasmas 11 (5) (2004) 2928–2934.
[11] R.W. Hockney, J.W. Eastwood, Computer Simulation Using Particles, Taylor & Francis, New York, 1988.
[12] M.T. Bettencourt, Controlling self-force for unstructured Particle-in-Cell (PIC) codes, IEEE Trans. Plasma Sci. 42 (5) (2014) 1189–1194.
[13] N.A. Gatsonis, A. Spirkin, A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay–Voronoi grids, J. Comput. Phys. 228 (10) (2009) 3742–3761.
[14] J. Droniou, Finite volume schemes for diffusion equations: introduction to and review of modern methods, Math. Models Methods Appl. Sci. 24 (8) (2014) 1575–1619.
[15] J.P. Verboncoeur, M.V. Alves, V. Vahedi, C.K. Birdsall, Simultaneous potential and circuit solution for 1D bounded plasma particle simulation codes, J. Comput. Phys. 104 (2) (1993) 321–328.
[16] V. Vahedi, G. DiPeso, Simultaneous potential and circuit solution for two-dimensional bounded plasma simulation codes, J. Comput. Phys. 131 (1) (1997) 149–163.
[17] Y. Saad, Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, 2003.
[18] O. Buneman, Time-reversible difference procedures, J. Comput. Phys. 1 (4) (1967) 517–535.
[19] N.A. Gatsonis, R.E. Chamberlin, S.N. Averkin, An unstructured direct simulation Monte Carlo methodology with kinetic-moment inflow and outflow boundary conditions, J. Comput. Phys. 233 (2013) 148–174.
[20] J.G. Laframboise, Theory of Spherical and Cylindrical Langmuir Probes in a Collisionless, Maxwellian Plasma at Rest, Univ. Toronto Institute of Aerospace Studies, Toronto, 1966.
[21] D.C. Montgomery, D.A. Tidman, Plasma Kinetic Theory, McGraw–Hill, New York, 1964.
[22] S. Ichimaru, Basic Principles of Plasma Physics: A Statistical Approach, Benjamin, New York, 1973.
[23] C. Geuzaine, J.-F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, Int. J. Numer. Methods Eng. 79 (11) (2009) 1309–1331.
[24] J.P. Boris, Relativistic plasma simulation–optimization of a hybrid code, in: Proceedings of the Fourth Conference on Numerical Simulation of Plasma, Washington D.C., 1970.
[25] M. Naumov, P. Castonguay, J. Cohen, Parallel Graph Coloring with Applications to the Incomplete-LU Factorization on the GPU, Nvidia Corp., Santa Clara, CA, 2015.
[26] M. Naumov, M. Arsaev, P. Castonguay, J. Cohen, J. Demouth, J. Eaton, S. Layton, N. Markovskiy, I. Reguly, N. Sakharnykh, V. Sellappan, R. Strzodka, AmgX: a library for GPU accelerated algebraic multigrid and preconditioned iterative methods, SIAM J. Sci. Comput. 37 (5) (2015) S602–S626.
[27] M. Naumov, Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU, Nvidia Corp., Santa Clara, CA, 2011.
[28] J.D. Jackson, Classical Electrodynamics, Wiley, 1999.
[29] R.W. Hockney, Measurements of collision and heating times in a two-dimensional thermal computer plasma, J. Comput. Phys. 8 (1) (1971) 19–44.
[30] N.A. Gatsonis, L. Ye, J.J. Blandino, M.A. Demetriou, N. Paschalidis, Micro pulsed plasma thrusters for attitude control of a low Earth orbiting CubeSat, J. Spacecr. Rockets 53 (1) (2016) 57–73.
[31] J.J. Blandino, N. Martinez, M.A. Demetriou, N.A. Gatsonis, N. Paschalidis, Feasibility for orbital life extension of a CubeSat flying in the lower thermosphere, J. Spacecr. Rockets 53 (5) (2016) 864–875.
[32] International reference ionosphere, NASA, online available: https://iri.gsfc.nasa.gov/ (Accessed 10 May 2017).
[33] R.M. Albarran, A. Barjatya, Plasma density analysis of CubeSat wakes in the Earth's ionosphere, J. Spacecr. Rockets 53 (3) (2016) 393–400.
[34] W.J. Miloch, J. Trulsen, H.L. Pécseli, Numerical studies of ion focusing behind macroscopic obstacles in a supersonic plasma flow, Phys. Rev. E 77 (5) (2008) 056408.