AFRL-AFOSR-VA-TR-2019-0103

A Multiscale Particle Computational Method for Chemically Reacting Microplasmas

Nikolaos Gatsonis
WORCESTER POLYTECHNIC INST MA

03/22/2019
Final Report

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTA2
Arlington, Virginia 22203
Air Force Materiel Command

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 03/11/2019	**2. REPORT TYPE** Final Report	**3. DATES COVERED** *(From - To)* 09/20/2014 - 09/29/2018

4. TITLE AND SUBTITLE A Multiscale Particle Computational Method for Partially Ionized Plasmas	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER FA9550-14-1-0366
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Nikolaos A. Gatsonis Sergey N. Averkin	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Worcester Polytechnic Institute 100 Institute Rd. Worcester, MA, 01609	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 875 N. Randolph Street Suite 325, Room 3112 Arlington, VA 22203-1768	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFOSR
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Distribution A-Approved for public release

13. SUPPLEMENTARY NOTES

14. ABSTRACT

An unstructured electrostatic Particle-In-Cell Collisional (EUPICC) method is developed on arbitrary tetrahedral grids for simulation of partially ionized plasmas bounded by arbitrary geometries. EUPICC is parallelized and includes a multi-weight approach to address collisionless and collisional plasmas with arbitrary differences in species number densities. The electric potential is obtained from a finite volume multi point flux approximation of Gauss' law with Dirichlet, Neumann and external circuit boundary conditions. The matrix equation for the nodal potential is solved with a restarted generalized minimal residual method and a preconditioner algorithm. The electric field is obtained using the gradient theorem. EUPICC includes a real-collision-counter method for collision sampling of particles with arbitrary particle weights and a particle splitting-merge approach that incorporates spread in velocity space. Grid and temporal sensitivity analysis is performed, and the heating, slowing-down, and deflection times are evaluated. An extended set of simulations is performed for validation and verification and assessment of parallelization efficiency. EUPICC is applied to the simulation of a multi-species plasma flow over a nanosatellite and to ion beam neutralization.

15. SUBJECT TERMS
particle in cell;multi-weight collisions;

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Nikolaos A. Gatsonis
U	U	U	UU	46	**19b. TELEPHONE NUMBER** *(Include area code)* 508-831-5576

Reset

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18
Adobe Professional 7.0

INSTRUCTIONS FOR COMPLETING SF 298

1. REPORT DATE. Full publication date, including day, month, if available. Must cite at least the year and be Year 2000 compliant, e.g. 30-06-1998; xx-06-1998; xx-xx-1998.

2. REPORT TYPE. State the type of report, such as final, technical, interim, memorandum, master's thesis, progress, quarterly, research, special, group study, etc.

3. DATES COVERED. Indicate the time during which the work was performed and the report was written, e.g., Jun 1997 - Jun 1998; 1-10 Jun 1996; May - Nov 1998; Nov 1998.

4. TITLE. Enter title and subtitle with volume number and part number, if applicable. On classified documents, enter the title classification in parentheses.

5a. CONTRACT NUMBER. Enter all contract numbers as they appear in the report, e.g. F33615-86-C-5169.

5b. GRANT NUMBER. Enter all grant numbers as they appear in the report, e.g. AFOSR-82-1234.

5c. PROGRAM ELEMENT NUMBER. Enter all program element numbers as they appear in the report, e.g. 61101A.

5d. PROJECT NUMBER. Enter all project numbers as they appear in the report, e.g. 1F665702D1257; ILIR.

5e. TASK NUMBER. Enter all task numbers as they appear in the report, e.g. 05; RF0330201; T4112.

5f. WORK UNIT NUMBER. Enter all work unit numbers as they appear in the report, e.g. 001; AFAPL30480105.

6. AUTHOR(S). Enter name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. The form of entry is the last name, first name, middle initial, and additional qualifiers separated by commas, e.g. Smith, Richard, J, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES). Self-explanatory.

8. PERFORMING ORGANIZATION REPORT NUMBER. Enter all unique alphanumeric report numbers assigned by the performing organization, e.g. BRL-1234; AFWL-TR-85-4017-Vol-21-PT-2.

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES). Enter the name and address of the organization(s) financially responsible for and monitoring the work.

10. SPONSOR/MONITOR'S ACRONYM(S). Enter, if available, e.g. BRL, ARDEC, NADC.

11. SPONSOR/MONITOR'S REPORT NUMBER(S). Enter report number as assigned by the sponsoring/ monitoring agency, if available, e.g. BRL-TR-829; -215.

12. DISTRIBUTION/AVAILABILITY STATEMENT. Use agency-mandated availability statements to indicate the public availability or distribution limitations of the report. If additional limitations/ restrictions or special markings are indicated, follow agency authorization procedures, e.g. RD/FRD, PROPIN, ITAR, etc. Include copyright information.

13. SUPPLEMENTARY NOTES. Enter information not included elsewhere such as: prepared in cooperation with; translation of; report supersedes; old edition number, etc.

14. ABSTRACT. A brief (approximately 200 words) factual summary of the most significant information.

15. SUBJECT TERMS. Key words or phrases identifying major concepts in the report.

16. SECURITY CLASSIFICATION. Enter security classification in accordance with security classification regulations, e.g. U, C, S, etc. If this form contains classified information, stamp classification level on the top and bottom of this page.

17. LIMITATION OF ABSTRACT. This block must be completed to assign a distribution limitation to the abstract. Enter UU (Unclassified Unlimited) or SAR (Same as Report). An entry in this block is necessary if the abstract is to be limited.

SUMMARY

Partially ionized and bounded plasmas from microscale to macroscale find diverse applications of interest to Air Force and other communities. Modeling and simulation with particle-based approaches require the development of new computational methods that address the presence of complex geometries, collisions, multiple scales, coupling between volume, surface, and transport non-equilibrium processes, and presence of species populations with vastly different number densities. The goals and objectives of research performed under FA9550-14-1-0366 were to:

- Develop a particle-based multiscale computational method for the simulation of partially ionized bounded plasmas and include methods for multi-weight collisions, coupling with external circuits, and parallelization.
- Validate the method and perform error analysis.
- Apply to the investigation of plasmas of interest to Air Force.

The mathematical and computational accomplishments under FA9550-14-1-0366 are:

- Developed an Electrostatic Unstructured Particle in Cell Collisional (EUPICC) method for unstructured tetrahedral meshes and implemented onto a code EUPICC.
- Developed and implemented gather/scatter, particle motion and tracing algorithms.
- Developed and implemented a new potential solver and electric filed evaluation with a Finite Volume Multi-point Flux Approximation (MFPA) of the integral form of Gauss' Law.
- Developed and implemented a correction method for electric fields on boundaries.
- Developed and implemented a potential solver based on GMRES with ILU(0) preconditioner using the level-scheduling and data dependency directed acyclic graph coloring approach. Implemented static number of operations analyzer to improve parallel efficiency.
- Developed and implemented a new approach for driven conductors by external circuits.
- Developed and implemented a multi weight collision approach for elastic collisions. It includes a new real-collision-counter method for collision sampling of particles and a particle splitting-merge method that is combined with spread in velocity space.
- Parallelized EUPICC with OpenMP and implemented load balancing on most parts of the EUPICC cycle.
- Performed validation-verification and error analysis: compared with analytic solutions; performed grid sensitivity and order of convergence analysis; investigated momentum and energy conservation of time inetrgators; performed two-stream instability simulations.
- Evaluated effects of times step, grid-size and particle/cell on heating, deflection, and slowing-down times.
- Performed high fidelity simulations of current collection of collisionless cylindrical Langmuir probes and compared with previous validated numerical results.
- Performed high-fidelity simulations of: plasma flows over realistic nanosats and plasma conditions in low earth orbit; ion beam neutralization phenomena.

The research has significant scientific and technical merit in computational mathematics. It addresses a gap in the current ability to simulate partially ionized plasmas bounded by surfaces of arbitrary geometrical complexity. The research is relevant with the interests and mission of the Air Force and has broader impacts for the US Defense. The EUPICC computation method allows the simulation of partially ionized plasmas found in areas of plasma/spacecraft interactions, plasma-aided materials processing, plasma discharges, and plasma propulsion.

1 A MUTISCALE PARTICLE COMPUTATIONAL METHOD FOR PARTIALLY IONIZED PLASMAS

Partially ionized plasmas appear in many applications of interest, such as plasma/spacecraft interactions, plasma-aided materials processing, plasmas discharges, and plasma propulsion. Simulation methods for a partially ionized, electrostatic, magnetized flow can be categorized as the Particle-In-cell/Monte Carlo Collisions (PIC-MCC) (Birdsall, 1991) and PIC-Direct Simulation Monte Carlo (PIC-DSMC) methods (Serikov et al., 1999; Nanbu, 2000; Gatsonis and Yin, 2001; Taccogna et al, 2012; Chanrion and Neubert , 2008; Kolev . et al., 2009). The PIC component of the combined PIC-MCC or PIC-DSMC approaches has been the method of choice for fully ionized, collisionless, electrostatic, magnetized plasmas (Birdsall and Langdon, 1991; Hockney and Eastwood, 1988). The PIC approach has its mathematical equivalent in the Vlasov-Poisson system and has fundamental computational limitations related to Debye lengths and electron time steps. The DSMC method was first derived by Bird (1994), is the method of choice for simulation of neutral gas flows with elastic and inelastic collisions as described by the Boltzmann equation. Both PIC and DSMC methods use computational particles, each representing a large number of real particles via the particle weight and use similar representations of the velocity distribution functions (Nanbu,2000). The neutrals in the majority of PIC-MCC applications are not modeled as particles. The PIC-MCC method is strictly applicable to low plasma densities ($n_e \le 10^{-16}$ m^{-3}) due to Debye length limitations and to problems where neutral densities dominate the flow. Several variations of PIC-DSMC have emerged over the years and applied to partially ionized plasmas (Serikov et al, 1999; Nanbu, 2000; Taccogna et al, 2012) The fundamental difference between these two approaches lies in the inclusion of neutral dynamics (Nanbu, 2000). In PIC-MCC the neutrals are considered to be background with known parameters. The collisions between charged species and neutrals are modeled by null-collision scheme (Birdsall, 1991) or its variation (Nanbu, 2000) and neutral interactions are disregarded. In contrast in PIC-DSMC the neutrals are included in the model and their collisions are executed at each time step. The electron-neutral collisions follow the same procedures as in PIC-MCC. Heavy particle collisions are executed by following DSMC procedures. An outstanding issue in the simulation of collisional plasmas in general, is the algorithmic implementation of charged-particle collisions which are fundamentally different from ion-neutral and neutral-neutral collisions due to long-range interaction forces. Nanbu (1997) presented a method that can be applied to Monte Carlo simulations with the introduction of a cumulative scattering angle that represents a succession of small-angle binary collisions. Dimarco et al (2009) extended the DSMC approach to take into account charged particles collisions.

In several applications of interest species in a partially ionized plasmas often have orders of magnitude differences in their number densities. In such cases a single-weight representation cannot describe accurately the collisional interactions (elastic and non-elastic) and the collision algorithms in PIC-DSMC methods require new multi-weight approaches.

We present in this report an unstructured collisional electrostatic Particle-In-Cell (EUPICC) method on arbitrary tetrahedral grids for simulation of partially ionized plasmas bounded by arbitrary geometries. Section 1.1 presents the algorithmic elements of the collisionless EUPICC method where each species is represented by its own weight: particle loading; charge assignment and electric field evaluation; particle motion; evaluation of macroscopic variables. In Section 1.2 we introduce a new multi-weight collision approach based on a real-collision-counter method for collision sampling of particles and a particle splitting-merge approach that is combined with spread in velocity space. This multi-weight approach is implemented for neutral-neutral collisions. In Section 1.3 we present high-fidelity simulations enable by EUPICC. The work presented in this report was developed under FA9550-14-1-0366 and appeared in Averkin and Gatsonis (2018), Averkin et al. (2018), Han et al. (2018), Averkin et al (2017), and Han et al. (2017).

1.1 Mathematical and Computational Aspects of the Electrostatic Unstructured Particle in Cell Collisional (EUPICC) Methodology

1.1.1 *State-of-the art in Unstructured PIC/DSMC*

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. (1995; 1999). 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 (1992). The code has been applied to ion beam neutralization by Wang et al (2012). Wu et al. (2007) developed a 3d finite-element, parallelized, electrostatic code using unstructured tetrahedral grids with dynamic domain decomposition. Petillo et al. (2005) developed a finite-element electromagnetic PIC code on structured and unstructured grids and applied to electron guns. A 3D 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. (2011) and applied to the simulation of a cylindrical cusped-plasma accelerator by Fabris et al (2015). Alternative approaches addressing complex geometries without using unstructured meshes include conformal mapping (Meierbachtol et al. 2004) and adaptive mesh refinement (Vay, et al., 2004. Unstructured PIC simulations have the potential for increased artificial collisions and numerical heating due to the exerted self-force on particles (Hockney and Eastwood, 1988). Bettencourt (2014) suggested an algorithm that allows controlling the amount of the self-force. Gatsonis and Spirkin (2009) 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 effectively in 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 (EUPICC). All algorithms of the EUPICC are parallelized and implemented using OpenMP methodology allowing large-scale plasma computations with complex geometrical domains on multiprocessors. The grid structure used in EUPICC 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 Gatsonis and Spirkin (2009). The evaluation of the electric potential follows the finite-volume formulation of the integral Gauss law 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 (Droniou, 2014). 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 (Gatsonis and Spirkin, 2009; Verboncoeur et al., 1997) during an EUPICC 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 (Saad, 2003). GMRES is parallelized in EUPICC 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 (Gatsonis and Spirkin, 2009). 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 EUPICC follows Buneman's time-centered leapfrog formulation Buneman, 1967. The particle search-locate algorithm is performed using an optimized version of an algorithm developed for Direct Simulation Monte Carlo method on tetrahedral (Gatsonis et al., 2013). Particle injection from surfaces in EUPICC follows (Gatsonis and Spirkin, 2009). Periodic boundary conditions are also implemented in EUPICC 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 weight. The methods and algorithms in EUPICC 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 (1966) results on current collection by cylindrical Langmuir probes in collisionless plasmas are used for verification of EUPICC simulations in the orbital motion limited and thin-sheath regimes. Finally, EUPICC is used for a simulation of the plasma flow around a CubeSat in Low Earth Orbit.

1.1.2 *The Vlasov-Poisson System in EUPICC*

For collisionless plasmas, the EUPICC method solves the electrostatic Vlasov-Poisson system for a multi-species fully ionized plasma (Montgomery & Tidman, 1964; Ichimaru, 1973). The single-particle distribution function for species s particles, gives the average number of particles in a volume $d^3rd^3v \equiv d\mathbf{r}d\mathbf{v}$ 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,

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

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 $\phi(\mathbf{r},t)$

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

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

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

or to the differential Poisson's equation

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

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

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

1.1.3 *Grid Generation and Data Structure*

The EUPICC method uses an unstructured three-dimensional mesh with tetrahedral cells obtained through a grid generator such as GMSH (Geuzaine & Remacle, 2009). The EUPICC 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 Gatsonis and Spirkin (2009). A tetrahedral cell is illustrated in Figure 1 and assigned to a global cell index $D=1,..,G_D$, 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 $d=1,..,G_d$, 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 $\sigma=1,..,G_\sigma$, 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 $\{i,j,k,l\}\in\{1,...,G_d\}$. 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 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 $\Delta\mathbf{r}_{i,j}=\mathbf{r}_i-\mathbf{r}_j$.

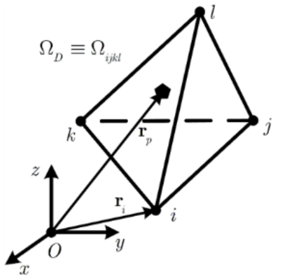
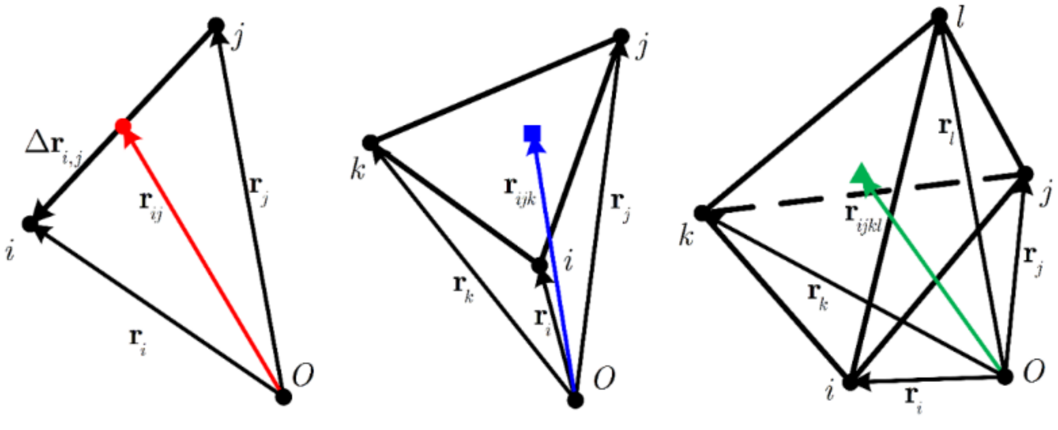


Figure 1. The tetrahedron Ω_D with global indexing in a domain Ω.

Centroids used in various aspects of an EUPICC cycle are depicted in Figure 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

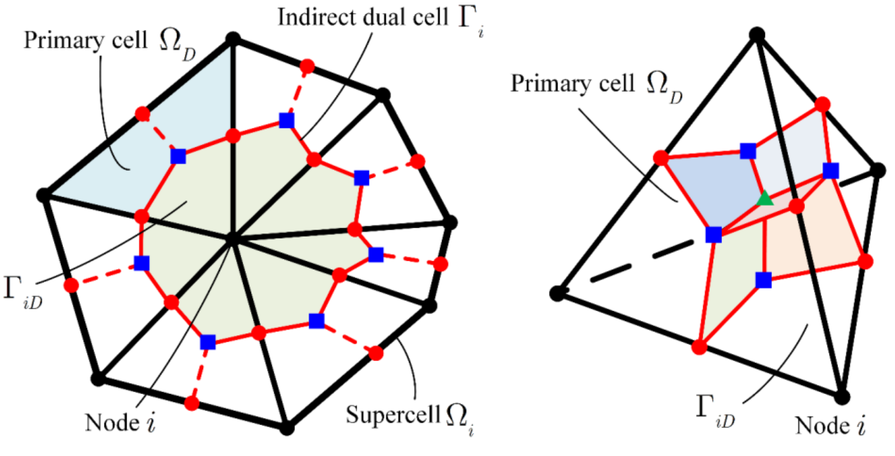
$$\begin{aligned} \mathbf{r}_{ij} &= \frac{1}{2}\left(\mathbf{r}_i + \mathbf{r}_j\right), \\ \mathbf{r}_{ijk} &= \frac{1}{3}\left(\mathbf{r}_i + \mathbf{r}_j + \mathbf{r}_k\right), \\ \mathbf{r}_{ijkl} &= \frac{1}{4}\left(\mathbf{r}_i + \mathbf{r}_j + \mathbf{r}_k + \mathbf{r}_l\right). \end{aligned}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



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

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

A triangular face of the cell Ω_{ijkl} Ω_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 $\mathbf{n}_{Dl}$.



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

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

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

The indirect dual cell used in Poisson's solver is shown in Figure 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 Figure 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 Ω_D (Ω_{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 A_{Dik} and can be written as

$$A_{Dik} = \Gamma_{iD} \cap \Gamma_{kD} . \tag{9}$$

The area of the face A_{Dik} is denoted by $\left|A_{Dik}\right|$. 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 $A_{Dki} \equiv A_{Dik}$ of the cell Γ_{kD}.

The data structure implemented in EUPICC allows evaluation of information on neighboring cells during particle motion without any additional operations as discussed in Sec.1.1.8.

1.1.4 *Charge Assignment, Nodal Number Density and Nodal Charge Density*

The particle loading and injection procedures in EUPICC follow those in Gatsonis and Spirkin (2009). The charge assignment functions used in EUPICC follow also Gatsonis and Spirkin (2009) 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

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

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

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

Using weights given by Eq. (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 Figure 3(a), as

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

1.1.5 *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 Figure 3a for 2D configuration yields

$$\sum_{D=1}^{G_D(i)} \sum_{\substack{k \in \Omega_D \\ k \neq i}} \iint_{\hat{A}_{Dik}} \nabla\phi \cdot d\mathbf{A} = -\frac{Q_i}{\epsilon_0}, \tag{15}$$

where $G_D(i)$ denotes the number of cells which share the same node i. Assuming furthermore a linear potential variation inside each tetrahedron $\Omega_D \cap \Gamma_i \neq \varnothing$, 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 equation

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

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

$$\mathbf{E}_D = -\nabla\phi_{\,D} = \frac{1}{3|\Omega_D|} \phi_i \mathbf{n}_{Di} |A_{Di}| + \phi_j \mathbf{n}_{Dj} |A_{Dj}| + \phi_k \mathbf{n}_{Dk} |A_{Dk}| + \phi_l \mathbf{n}_{Dl} |A_{Dl}| . \tag{17}$$

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

$$\sum_{D=1}^{G_D(i)} \nabla\phi_{\,D} \cdot \sum_{\substack{k \in \Omega_D \\ k \neq i}} \hat{\mathbf{n}}_{Dik} |A_{Dik}| = -\frac{Q_i}{\epsilon_0} . \tag{18}$$

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

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

and can be simplified to

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

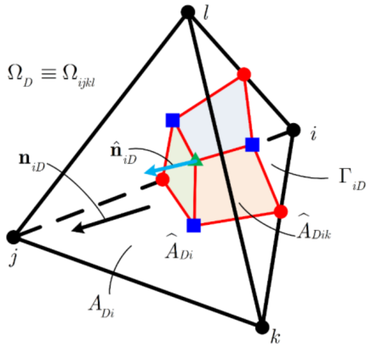


Figure 4. Γ_{iD} **is the part Gaussian surface** Γ_i **used in the evaluation of the electric potential.**

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

$$\sum_{D=1}^{G_D(i)} \frac{1}{9\left|\Omega_D\right|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk} \left|A_{Dk}\right| \cdot \mathbf{n}_{Di} \left|A_{Di}\right| = \frac{Q_i}{\epsilon_0} \tag{21}$$

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

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} & \cdots & R_{1G_d} \\ R_{21} & R_{22} & R_{23} & \cdots & R_{2G_d} \\ R_{31} & R_{32} & R_{33} & \cdots & R_{3G_d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{G_d 1} & R_{G_d 2} & R_{G_d 3} & \cdots & R_{G_d G_d} \end{pmatrix} \begin{pmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \vdots \\ \phi_{G_d} \end{pmatrix} = \begin{pmatrix} Q_1/\epsilon_0 \\ Q_2/\epsilon_0 \\ Q_3/\epsilon_0 \\ \vdots \\ Q_{G_d}/\epsilon_0 \end{pmatrix}, \tag{22}$$

where

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

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 (Droniou, 2014).

1.1.5.1 *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 Figure 5(a). The potential at the Dirichlet node $i \in A_{\mathrm{Di}}$ of a surface A_{Di} is

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

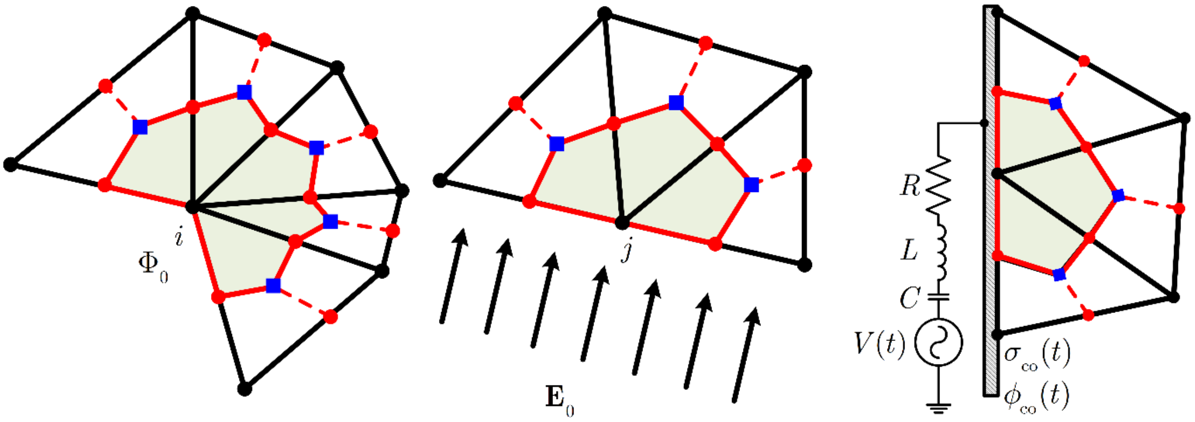
In the case of Neumann conditions the normal field at node $j \in A_{\mathrm{Ne}}$ of a surface A_{Ne} is

$$\mathbf{E}_{j}{}_{n} = \mathbf{E}_{o}{}_{n} . \tag{24}$$

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

$$\sum_{D=1}^{G_D(i)} \frac{1}{9\left|\Omega_D\right|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk} \left|A_{Dk}\right| \cdot \mathbf{n}_{Dj} \left|A_{Dj}\right| + \frac{1}{3} \mathbf{E}_0 \cdot \sum_{\substack{\sigma\in A_{\mathrm{Ne}} \\ A_\sigma \ni j}} \mathbf{n}_\sigma \left|A_\sigma\right| = \frac{Q_j}{\epsilon_0}, \tag{25}$$

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.



(a) Surface with Dirichlet (Φ_0) boundary conditions with corresponding Gaussian surface

(b) Surface with Neumann ($\mathbf{E}_0$) boundary conditions with corresponding Gaussian surface

(c) Surface with conductor driven by an LRC circuit with corresponding Gaussian surface

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

1.1.5.2 *Plasma with a Conductor Driven by an LRC Circuit*

Consider next a bounded plasma that includes a driven conductor as shown in Figure 5(c). The conductor of surface $A_{\rm co}$ is driven by a voltage source $V(t)$ connected in series with a resistance, R, inductance, L, and capacitance, C, as shown in Figure 5(c). The evaluation of the potential generalizes the approach in Gatsonis and Spirkin (2009) following Verboncoeur et al. (1993) and Vahdeiand DiPeso (1997). The integral Gauss' law Eq. (3) applied to the dual cell associated with a node $i \in A_{\rm co}$ shown in Figure 5(c) becomes

$$\epsilon_0 \sum_{D=1}^{G_D(i)} \frac{1}{9\left|\Omega_D\right|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk}\left|A_{Dk}\right| \cdot \mathbf{n}_{Di}\left|A_{Di}\right| - \sigma_i \sum_{\substack{A_\sigma \in A_{\rm co} \\ A_\sigma \ni i}} \left|A_\sigma\right| = Q_i, \tag{26}$$

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 $\phi_{\rm co}$, the total surface charge density $\sigma_{\rm co}(t)$ on the driven conductor with total area $\left|A_{\rm co}\right|$ is obtained by summing all surface charge densities as

$$\begin{aligned} \sigma_{\rm co}\left|A_{\rm co}\right| &\equiv \sum_{i\in A_{\rm co}} \sigma_i \sum_{\substack{A_\sigma \in A_{\rm co} \\ A_\sigma \ni i}} \left|A_\sigma\right| \\ &= \epsilon_0 \phi_{\rm co} \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{\left|A_{Di}\right|^2}{9\left|\Omega_D\right|} + \epsilon_0 \sum_{i\in A_{\rm co}} \sum_{D=1}^{G_D(i)} \frac{1}{9\left|\Omega_D\right|} \sum_{\substack{k\in\Omega_D \\ k\neq i}} \phi_k \mathbf{n}_{Dk}\left|A_{Dk}\right| \cdot \mathbf{n}_{Di}\left|A_{Di}\right| - \sum_{i\in A_{\rm co}} Q_i, \end{aligned} \tag{27}$$

where $\sum_{i\in A_{\rm co}} Q_i$ is the volume charge associated with the plasma in the dual cells of the conductor. In Eq. (27) $\phi_{\rm co}(t)$ and $\sigma_{\rm co}(t)$ 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

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

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 Figure 5(c) is

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

Substituting a second-order, backward Euler finite-difference approximation of Eq. (29) (Verboncoeur et al., 1993) into a first-order, backward Euler finite-difference approximation of Eq. (28) it becomes

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

where

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

and $\Delta Q_{\mathrm{cv}}(t)=Q_{\mathrm{cv}}(t)-Q_{\mathrm{cv}}(t-\Delta t)$ 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

$$\begin{aligned}&\left[\epsilon_0\sum_{i\in A_{\mathrm{co}}}\sum_{D=1}^{G_D(i)}\frac{\left|A_{Di}\right|^2}{9\left|\Omega_D\right|}-\left(\frac{9L}{4\Delta t^2}+\frac{3R}{2\Delta t}+C\right)^{-1}\right]\phi_{\mathrm{co}}(t)+\epsilon_0\sum_{i\in A_{\mathrm{co}}}\sum_{D=1}^{G_D(i)}\frac{1}{9\left|\Omega_D\right|}\sum_{\substack{k\in\Omega_D\\ k\neq i}}\phi_k\mathbf{n}_{Dk}\left|A_{Dk}\right|\cdot\mathbf{n}_{Di}\left|A_{Di}\right|\\ &=\left|A_{\mathrm{co}}\right|\sigma_{\mathrm{co}}(t-\Delta t)+\Delta Q_{\mathrm{cv}}(t)+\sum_{i\in A_{\mathrm{co}}}Q_i+\ \Delta Q_{\mathrm{ci}}(t-\Delta t)+V(t)\ \left(\frac{9L}{4\Delta t^2}+\frac{3R}{2\Delta t}+C\right)^{-1}.\end{aligned} \tag{32}$$

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_{Ne} with an applied electric field $\mathbf{E}_0$ then Eq. (32) coupled with the system (22) for the interior nodes results in

$$\begin{pmatrix} R_{11} & \cdots & R_{1i} & \cdots & R_{1j} & \cdots & R_{1G_d} & 0\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots\\ 0 & \cdots & 1 & \cdots & 0 & \cdots & 0 & -1\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots\\ 0 & \cdots & 0 & \cdots & 1 & \cdots & 0 & 0\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots\\ R_{k1} & \cdots & R_{ki} & \cdots & R_{kj} & \cdots & R_{kG_d} & 0\\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots\\ R_{G_d1} & \cdots & R_{G_di} & \cdots & R_{G_dj} & \cdots & R_{G_dG_d} & 0\\ \tilde{R}_1 & \cdots & \tilde{R}_i & \cdots & \tilde{R}_j & \cdots & \tilde{R}_{G_d} & 1 \end{pmatrix}\begin{pmatrix}\phi_1\\ \vdots\\ \phi_i\\ \vdots\\ \phi_j\\ \vdots\\ \phi_k\\ \vdots\\ \phi_{G_d}\\ \phi_{\mathrm{co}}\end{pmatrix}=\frac{1}{\epsilon_0}\begin{pmatrix}Q_1\\ \vdots\\ 0\\ \vdots\\ \epsilon_0\Phi_0\\ \vdots\\ Q_k-\dfrac{1}{3}\epsilon_0\mathbf{E}_0\cdot\sum_{\substack{\sigma\in A_{\mathrm{Ne}}\\ A_\sigma\ni k}}\mathbf{n}_\sigma\left|A_\sigma\right|\\ \vdots\\ Q_{G_d}\\ \tilde{Q}\end{pmatrix}, \tag{33}$$

where the i th row corresponds to the nodes of the driven conductor A_{co}, the row j is for the conductor at fixed potential Φ_0, the k th row is for the Neumann boundary A_{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

$$\tilde{R}_i=\begin{cases}\left[\epsilon_0\sum_{j\in A_{\mathrm{co}}}\sum_{D=1}^{G_D(j)}\frac{\left|A_{Dj}\right|^2}{9\left|\Omega_D\right|}-\left(\frac{9L}{4\Delta t^2}+\frac{3R}{2\Delta t}+C\right)^{-1}\right]^{-1}\left(\epsilon_0\sum_{\substack{D=1\\ j\in\Omega_D\\ j\neq i}}^{G_D(i)}\frac{1}{9\left|\Omega_D\right|}\mathbf{n}_{Di}\left|A_{Di}\right|\cdot\mathbf{n}_{Dj}\left|A_{Dj}\right|\right), i\in A_{\mathrm{co}},\\ 0, i\notin A_{\mathrm{co}}\end{cases},$$

$$\tilde{Q}=\frac{\epsilon_0\left|A_{\mathrm{co}}\right|\sigma_{\mathrm{co}}(t-\Delta t)+\epsilon_0\Delta Q_{\mathrm{cv}}(t)+\epsilon_0\sum_{i\in A_{\mathrm{co}}}Q_i(t)+\epsilon_0\dfrac{\Delta Q_{\mathrm{ci}}(t-\Delta t)+V(t)}{\dfrac{9L}{4\Delta t^2}+\dfrac{3R}{2\Delta t}+C}}{\epsilon_0\sum_{i\in A_{\mathrm{co}}}\sum_{D=1}^{G_D(i)}\dfrac{\left|A_{Di}\right|^2}{9\left|\Omega_D\right|}-\left(\dfrac{9L}{4\Delta t^2}+\dfrac{3R}{2\Delta t}+C\right)^{-1}}.$$

The system of sparse linear equations given by Eq. (33) is solved using restarted GMRES solver with the ILU(0) preconditioner following Saad (2003). 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 Figure 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 Eq. (28) in discrete form becomes

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

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

$$\begin{aligned}&\epsilon_0\sum_{i\in A_{\mathrm{co}}}\sum_{D=1}^{G_D(i)}\frac{\left|A_{Di}\right|^2}{9\left|\Omega_D\right|}\phi_{\mathrm{co}}(t)+\epsilon_0\sum_{i\in A_{\mathrm{co}}}\sum_{D=1}^{G_D(i)}\frac{1}{9\left|\Omega_D\right|}\sum_{\substack{k\in\Omega_D\\ k\neq i}}\phi_k\mathbf{n}_{Dk}\left|A_{Dk}\right|\cdot\mathbf{n}_{Di}\left|A_{Di}\right|\\ &=\left|A_{\mathrm{co}}\right|\sigma_{\mathrm{co}}(t-\Delta t)+\Delta Q_{\mathrm{cv}}(t)+\sum_{i\in A_{\mathrm{co}}}Q_i\end{aligned}\tag{35}$$

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

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

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

1.1.6 *Particle Injection from Surfaces and Periodic Boundary Conditions on Unstructured Meshes*

Particle injection from surfaces follows Gatsonis and Spirkin (2009). Periodic boundary conditions are also implemented in EUPICC for a number of periodic surfaces pairs obtained by translation by a vector $\mathbf{R}$, 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 $\mathbf{R}$ is stored. When a particle crosses a periodic face, its position is shifted by $\mathbf{R}$ 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. 1.1.5.1.

1.1.7 *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 Figure 3

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

or after carrying out all integrations in fully discrete form

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

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 Eq. (24). For a Dirichlet node or driven-circuit node $i \in A_{\mathrm{Di}}$ the electric field is given by

$$\mathbf{E}_i = \frac{\sigma_i}{\epsilon_0} \mathbf{n}_i , \tag{39}$$

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

$$\mathbf{E}_i = -\frac{1}{3} \frac{\sigma_i}{\epsilon_0} \sum_{\substack{\sigma \in A_{\mathrm{Di}} \\ A_\sigma \ni j}} \mathbf{n}_\sigma \left|A_\sigma\right| . \tag{40}$$

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

$$\sigma_i = \left(\epsilon_0 \sum_{D=1}^{G_D(i)} \frac{1}{9\left|\Omega_D\right|} \sum_{k\in\Omega_D} \phi_k \mathbf{n}_{Dk} \left|A_{Dk}\right| \cdot \mathbf{n}_{Di} \left|A_{Di}\right| - Q_i \right) \Big/ \left(\sum_{\substack{A_\sigma \in A_{\mathrm{Di}} \\ A_\sigma \ni i}} \left|A_\sigma\right| \right). \tag{41}$$

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 Eq. (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. 1.1.4 as

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

1.1.8 *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, $\mathbf{B}_{\mathrm{ext}}$, is integrated as in Gatsonis and Spirkin (2009) following Buneman's time-centered leapfrog formulation (Buneman, 1967)

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

combined with Boris's (1970) algorithm for the implementation of the $\mathbf{v}\times\mathbf{B}_{\mathrm{ext}}$ rotation in case of an applied external magnetic field.

The particle search-locate algorithm in EUPICC is carried out using an improved version of the successive-neighbor algorithm developed in Gatsonis et al. (2013) for Direct Simulation Monte Carlo method on unstructured three-dimensional meshes. The current algorithm requires less number of floating point operations. The steps for particles tracking in tetrahedra shown in Figure 6 are outlined below:

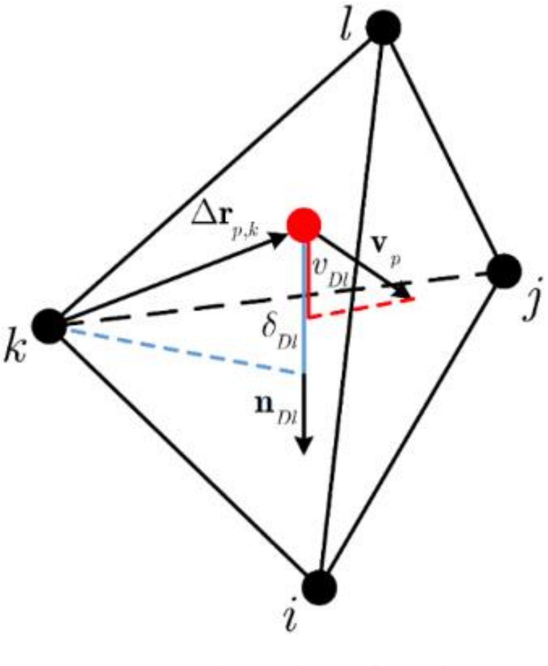


Figure 6. Particle tracking algorithm.

(S.1) For each tetrahedral cell D using the outward normal $\mathbf{n}_{Dl}$ for each face 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 $\delta t_{Dl} = 1 + \Delta t$

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

1.2. 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 Figure 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, $\delta t_{Dl} = \delta_{Dl} / v_{Dl}$.

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

2.1. If $\delta t_{\min} \leq \Delta t$ 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:

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.

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

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

1.1.9 *Nodal Macroscopic Properties*

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

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

where the number of real particles of species s inside a supercell Ω_i shown in Figure 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

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

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

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

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

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

1.1.10 *Parallelization Implementation*

The EUPICC 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 (Naumov et al., 2015) and calculation ordering based on the level-scheduling scheme (Naumov et al., 2015b). 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 EUPICC 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

$$e^{n}_{\mathrm{GMRES}} = \frac{\left\|r_n\right\|}{\max\ \epsilon_0, \left\|\tilde{Q}\right\|} \leq \epsilon\ . \tag{51}$$

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, $\left\|\ldots\right\|$ is the either L^2 or L^∞ norm, ϵ_0 is the constant that determines the minimum possible value of $\left\|\tilde{Q}\right\|$ 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

$$\bar{e}^{m}_{\mathrm{GMRES}} = \min\left(\frac{\left\|M^{-1}r_m\right\|}{\max\ \epsilon_0, \left\|M^{-1}\tilde{Q}\right\|}, \frac{\left\|r_0\right\|}{\left\|M^{-1}r_0\right\|}\frac{\left\|M^{-1}r_m\right\|}{\max\ \epsilon_0, \left\|\tilde{Q}\right\|}\right) \leq \epsilon\ , \tag{52}$$

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 $\epsilon = 10^{-11}, \epsilon_0 = 10^{-13}$ 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 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 by Eq. (40).

1.1.11 *Parallelization Efficiency*

In this section the parallelization efficiency of the OpenMP implementation in EUPICC 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 2 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 EUPICC 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%. Figure 7 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 Figure 7, the performance drops due to the non-uniform memory access (NUMA) architecture and particle data structure in the EUPICC code. The particle data in EUPICC 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 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 Figure 7. 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 Figure 7. 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 Figure 7. 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 interfaces (MPI) for each CPU of the node. This strategy will also allow to run simulations on multiple nodes.

Table 1. The average iteration time in the EUPICC simulation of the current collection by a cylindrical Langmuir probe and the distribution of different

Number of threads	Average time per iteration (sec)	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

To better evaluate the impact of memory access on operations Table 1 presents the averaged timestep 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 (Naumov et al., 2015). 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.

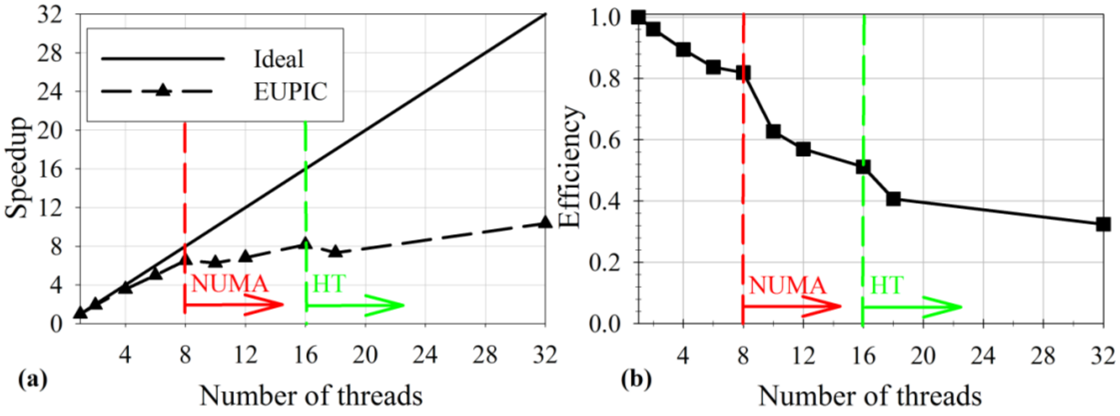


Figure 7. Speedup (a) and parallel efficiency (b) of the EUPICC simulation of the current collection by a cylindrical Langmuir probe in the thing sheath regime as a function of the number of OpenMP threads. The red line shows the number of threads after which NUMA effects start playing the role. The green line shows the number of threads after which the hyper-threading technology is used.

1.1.12 *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 $\mathbf{E}(x,y,z=5R)=\mathbf{E}_o$. The analytic solution (Jackson, 1999) 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. Figure 8 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 $\|f\|_\infty=\max_i|f_i|$. Figure 8 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 $\Delta r=\sum_{i=1}^{G_d} l_i \big/ G_d$ and a volume-averaged with $\Delta r=(6\sqrt{2}/|\bar{\Omega}|)^{1/3}$ and $|\bar{\Omega}|=\sum_{i=1}^{G_d}|\Omega_i| \big/ G_d$. Figure 8 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 (Droniou, 2014).

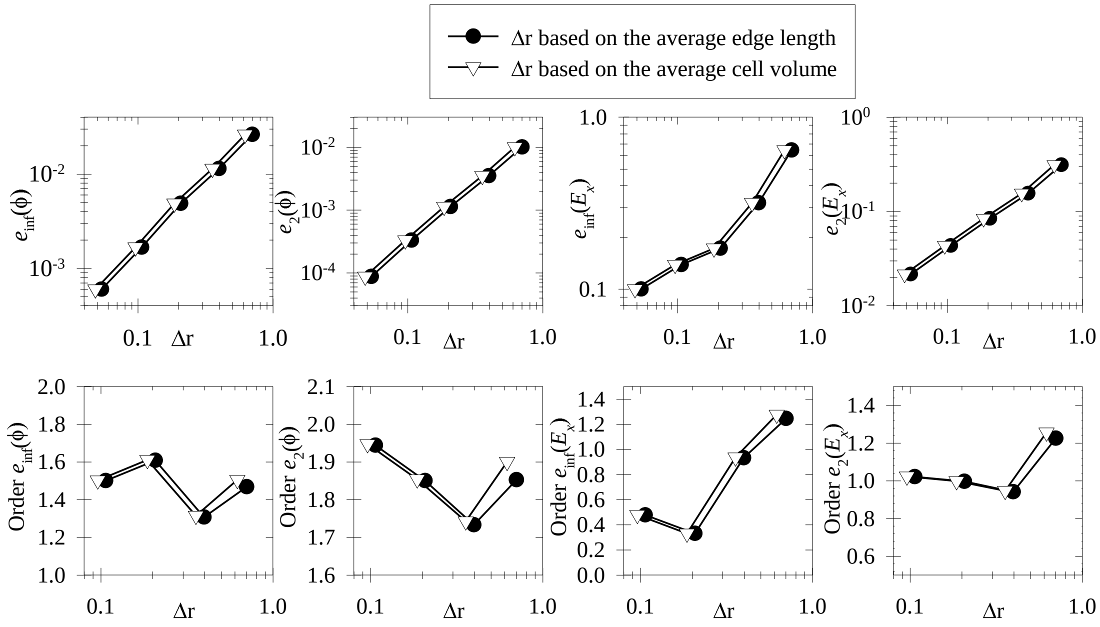


Figure 8. Effects of edge-averaged and volume-averaged cell size on L^2 and L^∞ norms of the relative error in potential and x-component of electric field (Top) and corresponding orders of convergence (Bottom).

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 filed 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 $\phi_0 = en_0R^2 / 12\epsilon_0$ 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}, $\alpha = 2\times10^{-2}; 4\times10^{-2}; 8\times10^{-2}$ and with 300 computational particles per cell.

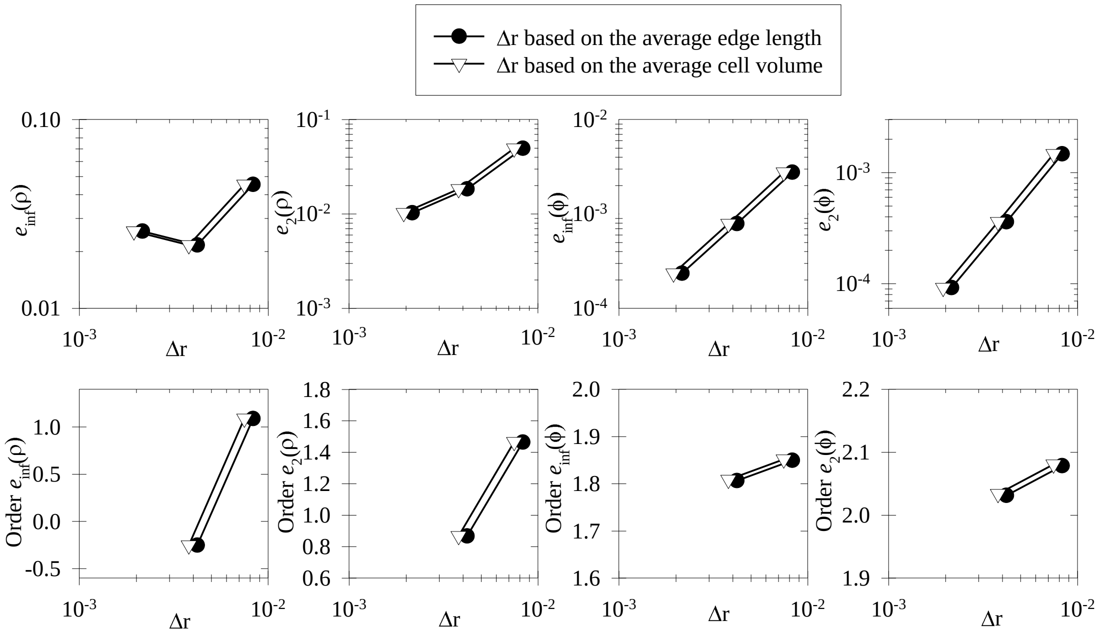


Figure 9. Effects of edge-averaged and volume-averaged cell size on L^2 and L^∞ norms of the relative error in charge density and potential (Top) and corresponding orders of convergence (Bottom).

Figure 9 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 $\omega_e = \sqrt{(e^2 n_0)/(m_e \epsilon_0)}$. The simulations were performed with R=0.1 m, $n_0 = 10^{14}$ m^{-3}, α=3×10^{-2} and variable timesteps $\Delta t = 5\times10^{-11}; 5\times10^{-10}; 5\times10^{-9} s$ 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 Figure 10(a) and x-component of momentum in Figure 10(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.

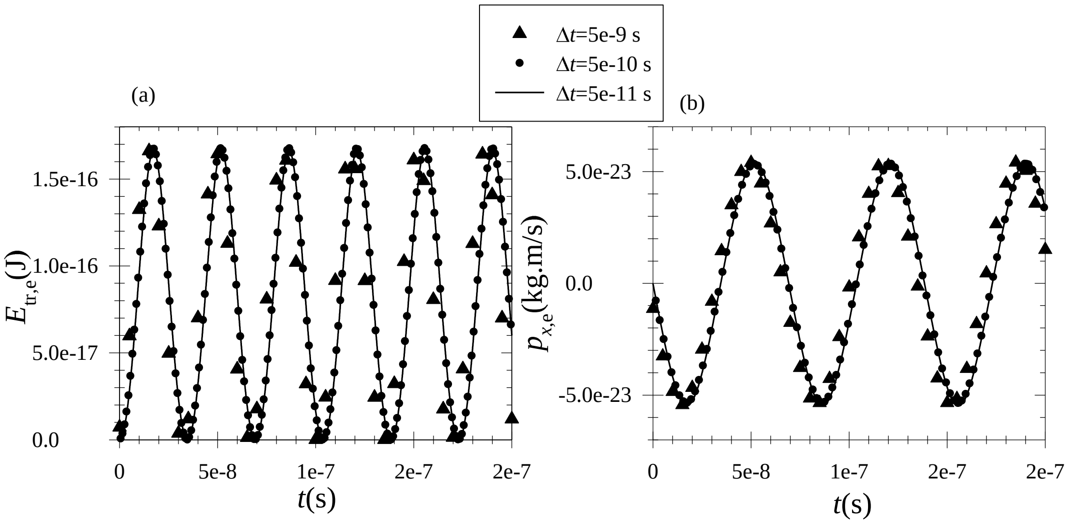


Figure 10. Effects of time step on translational energy (a) and x-component of momentum (b) of an electron in a potential well.

The fourth test case provides analysis of heating, slowing down and deflection times of EUPICC following Hockney and Eastwood (1988) and Gatsonis and Spirkin (2009). 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 $k_B T_s/2$,

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

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

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

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 (Hockney, 1971) and is defined as the time for the root-mean-square average deflection angle to reach 90 degrees,

$$\sqrt{\langle \phi_s^2\ \tau_d \rangle} = \sqrt{\frac{1}{\mathrm{N}_s} \sum_{p=1}^{N_s} F_{Wp} \varphi_p^2(\tau_d)} = \pi/2\,, \tag{58}$$

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

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

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}, $T_e = T_i = 2\,\mathrm{eV}$ and zero drift velocity. Previous investigations (Hockney and Eastwood,1988; Gatsonis and Spirkin, 2009) 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.

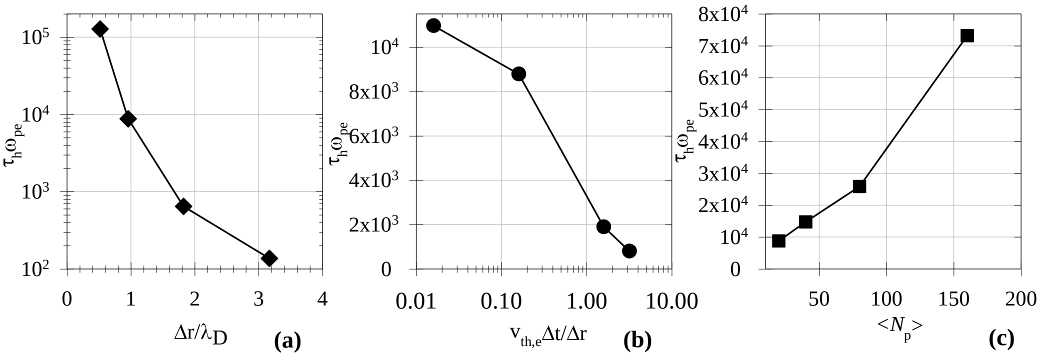


Figure 11. Heating time in EUPICC simulations showing the effects of grid size (a), time step (b) and the average number of computational particles per cell (c).

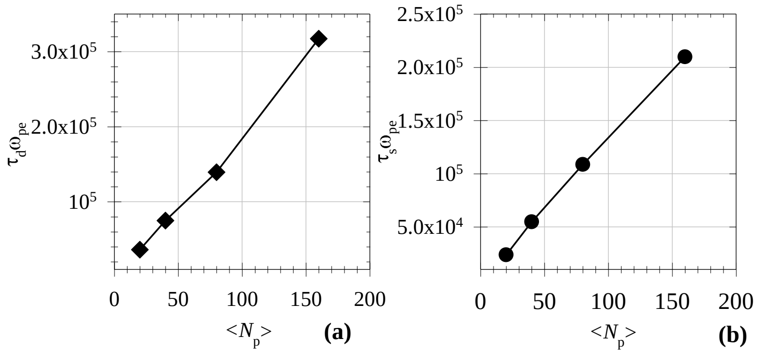


Figure 12. Deflection (a) and slowing down (b) times in EUPICC simulations as a function of the average number of computational particles per cell with $\Delta r = \lambda_{De}$ and $\Delta t = 0.1\omega_{pe}^{-1}$.

Figure 11a 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 Figure 11b. The general trend is that increasing time step leads to decreasing the heating time. At time steps below $\Delta t = 0.1595(\Delta r / v_{\text{th},e}) = 0.1\omega_{pe}^{-1}$ the further decrease in time step by an order of magnitude from $\Delta t = 0.1595(\Delta r / v_{\text{th},e}) = 0.1\omega_{pe}^{-1}$ to $\Delta t = 0.01595(\Delta r / v_{\text{th},e}) = 0.01\omega_{pe}^{-1}$ only slightly increases the heating time from $8.8\times10^3\omega_{pe}^{-1}$ to $1.1\times10^4\omega_{pe}^{-1}$. Similar behaviour was found in the earlier simulations of unstructured PIC on Delaunay-

Voronoi meshes (Gatsonis & Spirkin, 2009). 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 Figure 11c. 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 avergae 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 Figure 11 are used as a guide in the determining of the EUPICC simulation parameters in the subsequent simulations. Figure 12 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 $\tau_d = 3.6\times10^4\omega_{pe}^{-1}$ and $\tau_S = 2.4\times10^4\omega_{pe}^{-1}$ respectively at $\langle N_p \rangle = 20$ to $\tau_d = 3.2\times10^5\omega_{pe}^{-1}$ and $\tau_S = 2.1\times10^5\omega_{pe}^{-1}$ 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.

Table 2. Input conditions for EUPICC simulations of cylindrical Langmuir probes.

Case	Thin sheath	OML
R_D (m)	1.2x10^{-2}	5x10^{-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

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 2 and cover the regime of operation from thin-sheath ($r_p \gg \lambda_D$) to orbital-motion limited (OML) ($r_p \ll \lambda_D$). Figure 13a 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 12 r_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 2. Computational particles reaching the probe and the outer surface are removed from the 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. Figure 13b 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 EUPICC simulations compare favorably to the experimentally validated numerical predictions by Laframboise (1966).

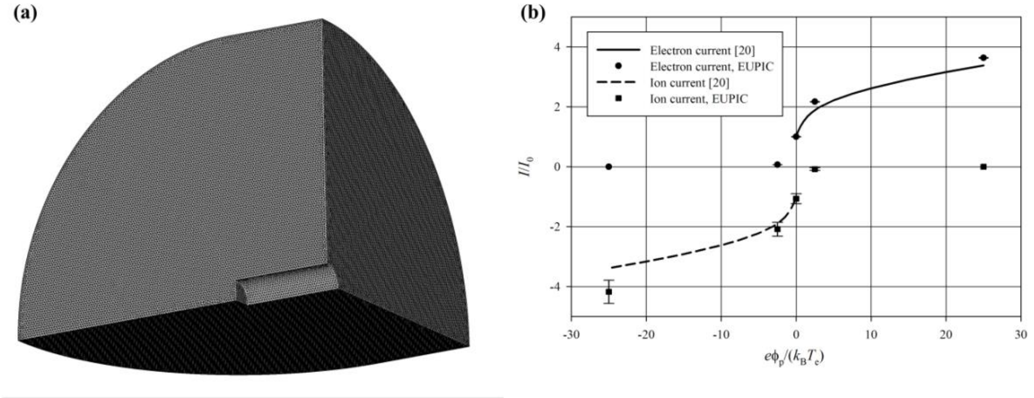


Figure 13. EUPICC simulation of current collection by a cylindrical probe in thin sheath regime ($r_p \gg \lambda_D$) (a) Computational domain; (b) the normalized collected electron and ion currents.

Figure 14a show the cut of the computational domain for the EUPICC 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 2 corresponding to the OML case. Figure 14a shows the comparison of the electron current normalized by the random electron current calculated by EUPICC method with the Laframboise simulations (1966) as a function of the probe potential normalized by the electron temperature. The results of these simulations compare well.

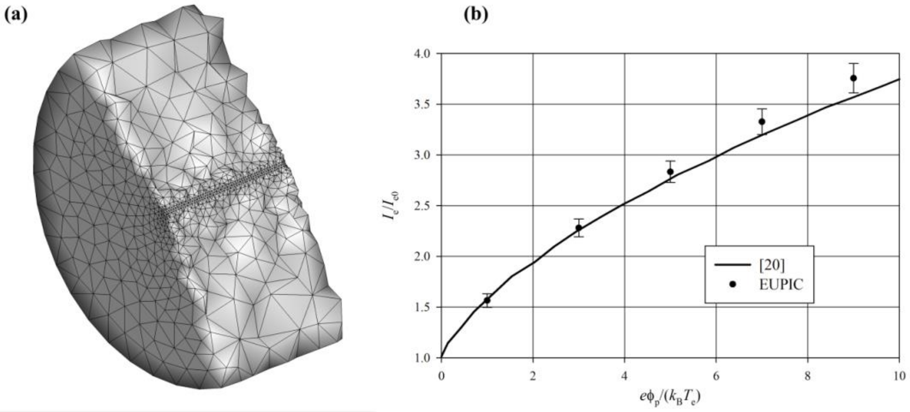


Figure 14. EUPICC simulation of current collection by a cylindrical probe in the OML regime $r_p \ll \lambda_D$ (a) Computational domain; (b) the normalized collected electron current as a function of normalized probe potential.

1.2 Mathematical and Computational Aspects Multi-Weight Collison Method in EUPICC

1.2.1 *State-of-the art in Multi-Weight Collision Methods*

For the simulation of an electrostatic partially ionized plasma the EUPICC method must be supplemented with collision algorithms. In this case the system of equations solved becomes the Boltzmann-Poisson system and the rhs of eq. (1) includes a collision operator. In the absence of electric field s EUPICC reduces to an unstructured DSMC code governed by the Boltzmann equation (Gatsonis et al, 2013). As discussed in Section 1.1 for collisionless plasma simulations using EUPICC, all particle species can have their own weights (Averkin and Gatsonis, 2018). For simulations of collisional, partially-ionized plasmas with large differences in species number densities, efficient multi-weight collision algorithms are needed. We present the new multi-weight real-time-counter method for neutral collisions which can be extended to charged particle collisions.

Bird proposed originally to use multiple species weights in neutral-neutral collisions. In this scheme during a collisions between two particles with different weights W_1 and W_2 ($W_1 > W_2$) velocity components of a particle with smaller weights, W_2, are always replaced by postcollision velocities as if a collision of two particles of the same weight, W_2, would occur. For the particle with the larger weight, W_1, postcollision velocities are assigned with a probability W_2/W_1. This scheme doesn't conserve energy and momentum in each collision but rather conserves them in average. Similar scheme was also proposed in Serikov et al, (1999). Boyd (1996) introduced a collision scheme that strictly conserves a linear momentum during each collision with energy conservation per all collisions. In this scheme during a collision of two particles with different weights W_1 and W_2 ($W_1 > W_2$) and velocities $\mathbf{v}_1$ and $\mathbf{v}_2$ respectively, a particle with larger weight, W_1, is split into two particles with the weights W_2 and $W_1 - W_2$. Then the collision between particles with the same weight W_2 is modeled and postcollision velocities $\mathbf{v}_1'$ and $\mathbf{v}_2'$ are assigned to these particle. Then two parts of the split particle with weights W_2 and $W_1 - W_2$ and velocities $\mathbf{v}_1'$ and $\mathbf{v}_1$ respectively are merged. The velocity assigned to the merged particle is $\mathbf{v}_1'' = (1 - W_2/W_1)\mathbf{v}_1 + (W_2/W_1)\mathbf{v}_1'$ to guarantee linear momentum conservation. However, the total energy of these two particle is not conserved. To force conservation the excess of energy is added to the center of mass energy of the next collision of particles with the largest weights. Thus, the scheme conserved energy for all collisions during a time step. Wu et al. (2003) extended this method to include volume and surface chemical reactions. In this extended version chemically reacting particles are split into reactive and non-reactive parts. The reactive parts have the same weight and the corresponding chemical reaction follows standard total collision energy approach of Bird (1994). The non-reactive parts are elastically collided following (Boyd, 1996). Rjasanow and Wagner (1996) proposed a more rigorous approach of dealing with collisions of particles with multiple weights. They introduced a stochastic weighted particle method (SWPM), a variant of the DSMC method, with all particle having their own weights. The collisions are simulated using random weight transfer approach, i.e. during each collision particles are split and only parts of them with the same weight actually collide and get post-collision velocities similar to traditional DSMC. In contrast to Boyd's scheme no merging of particles happen after collision. It leads to increasing number of particles during each collision. To control the number of particles a separate particle merge step is used. During this step particles are divided into groups and each group is replaced by a pair of particles by requiring conservation of mass, energy and momentum. It should be noted that Boyd's collision algorithm is a particular version of the more general SWPM where the weight transfer function equals to the smallest weight of two colliding particles and merge step happens right after collision

We present in this report a new multi-weight elastic collisions approach that conserves both energy and linear momentum for each collision. This algorithm is derived for a general case where each particle can have its own weight and is based on the real collision counter (RCC) method for collisions sampling and spread in velocity distribution (VSD) function approach. The RCC method is implemented in EUPICC and is verified by considering an equilibration of two Maxwellian distribution functions with different temperatures represented by species with different weights.

1.2.2 *Real Collision Counter (RCC) Approach with Velocity Spread (VS)*

The real collision counter (RCC) formulation is an extension of the no-time counter (NTC) algorithm of Bird (1994). In the original NTC method a maximum number of tentative collisions between computational particles of species p and q, $N_{pq\ \max}$, with the same particle weight W in a cell with cell volume V_c during time step Δt is given as

$$N_{pq\ \max} = \frac{N_p W\ N_q - \delta_{pq}\ \ \sigma_{T,pq} G_{r,pq\ \ \max}}{V_c(1+\delta_{pq})}\Delta t\,, \tag{60}$$

where $\sigma_{T,pq}$ is the total collision cross-section of collisions between species p and q, $G_{r,pq}$ is the relative speed between particles of species p and q, δ_{pq} is the Kronecker delta. Expression (60) takes into account the modification of the number of collisions for the same species as in Bird (2007).

Instead of using number of collisions between computational particles, our algorithm considers a maximum number of collisions between real particles, $\mathrm{N}_{pq\ \ \max}^{\mathrm{RCC}}$, defined as

$$\mathrm{N}_{pq\ \ \max}^{\mathrm{RCC}} = \frac{\sum_{i=1}^{N_p}\sum_{j=1}^{N_q} W_i W_j - \delta_{pq}\sum_{i=1}^{N_p} W_i^2}{V_c(1+\delta_{pq})}\ \ \sigma_{T,pq} G_{r,pq\ \ \max}\ \Delta t \tag{61}$$

If we set all weights equal to 1 then Eq. (61) represents maximum number of real collisions of real particles in the volume V_c in time Δt. For the same-weight collisions Eq. (61) reduces to the standard NTC result given by Eq. (60).

We propose the following algorithm for multi-weight collisions:

1. For each pair of species p and q calculate the maximum number of tentative real collisions $\mathrm{N}_{pq\ \ \max}^{\mathrm{RCC}}$ and nullify real collisions counter $\mathrm{N}_{pq}^{\mathrm{RCC}}$.
2. Choose a pair of particles from species p and q with probabilities $\dfrac{W_i}{\sum_{i=1}^{N_p} W_i}$ and $\dfrac{W_j}{\sum_{j=1}^{N_q} W_j}$. Increase real collisions counter as $\mathrm{N}_{pq}^{\mathrm{RCC}} = \mathrm{N}_{pq}^{\mathrm{RCC}} + \min(W_i, W_j)$.
3. Calculate effective real speed between collision particles $G_{\mathrm{eff},pq}$ and accept collision with the probability $\sigma_{T,pq}(G_{\mathrm{eff},pq})G_{\mathrm{eff},pq}\Big/\ \sigma_{T,pq} G_{r,pq\ \ \max}$.

4. Assign post-collision parameters according to multi-weight collision algorithm described in the next section.
5. Continue to step 2 and repeat calculations while $\mathrm{N}_{pq}^{\mathrm{RCC}} < {\mathrm{N}_{pq}^{\mathrm{RCC}}}_{\max}$.

As can be seen from the description of the presented multi-weight collision algorithm it shares some similarities between the standard DSMC and SPWM methods. For the same weight collisions we reduce to the standard NTC scheme as long as our assignment of post-collision parameters is consistent with the standard DSMC method. In the next section we present our split-merge algorithm that conserves energy and linear momentum and reduces to the standard NTC algorithm for the same weight collisions.

1.2.3 *Energy and Linear and Angular momentum conserving Split-merge Algorithm*

The assignment of post-collision parameters is the key algorithmic step in the new multi-weight collision scheme proposed in the present article. Consider a pair of particles of species p and q. Let W, M and $\mathbf{v}$ be the weight, mass and velocity of the particle from species p and w, m and $\mathbf{u}$ be the weight, mass and velocity of the particle of species q. In the following discussion we assume that $W \geq w$. We also introduce two ratios $\mu = m / M$ $(0 < \mu < +\infty)$ and $\omega = w / W$ $(0 < \omega \leq 1$) that will be useful in the following discussion. In addition, we assign the orientation of collision plane by unit vector $\mathbf{n}$. It is known from the simple collision theory (Vincenti and Kruger, 1967) that conservation of angular momentum requires that the collision plane should be normal to the total angular momentum vector in the relative reference frame. Thus, the unit vector $\mathbf{n}$ is collinear with the total angular momentum vector. Conservation of total linear and angular momentum for the collision pair of two particles with different weights can be written as

$$\begin{aligned} &WM\mathbf{v} + wm\mathbf{u} = WM\tilde{\mathbf{v}} + wm\tilde{\mathbf{u}}, \\ &WMv^2 / 2 + wmu^2 = WM\tilde{v}^2 + wm\tilde{u}^2, \end{aligned} \tag{62}$$

where tilded are the post-collision velocities in the multi-weight collision and $v = \left|\mathbf{v}\right|$, $u = \left|\mathbf{u}\right|$.

On the other hand, to properly describe transport properties in the traditional DSMC algorithm we require the following conservation laws

$$\begin{aligned} &M\mathbf{v} + m\mathbf{u} = M\mathbf{v}' + m\mathbf{u}', \\ &Mv^2 / 2 + mu^2 = Mv'^2 / 2 + mu'^2 / 2, \end{aligned} \tag{63}$$

where prime variables correspond to post-collision parameters.

Without additional modifications Eqs. (62) and Eqs. (63) cannot be satisfied simultaneously. In Boyd (1996) the particle splitting is introduced. It allows to satisfy momentum conservation in both equations (62) and (63) and energy conservation in Eq. (63) by splitting the particle with the larger weight. However, it introduces the excess energy that was redistributed among particles of the next collision pair of the same species. In our current algorithm we go further in splitting of colliding particles and introduce additional parameters for storing this excess of energy for each colliding particle. This energy can be attributed to a spread in the velocity distribution function in the velocity space for each computational particle. This interpretation resembles the ideas used in the construction of the distributional DSMC method (Shrock and Wood, 2012). In this case we assign to the particle with weight W the additional energy WE and to the particle with the weight w the energy we. Thus, we have a modified energy and momentum conservation equation that can be written as

$$
\begin{aligned}
&WM\mathbf{v} + wm\mathbf{u} = WM\tilde{\mathbf{v}} + wm\tilde{\mathbf{u}}, \\
&W\left(M\frac{v^2}{2} + E\right) + w\left(m\frac{u^2}{2} + e\right) = W\left(M\frac{\tilde{v}^2}{2} + \tilde{E}\right) + w\left(m\frac{\tilde{u}^2}{2} + \tilde{e}\right).
\end{aligned}
\tag{64}
$$

The introduction of additional energy also requires modifications in sampling procedures. The sampling of translational energy of species p with mass m_p in a cell with N_p computational particles with weights W_i and velocities $\mathbf{v}_i$ and spreads E_i is performed as follows

$$
T_{\mathrm{tr},p} = \frac{m_p}{3k_B}\left(\sum_{i=1}^{N_p} W_i\right)^{-1}\left[\sum_{i=1}^{N_p} W_i(v_i^2 + E_i) - \left(\sum_{i=1}^{N_p} W_i\right)^{-1}\left(\sum_{i=1}^{N_p} W_i\mathbf{v}_i\right)^2\right].
\tag{65}
$$

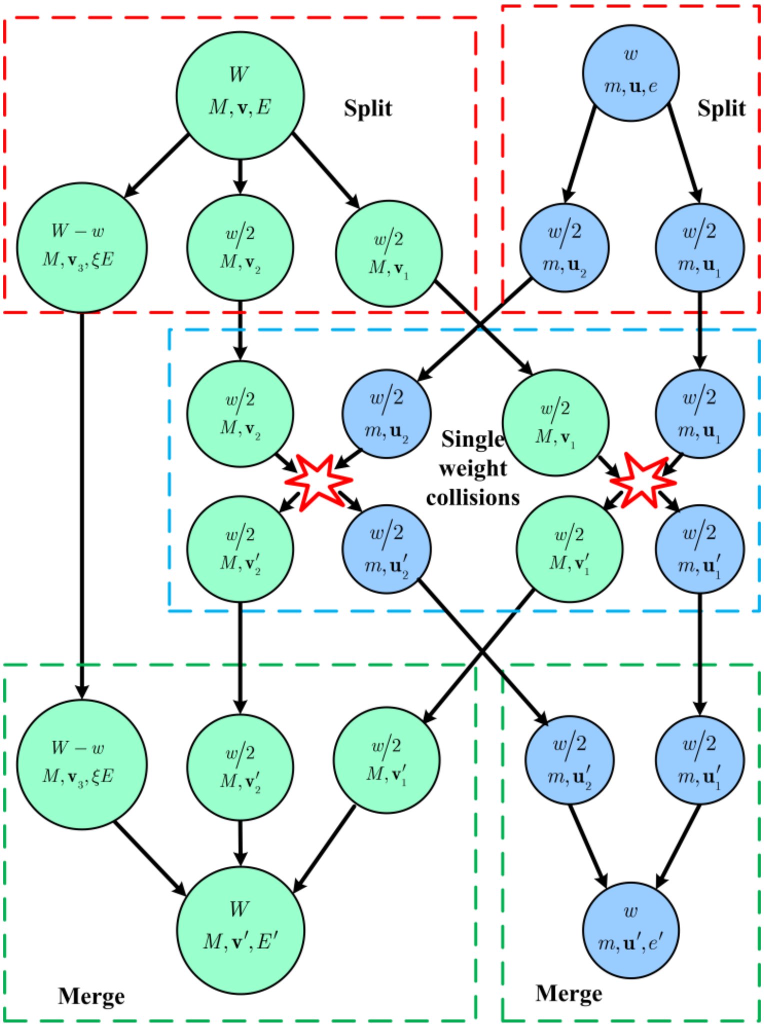


Figure 15. Momentum and energy conserving split-merge algorithm (MWC-II).

Two variations of the splitting scheme were considered using the RCC method for collision selection. The first splitting scheme follows Boyd (1996) where splitting occurs only for the larger-weight particle, then performing single-weight collisions and assigning the excess of the energy to the particle with larger weight. We refer to this algorithm as MWC-I (Averkin et al, 2017). The second splitting scheme is summarized in Figure 15 and is referred to as MWC-II (Averkin et al, 2018). In this algorithm we split the particle with the larger weight W into three particles that have weights $w/2$, $w/2$ and $W-w$ and velocities $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$ respectively. The particles with weights $w/2$ have zero spreads and the particle with weight $W - w$ has the spread ξE. The smaller weight particle is split into two particles with zero spreads and weights $w/2$ and $w/2$ and velocities $\mathbf{u}_1$ and $\mathbf{u}_2$ respectively. The actual collisions take place between particles with weights $w/2$ and velocities $\mathbf{v}_1$ and $\mathbf{u}_1$, $\mathbf{v}_2$ and $\mathbf{u}_2$. These are the same weight collisions that obey standard rules. We denote the post-collision velocities of the same-weight collisions with primes as in the standard DSMC method. After collisions we merge corresponding particles back and calculate their post-collision parameters (velocities and spreads) based on the conservation of energy and momentum. Writing down conservation of energy and momentum for all particles at each split-merge step and during collisions the following system of equations is obtained

$$
\begin{aligned}
&\tilde{\mathbf{v}} = \omega\left(\frac{\mathbf{v} + \mu\mathbf{u}}{1+\mu} + \frac{1}{2}\frac{\mu}{1+\mu}\; 2\mathbf{f} + \mathbf{G}'_1 + \mathbf{G}'_2\;\right) + (1-\omega)\mathbf{v},\\
&\frac{8\tilde{E}}{M} = \omega\left(\frac{\mathbf{g} + \mu\mathbf{h}}{1+\mu} + \frac{\mu}{1+\mu}\; \mathbf{G}'_1 - \mathbf{G}'_2\;\right)^2 + (1-\omega)\xi\frac{8E}{M}\\
&+4\omega(1-\omega)\left(\frac{1}{2}\frac{\mu}{1+\mu}\; 2\mathbf{G} - \mathbf{G}'_1 - \mathbf{G}'_2\; + \frac{1+\omega\mu}{(1-\omega)(1+\mu)}\mathbf{f}\right)^2,\\
&\tilde{\mathbf{u}} = \frac{\mathbf{v} + \mu\mathbf{u}}{1+\mu} - \frac{1}{2}\frac{1}{1+\mu}\; 2\mathbf{f} + \mathbf{G}'_1 + \mathbf{G}'_2\;,\\
&\frac{8\tilde{e}}{m} = \left(\frac{\mathbf{g} + \mu\mathbf{h}}{1+\mu} + \frac{1}{1+\mu}\left(\mathbf{G}'_2 - \mathbf{G}'_1\right)\right)^2.
\end{aligned}
\tag{66}
$$

The vectors $\mathbf{g}$ and $\mathbf{f}$ in the above equations control the splitting of the particle with weight W while a vector $\mathbf{h}$ is responsible for splitting of the second particle with the weight w as can be seen from the following expressions that are based on the conservation of energy and momentum for each split

$$
\begin{aligned}
&\mathbf{v}_1 = \mathbf{v} + \mathbf{g}/2 - \mathbf{f},\\
&\mathbf{v}_2 = \mathbf{v} - \mathbf{g}/2 - \mathbf{f},\\
&\mathbf{v}_3 = \mathbf{v} + \frac{\omega}{1-\omega}\mathbf{f},\\
&\mathbf{u}_1 = \mathbf{u} + \mathbf{h}/2,\\
&\mathbf{u}_2 = \mathbf{u} - \mathbf{h}/2.
\end{aligned}
\tag{67}
$$

$\mathbf{G}_1 = \mathbf{v}_1 - \mathbf{u}_1$ and $\mathbf{G}_2 = \mathbf{v}_2 - \mathbf{u}_2$ are relative velocity vectors before collision for single weight collision pairs. Introducing the relative velocity vector of the original particles pair $\mathbf{G} = \mathbf{v} - \mathbf{u}$ we can write the following expressions for $\mathbf{G}_1$ and $\mathbf{G}_2$

$$
\begin{aligned}
&\mathbf{G}_1 = \mathbf{G} + \frac{1}{2}\; \mathbf{g} - \mathbf{h}\; - \mathbf{f},\\
&\mathbf{G}_2 = \mathbf{G} - \frac{1}{2}\; \mathbf{g} - \mathbf{h}\; - \mathbf{f}.
\end{aligned}
\tag{68}
$$

$\mathbf{G}'_1$ and $\mathbf{G}'_2$ in Eqs. (66) are post-collision relative velocities of each collision pair and are given following Vincenti and Kruger (1967) as

$$\begin{aligned}\mathbf{G}_1' &= \mathbf{G}_1 \cos\chi_1 + \sin\chi_1[\mathbf{G}_1 \times \mathbf{n}_1],\\ \mathbf{G}_2' &= \mathbf{G}_2 \cos\chi_2 + \sin\chi_2[\mathbf{G}_2 \times \mathbf{n}_2],\end{aligned} \tag{69}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are normals to collision planes for each collision pair and χ_1 and χ_1 are the angle between relative velocity vectors of each pair. The unit normal vector $\mathbf{n}_i$ ($i = 1, 2$) from Eqs. (69) is usually defined in terms of angle ϵ_i as

$$\mathbf{n}_i = \frac{\mathbf{G}_i \times \mathbf{i}}{\left|\mathbf{G}_i \times \mathbf{i}\right|} \cos\epsilon_i + \left[\frac{\mathbf{G}_i \times \mathbf{i}}{\left|\mathbf{G}_i \times \mathbf{i}\right|} \times \frac{\mathbf{G}_i}{G_i}\right] \sin\epsilon_i, \tag{70}$$

where ϵ_i is an angle responsible for collision plane orientation.

Based on the energy conservation we have the following relationships between vectors $\mathbf{g}$, $\mathbf{f}$, $\mathbf{h}$ and pre-collision spreads E and e

$$\begin{aligned}&(1 - \xi + \omega\xi)\frac{8E}{\omega M} = g^2 + \frac{4}{1-\omega} f^2,\\ &8\frac{e}{m} = h^2.\end{aligned} \tag{71}$$

If we choose the same relative speed of each collision pair $G_1 = G_2$ then we get the same collision cross-sections for these collisions and the same angles between relative velocity vectors $\chi_1 = \chi_2 = \chi$. It also implies that two particle pairs collide or not collide simultaneously. The condition $G_1 = G_2$ can be rewritten as

$$(\mathbf{G} - \mathbf{f}) \cdot (\mathbf{g} - \mathbf{h}) = 0. \tag{72}$$

Thus, we have 10 unknown parameters ξ , $\mathbf{f}$, $\mathbf{g}$ and $\mathbf{h}$ and only 3 equations for their determination given by Eqs. (71) and (72). We choose the direction of $\mathbf{f}$ to be parallel to the relative velocity vector $\mathbf{G}$ while $\mathbf{g}$ and $\mathbf{h}$ are chosen to be parallel to each other and normal to $\mathbf{f}$ as follows

$$\begin{aligned}\mathbf{f} &= f\frac{\mathbf{G}}{G},\\ \mathbf{g} &= g\frac{\mathbf{G}}{G} \times \frac{\mathbf{u} \times \mathbf{v}}{\left|\mathbf{u} \times \mathbf{v}\right|},\\ \mathbf{h} &= -\sqrt{\frac{2e}{m}}\frac{\mathbf{G}}{G} \times \frac{\mathbf{u} \times \mathbf{v}}{\left|\mathbf{u} \times \mathbf{v}\right|}.\end{aligned} \tag{73}$$

We also introduce angle $\Omega \in \left[0, \pi/2\right]$ that is responsible for splitting of the larger weight particle given by Eq. (71)

$$\begin{aligned}f &= \cos\Omega\sqrt{\left(\frac{1-\xi}{\omega} + \xi\right)(1-\omega)}\sqrt{\frac{2E}{M}},\\ g &= \sin\Omega 2\sqrt{\frac{1-\xi}{\omega} + \xi}\sqrt{\frac{2E}{M}}.\end{aligned} \tag{74}$$

Now we have only two free parameters Ω and ξ. We used numerical multi-parametric investigation in order to determine these parameters at different particle weights ω (Averkin et al, 2018). A summary of the MWC-II algorithm for a pair of particles is:

1. If $\mathbf{v} \nparallel \mathbf{u}$ then we calculate $\mathbf{f}$, $\mathbf{g}$, $\mathbf{h}$ using Eqs. (73). Otherwise we pick a random unit vector $\mathbf{e}_r$ that is normal to $\mathbf{v}$ and calculate $\mathbf{f}$, $\mathbf{g}$, $\mathbf{h}$ as

$$\begin{aligned} \mathbf{f} &= f\frac{\mathbf{G}}{G}, \\ \mathbf{g} &= g\mathbf{e}_r, \\ \mathbf{h} &= -\sqrt{\frac{2e}{m}}\mathbf{e}_r. \end{aligned} \qquad (75)$$

2. Calculate G_{eff} as

$$G_{eff} = G_1 = G_2 = \sqrt{\left(G - \cos\Omega\sqrt{\left(\frac{1-\xi}{\omega} + \xi\right)(1-\omega)\frac{2E}{M}}\right)^2 + \left(\sin\Omega\sqrt{\left(\frac{1-\xi}{\omega} + \xi\right)\frac{2E}{M}} + \sqrt{\frac{2e}{m}}\right)^2}. \qquad (76)$$

3. Accepting a collision with the probability $\dfrac{\sigma_T(G_1)G_1}{\left[\sigma_T(G)G\right]_{\max}}$
4. If a collision takes place then Calculate $\mathbf{G}_1$ and $\mathbf{G}_2$ from Eqs. (68)
5. Calculate $\mathbf{G}_1'$ and $\mathbf{G}_2'$ based on the same values of χ and ϵ from Eqs. (70)
6. Calculate $\tilde{\mathbf{v}}$, $\tilde{E}$, $\tilde{\mathbf{u}}$, and $\tilde{e}$ from Eqs. (66).

1.2.4 *EUPICC Multi-Weight Simulations*

In this section, we present a numerical multi-parametric investigation of the MWC-II algorithm as implemented in the EUPICC code (Gatsonis and Spirkin 2009; Gatsonis et al, 2013; Averkin and Gatsonis, 2018). The goal is to determine the parameters Ω and ξ as a function of particles weight ratio ω that lead to equilibration of two Maxwellian distribution functions with different initial temperatures. We consider two Maxwellian distributions of the same species particles (the same mass and cross-section parameters) at two temperatures 1000 K and 9000 K. We use more than 10000 computational particles to reduce statistical fluctuations. The geometry used to represent a space-homogenous gas is a box with periodic boundary conditions, with the cell size chosen as a fraction of the smallest mean free path. For arbitrary parameters Ω and ξ at different weights the temperatures of both species equilibrate to steady state values that are not necessarily the same as it is expected as can be seen from Figure 16 while the total energy and momentum can be conserved as can be seen from Figure 17.

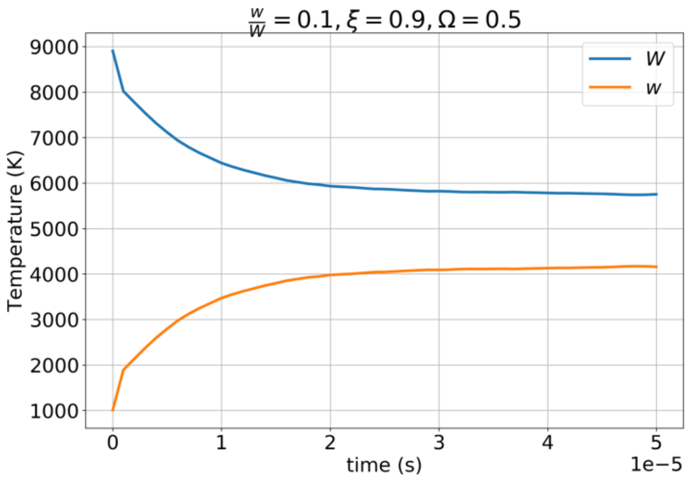


Figure 16. Typical MWC-II simulation result of equilibration problem.

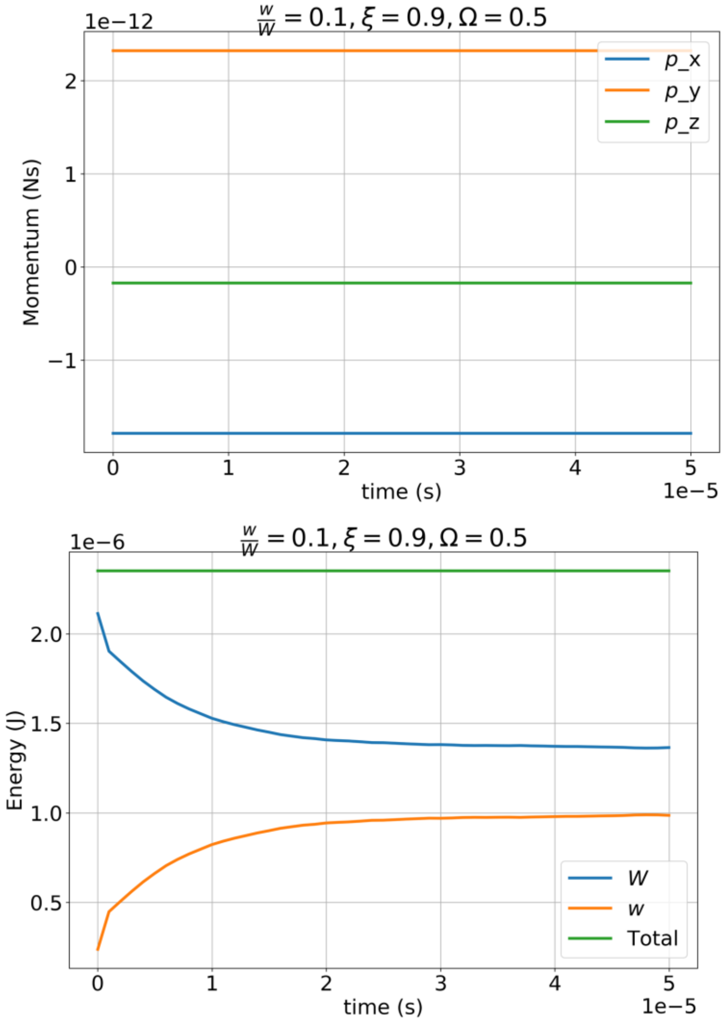


Figure 17. Total linear momentum (left) and energy (right) of a typical MWC-II simulation.

We performed 8,000 individual simulations in order to find the relationship between parameters Ω , ξ and ω . A typical successful equilibration simulation is shown in Figure 18.

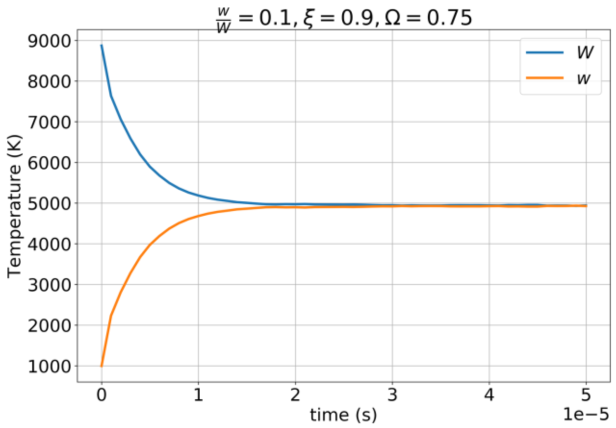


Figure 18. An example of successful equilibration of two Maxwellian distribution functions.

We combined all of the successful results to obtain a 3D plot of parameters that lead to the equilibration as shown in Figure 19.

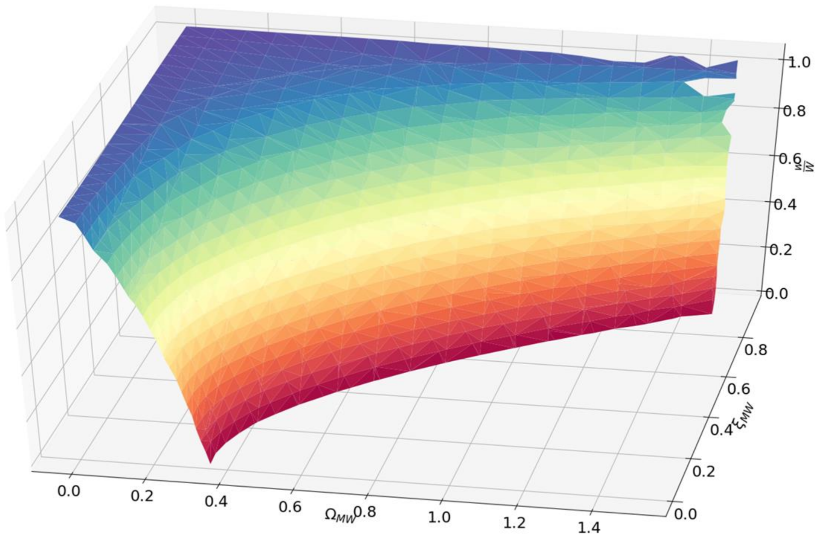


Figure 19. Surface in parametric space showing the relationship between Ω, ξ and ω leading to successful equilibration.

Figure 20 shows a X-Y plot of the parameters leading to the equilibration of two Maxwellian distribution functions.

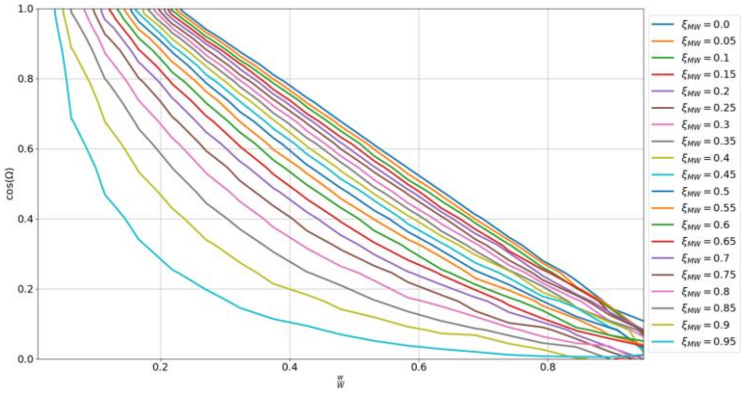


Figure 20. Parameters Ω, ξ and ω that lead to successful equilibration.

1.3 High-fidelity Simulations Enabled by EUPICC

1.3.1 *Plasma Flow over Cubesat*

EUPICC is used next for the simulation of the plasma flow over a CubeSat shown in Figure 21 (Gatsonis, et al., 2016). The CubeSat features an instrument for in-situ measurements in the front unit and an electric propulsion system in the 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 Figure 21. 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 EUPICC computational domain is a rectangular region of $1\mathrm{m}\times 1\mathrm{m}\times 3\mathrm{m}$ shown in Figure 22 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 $n_{\mathrm{O}+}=9.06\times 10^{9}\ \mathrm{m}^{-3}$, $n_{\mathrm{H}+}=2.44\times 10^{9}\ \mathrm{m}^{-3}$, $n_e=1.23\times 10^{10}\ \mathrm{m}^{-3}$, $T_e=2243.1\,\mathrm{K}$, $T_{\mathrm{O}+}=T_{\mathrm{H}+}=1238.5\,\mathrm{K}$ [32]. The particles are injected with velocity $V_x=0, V_y=0, V_z=7455$ m/s from the left as shown in Figure 22. These conditions result in a mesothermal ion flow where the thermal speed of electrons $V_{th,e}>>V_z>V_{th,i}$. The potential boundary conditions are Dirichlet with $\phi=0\,\mathrm{V}$ 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.

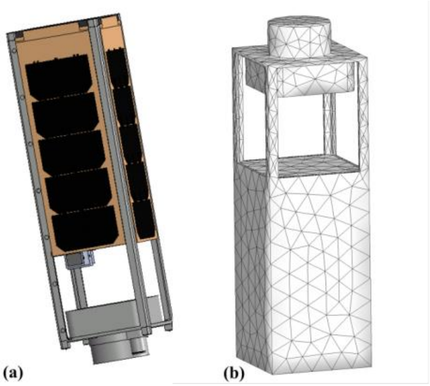


Figure 21. CubeSat used in EUPICC simulation (a) and EUPICC surface grid (b).

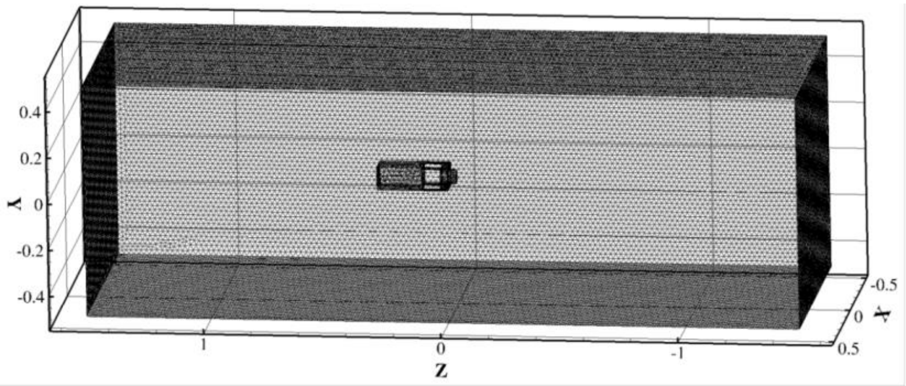


Figure 22. Computational domain used in the EUPICC simulation of the flow over a CubeSat.

Figure 23 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) (Albaran et al, 2016). The total charge density in Figure 24a and Figure 25a is positive in the vicinity of the CubeSat and shows a negative region in its wake. The electron density in Figure 24b and Figure 25b 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. Figure 24c-d and Figure 25c-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 (ALbaran et al, 2016; Miloch et al., 2008).

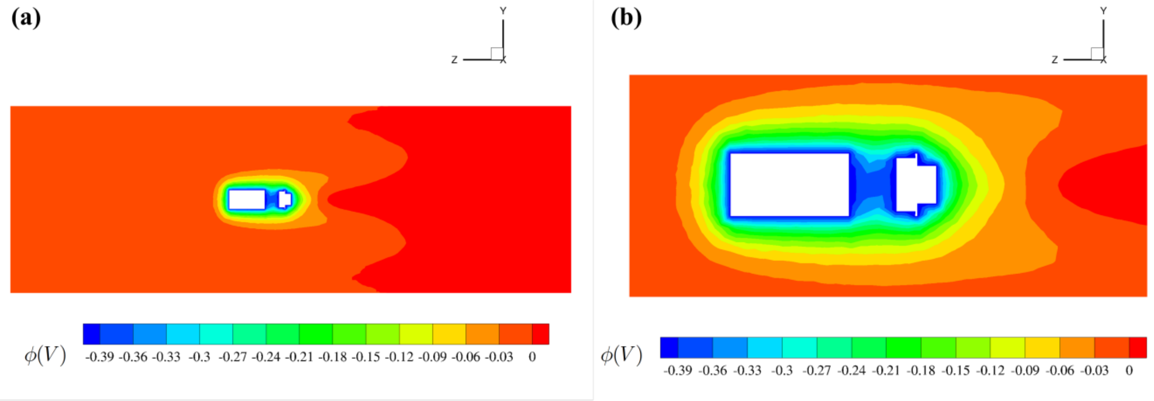


Figure 23. Potential (V) on the (X=0, Y, Z) plane from the EUPICC simulation of the plasma flow around a CubeSat in LEO (a) and near the CubeSat surface (b).

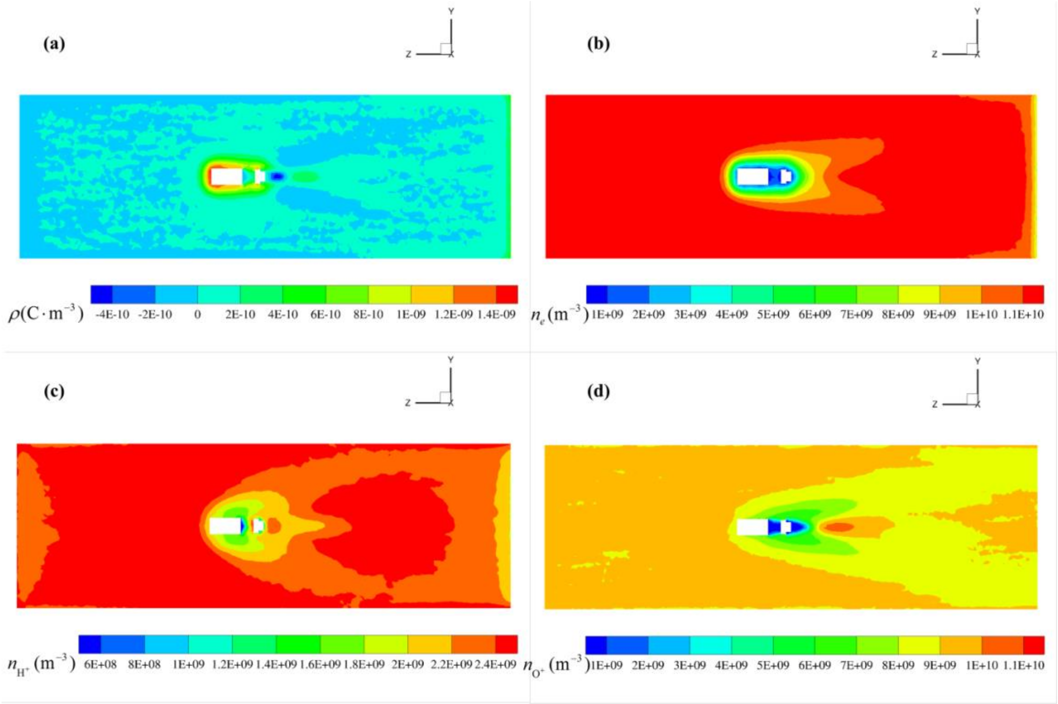


Figure 24. Flowfield properties on the (X=0,Y,Z) plane from EUPICC simulation of the plasma flow around a CubeSat in LEO. Total charge density (a), electron number density (b), H^{+} number density (c), O^{+} number density (d).

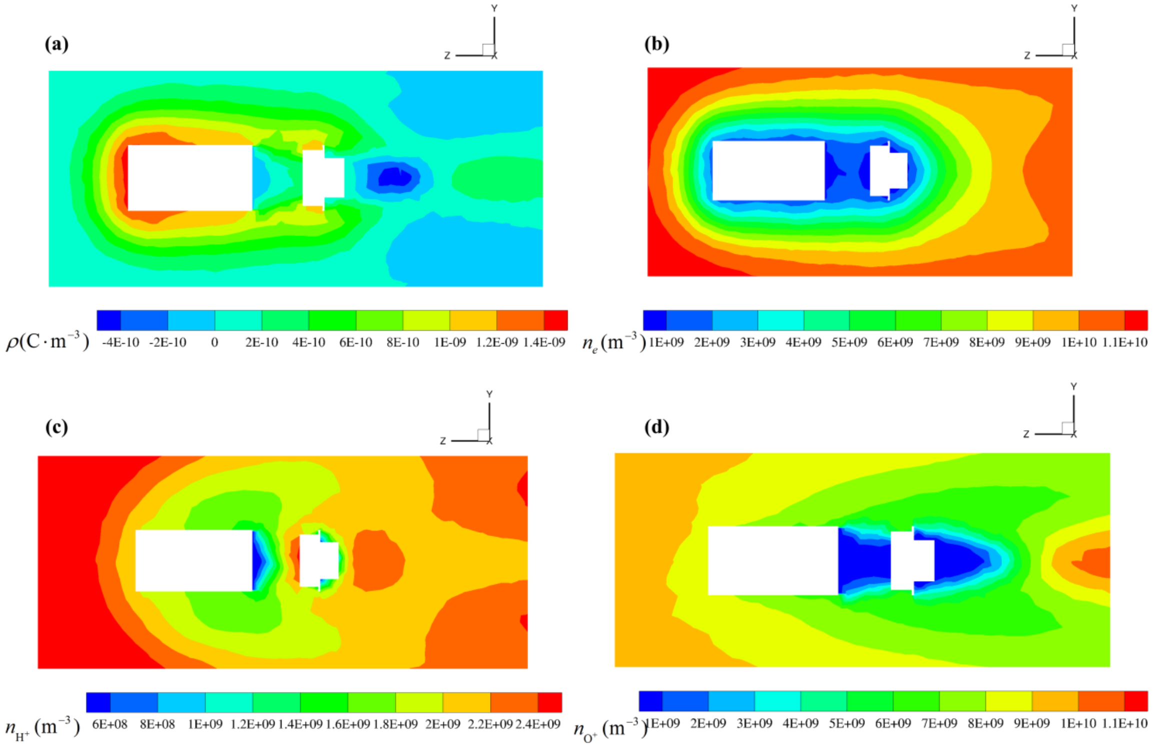


Figure 25. Flowfield properties on the (X=0,Y,Z) plane near the CubeSat surface from EUPICC simulation of the plasma flow around a CubeSat in LEO. Total charge density (a), electron number density (b), H^+ number density (c), O^+ number density (d).

1.3.2 *Ion Beam Neutralization Simulations*

We perform next a series of EUPICC simulations to capture ion beam neutralization phenomena. Ion beam neutralization during operation of electric propulsion devices requires both current and charge density matching of the ion beam using emitted electrons. This current coupling is easily accomplished in practice, yet the exact process has not been adequately described. Proper modeling of current coupling and neutralization will enable development of low-current neutralizers and optimization of neutralizers for micro-propulsion devices and clusters of thrusters. Explanation of the beam coupling mechanism also has bearing on space instrument calibration, electrodynamic tethers, and ionospheric research.

PIC codes have been used widely for beam neutralization simulations and require control of numerical heating that is inherent in PIC codes. To develop a comprehensive background on 1D beam neutralization by co-emitted electrons a series of simulations are performed. Ion beams of Xe with non-dimensional kinetic energy $(KE_i)/(kT_e)\sim 1\times$ 〚10〛^3 $-1\times$ 〚10〛^6 are considered, while the corresponding velocity ratio is $v_{di}/v_{te}\sim 0.1-1.0$, covering the range expected in electric propulsion devices as well as high-energy beams. The simulation domain shown in **Error! Reference source not found.** is a long rectangular tube ($10\times 10\times 2000$ mm) where the ion beam drifts along x-direction. Beam ions are injected at X_{min} boundary and thermal, stationary electrons are accelerated under the self-consistent electrical field generated by the beam front. Reflection particle boundary conditions are set at side walls. Particles reaching the X_{max} boundary are removed from the simulation. The potential at X_{min} boundary is fixed to be zero while zero-Neumann E field boundary conditions are applied to side walls as well as X_{max} boundary. EUPICC simulation results are presented for both fast ions and mesothermal ions

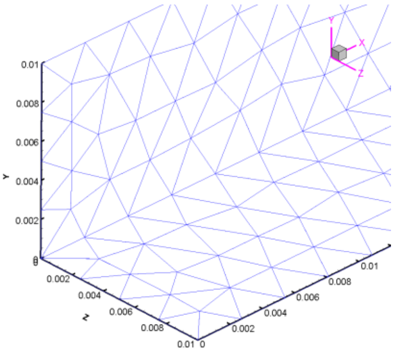
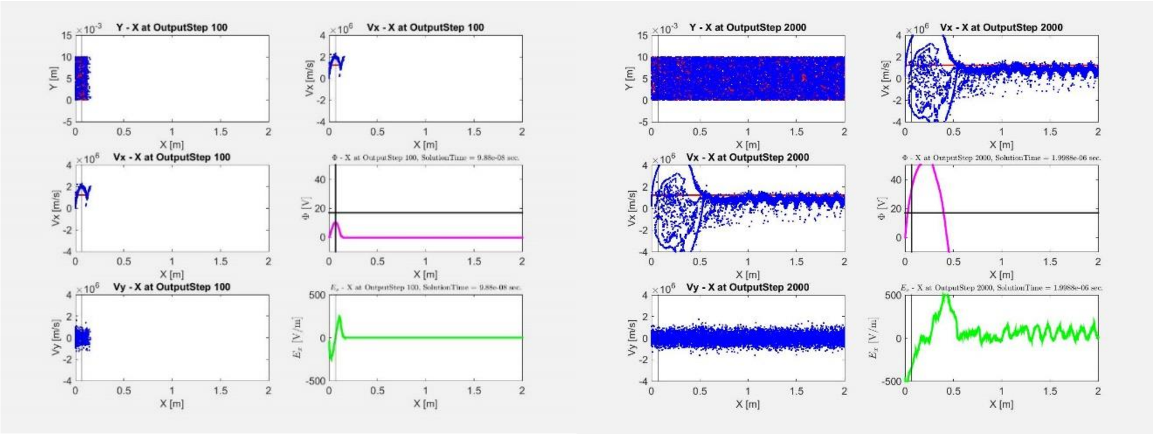


Figure 26. Domain used in EUPICC ion beam simulations.

The ion beam drift velocity considered for “fast ions” case is set to be 1.212×10^6 m/s. The electron temperature is set to be 1 eV. Figure 27 shows the phase space and E field profiles of the fast ion beam neutralization process at different time steps. It is observed that, for a fast ion beam ($v_{di} > v_{te}$), the neutralization process obtained from the PIC simulation matches the idealized 1D neutralization model quite well with regard to the non-dimensional neutralization distance. As the ion beam propagates downstream, the electrons eventually gain a bulk drift velocity same as the ion drift velocity (v_{di}) and thus current coupling is achieved.

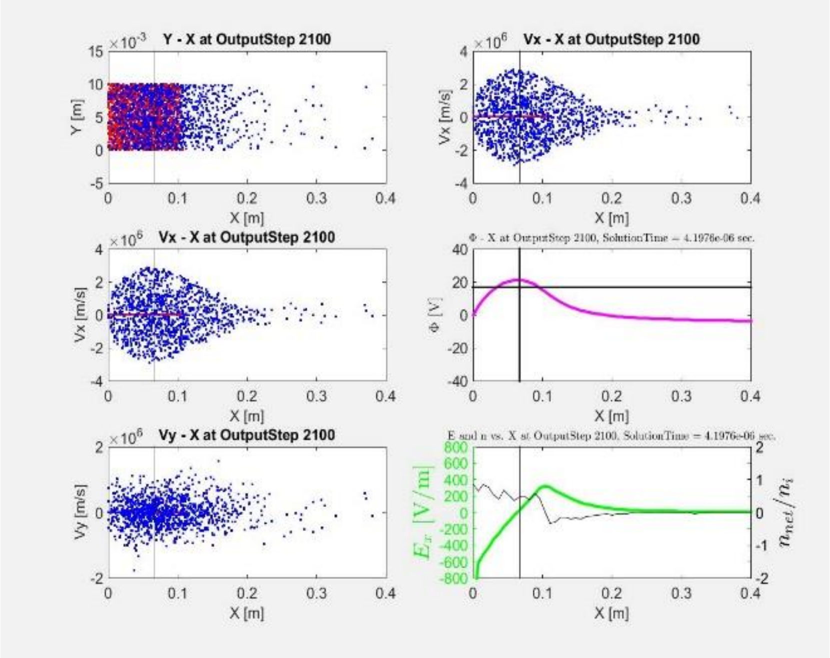


a) Phase space and E field at an early stage of the fast ion beam neutralization. The vertical line in the x-axis indicates the non-dimensional neutralization distance predicted by the idealized 1D model. At this stage, the PIC simulation results match the idealized 1D model well.

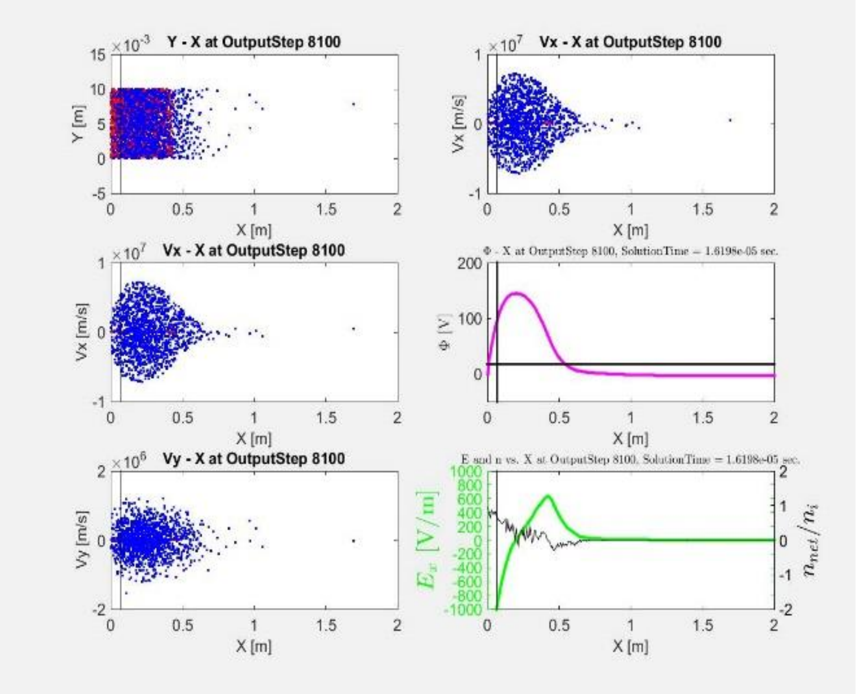
b) Phase space and E field at a later stage of the fast ion beam neutralization. The vertical line in the x-axis indicates the non-dimensional neutralization distance predicted by the idealized 1D model. At this stage, the ion beam has achieved current and density neutralization.

Figure 27. Phase space and electric field of 1D ion beam neutralization for fast ions

The ion beam drift velocity considered for “mesothermal ions” case is set to be 0.024×10^6 m/s. The electron temperature is set to be 1 eV, same as the “fast ions” case. Figure 28 shows the phase space and E field profiles of the mesothermal ion beam neutralization process. The kinetics of electrons are quite different from that of the “fast ions” case in a way that in a mesothermal ion beam, ion drift velocity is smaller than electron thermal velocity ($v_{di} < v_{te}$). The potential well caused by the ion beam front at the early stage does not trap enough electrons to neutralize the ion beam. Therefore, idealized 1D ion beam neutralization models, e.g. Humphries, 2012, are not sufficient to resolve the neutralization process of the mesothermal ion beams emitted by typical electric propulsion devices.



a) Phase space and E field at an early stage of the fast ion beam neutralization. The vertical line in the x-axis indicates the non-dimensional neutralization distance predicted by the idealized 1D model. At this stage, the PIC simulation results match the idealized 1D model well.



b) Phase space and E field at a later stage of the fast ion beam neutralization. The vertical line in the x-axis indicates the non-dimensional neutralization distance predicted by the idealized 1D model. At this stage, the ion beam has achieved current and density neutralization.

Figure 28 Phase space and electric field of 1D ion beam neutralization for mesothermal ions. The kinetics of electrons do not match the idealized 1D model because of the smaller ion drift velocity compared to the electron thermal velocity.

1.4 Conclusions

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

The electric potential in EUPICC 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 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 EUPICC 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 and 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 momentum conservation properties of the EUPICC 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 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 to 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. EUPICC 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 EUPICC 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.

We developed two multi-weight collision algorithms (MWC -I and MWC-II) that use the Real-Counter-Collisional approach for particle selection and spread in velocity space for energy accommodation in order to conserve energy and linear momentum. A number of different split-merge strategies have been also investigated. MWC-I used a splitting of the larger-weight particle into two particles . MWC-II splits two particles into five particles. The main difficulty associated with the splitting in the MWC-II algorithm is the choice of free parameters. In MWC-II there are 10 parameters and only 3 constrains. Five parameters are chosen arbitrarily and a series of multi-parametric simulations is performed in order to determine the remaining parameters in order to achieve equilibration To resolve the lack of angular momentum conservation and microscopic detailed balance in the MWC-I and MWC – II algorithms, an extension is under development (Averkin et al, 2019).

Simulations were also performed to demonstrate the EUPICC to simulate complex geometrical bodies embedded in flowing multi-species plasmas. The test case involves the EUPICC simulation of the plasma flow over a CubeSat under LEO conditions. EUPICC results show the structure of the ion and electron wakes, the formation of ion enhancement regions in the wake, and the potential structure. The EUPICC results are quantitatively and qualitatively consistent with previous 2D and 3D PIC investigations over simpler geometries.

Finally, we used EUPICC in order to resolve the role of electrons in the beam neutralization process. A series of EUPICC simulations of ion beam neutralization is performed for both fast ions ($v_{di} > v_{te}$) and mesothermal ions ($v_{di} < v_{te}$) in an idealized 1D configuration. Results show the current coupling process of both fast and mesothermal ion beams. The numerical simulation successfully resolved the current and density neutralization process of 1D fast ion beams, which is in good agreement with the theoretical 1D beam neutralization model. For mesothermal ions, the numerical simulation does not yield a good agreement with the theoretical 1D model. Further investigations will study effects of background plasmas as well as 2-D and 3-D effects.

2 PERSONNEL SUPPORTED

The following personnel were supported during the period of performance of this grant. The educational status at the time of their support as well as their current position is also indicated.

1. Dr. N. A. Gatsonis, PI, Professor, WPI, Worcester, MA.
2. Dr. Sergey Averkin, Postdoctoral Fellow, WPI, Worcester, MA; 3/15-8/2016. Currently at Tech-X Corporation, Boulder, CO.
3. Dr. Daoru Han, Research Assistant Professor, WPI, Worcester, MA; 9/16-8/17. Currently Assistant Professor at Missouri S&T, Rolla, MO.
4. Shin Tian, Research Assistant, WPI, Worcester, MA; 6/17-6/18.

3 PUBLICATIONS AND PRESENTATIONS

The following journal publications, conference proceedings and abstracts appeared during the period of performance of this grant.

1. Averkin, S. and N.A. Gatsonis, “A parallel electrostatic Particle-in-Cell method on unstructured tetrahedral grids for large-scale bounded collisionless plasma simulations,” Journal of Computational Physics, Volume 363, 15 June 2018, pg 178-199, 2018.

2. Han, D., S. Averkin, S. and N.A Gatsonis, “3D Particle-in-Cell Simulations of Electric Propulsion Ion Beam Neutralization”, IEPC-2017-473, 35th International Electric Propulsion Conference, Atlanta, Georgia, USA, October 8 – 12, 2017.
3. (Invited) Averkin, S. N., N. A. Gatsonis, “Development of a Multi-Weight Collision Algorithm For DSMC/PIC Simulations”, International Conference on Plasma Science, Denver, CO, 26 June, 2018.
4. Han, D., Averkin, S. and N. A. Gatsonis , Current Coupling Via Electron Emission In Fast And Mesothermal Ion Beams, International Conference on Plasma Science, Denver, CO, 26 June, 2018.
5. (Invited) Averkin, S. N., N. A. Gatsonis, Particle-In-Cell Algorithm on Unstructured Tetrahedral Meshes, International Conference on Plasma Science (ICOPS), Atlantic City, MD, 22 May, 2017.
6. Averkin, S. N., Han, D. and N. A. Gatsonis, “New Multi-Weight Collision Algorithm for DSMC/PIC Simulations of Gases And Plasma Flows,” ICOPS 2017, Atlantic City, MD, May 22, 2017.
7. Han, D. and N. A. Gatsonis, “Modeling Ion Beam Neutralization with 3d Particle-In-Cell Simulations,” ICOPS 2017, Atlantic City, MD, May 22, 2017.

4 REFERENCES

1. Albarran, R.M., A. Barjatya, Plasma density analysis of CubeSat wakes in the Earth’s ionosphere, J. Spacecr. Rockets 53(3) (2016) 393–400.
2. Averkin, S.N., D. Han and N. A. Gatsonis, "A Multi-weight Collision Algorithm for DSMC/PIC Simulations," in preparation, 2019.
3. Averkin, S. and N.A. Gatsonis, “A parallel electrostatic Particle-in-Cell method on unstructured tetrahedral grids for large-scale bounded collisionless plasma simulations,” Journal of Computational Physics, Volume 363, 15 June 2018, pg 178-199, 2018.
4. Averkin, S.N., D. Han and N. A. Gatsonis, "Development of a Multi-weight Collision Algorithm for DSMC/PIC Simulations," in *45th IEEE International Conference on Plasma Science (ICOPS)*, Denver, Colorado, 2018.
5. Averkin, S.N., D. Han and N. A. Gatsonis, "New Multi-Weight Collision Algorithm for DSMC/PIC Simulations of Gases and Plasma Flows," in 44th IEEE International Conference on Plasma Science (ICOPS), Atlantic City, New Jersey, 2017.
6. Bettencourt, M.T, Controlling self-force for unstructured Particle-in-Cell (PIC) codes, IEEE Trans. Plasma Sci. 42(5) (2014) 1189–1194.
7. Bird, G., Molecular gas dynamics and the direct simulation monte carlo of gas flows, Oxford: Clarendon Press, 1994.
8. Bird, G.A., "Sophisticated dsmc," in Notes prepared for a short course at the DSMC07 meeting, Santa Fe, USA, 2007.
9. Birdsall C.K. and A. B. Langdon, Plasma Physics via Computer Simulation, CRC Press, 2004.
10. Blandino, J.J., N. Martinez, M.A. Demetriou, N.A. Gatsonis, N. Paschalidis, Feasibility for orbital life extension of a CubeSat flying in the lower thermo-sphere, J. Spacecr. Rockets 53(5) (2016) 864–875.
11. Boris, J.P., Relativistic plasma simulation–optimization of a hybrid code, in: Proceedings of the Fourth Conference on Numerical Simulation of Plasma, Washington D.C., 1970.
12. Boyd, I.D, "Conservative species weighting scheme for the direct simulation Monte Carlo method," Journal of Thermophysics and Heat Transfer, vol. 10, no. 4, pp. 579-585, 1996.
13. Buneman, O., Time-reversible difference procedures, J. Comput. Phys. 1(4) (1967) 517–535.

14. Chanrion O. and Neubert T., A PIC-MCC code for simulation of streamer propagation in air, *J. Comp. Phys* 227, 7222–7245, 2008.
15. Dimarco, G., Caflisch, R., and Pareschi, L., Direct simulation Monte Carlo schemes for Coulomb interactions in plasmas, *Comm. App. Ind. Math*., 1, 1, 72–91, 2010.
16. Droniou, J., Finite volume schemes for diffusion equations: introduction to and review of modern methods, Math. Models Methods Appl. Sci. 24(8) (2014) 1575–1619.
17. Fabris, A.L., 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.
18. Gatsonis, N.A., 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), 57–73, 2016.
19. Gatsonis, N.A., R. E. Chamberlin and S. N. Averkin, "An unstructured direct simulation Monte Carlo methodology with Kinetic-Moment inflow and outflow boundary conditions," Journal of Computational Physics, vol. 233, pp. 148-174, 2013.
20. Gatsonis, N. A., A. Spirkin, A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay–Voronoi grids, J. Comput. Phys. 228(10), 3742–3761, 2009.
21. Gatsonis, N.A. and Yin, X., Hybrid (Particle/Fluid) Modeling of Pulsed Plasma Thruster Plumes, *Journal of Propulsion and Power,* Vol. 17, No. 5, pp. 945-958, Sept-Oct. 2001.
22. Geuzaine, C., 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.
23. Humphries, S. Jr., “Charged Particle Beams”, Dover Publications, Inc., Mineola, New York, 2012, pp. 501-511.
24. Hockney, R. C. and J. W. Eastwood, Computer Simulation Using Particles, New York: Taylor & Francis, 1988.
25. Hockney, R.W., Measurements of collision and heating times in a two-dimensional thermal computer plasma, J. Comput. Phys. 8(1), 19–44, 1971.
26. Ichimaru, G., Basic Principles of Plasma Physics: A Statistical Approach, Benjamin, New York, 1973.
27. International reference ionosphere, NASA, online available: https://iri .gsfc .nasa .gov/(Accessed10 May 2017).
28. Jackson, J.D., Classical Electrodynamics, Wiley, 1999.
29. Kolev S., Hagelaar G.J.M., and Boeuf J.P., Particle-in-cell with Monte Carlo collision modeling of the electron and negative hydrogen ion transport across a localized transverse magnetic field, *Physics of Plasmas* 16, 042318, 2009
30. Laframboise, J.,G., Theory of Spherical and Cylindrical Langmuir Probes in a Collisionless, Maxwellian Plasma at Rest, Univ. Toronto Institute of Aerospace Studies, Toronto, 1966.
31. Meierbachtol, C.S., A.D. Greenwood, J.P. Verboncoeur, B. Shanker, Conformal electromagnetic particle in cell: a review, IEEE Trans. Plasma Sci. 43(11) (2015) 3778–3793.
32. Miloch, W.J., 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.
33. Montgomery, D.C., D.A. Tidman, Plasma Kinetic Theory, McGraw–Hill, New York, 1964.
34. Nanbu, K., Probability Theory of Electron-Molecule, Ion-Molecule, Molecule-Molecule, and Coulomb Collisions for Particle Modeling of Materials Processing Plasmas and Gases, *IEEE Trans. Plasma Sci.*, Vol. 28, No. 3, June, 2000.

35. Nanbu, K., Theory of cumulative small-angle collisions in plasmas, *Phys. Review E,* Volume 55, Issue 4, pp.4642-4652, 1997.
36. Naumov, M., 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 GPUaccelerated algebraic multigrid and preconditioned iterative methods, SIAM J. Sci. Comput. 37(5), S602–S626, 2015.
37. Naumov, M., P. Castonguay, J. Cohen, Parallel Graph Coloring with Applications to the Incomplete-LU Factorization on the GPU, Nvidia Corp., Santa Clara, CA, 2015.
38. Naumov, M., Parallel Solution of Sparse Triangular Linear Systems in the Preconditioned Iterative Methods on the GPU, Nvidia Corp., Santa Clara, CA, 2011.
39. Pavarin, D., 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.
40. Petillo, J.J 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.
41. Rjasanow, S. and W. Wagner, "A stochastic weighted particle method for the Boltzmann equation," Journal of Computational Physics, vol. 124, no. 2, pp. 243-253, 1996.
42. Saad, Y., Iterative Methods for Sparse Linear Systems, 2nd ed., SIAM, 2003.
43. Schrock, C.R. and A. W. Wood, "Distributional Monte Carlo solution technique for rarefied gasdynamics," Journal of Thermophysics and Heat Transfer, , vol. 26, no. 1, pp. 185-189, 2012.
44. Serikov, V.V., S. Kawamoto and K. Nanbu, "Particle-in-cell plus direct simulation Monte Carlo (PIC-DSMC) approach for self-consistent plasma-gas simulations," IEEE transactions on plasma science, vol. 27, no. 5, pp. 1389-1398, 1999.
45. Taccogna, F., Minelli, P., Bruno D., Longo S., and Schneider R., Kinetic divertor modeling, *Chem. Phys.*, Volume 398, Pages 27–32, 2012.
46. Vahedi, V., G. DiPeso, Simultaneous potential and circuit solution for two-dimensional bounded plasma simulation codes, J. Comput. Phys. 131(1) (1997) 149–163.
47. Vay, J.L., 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.
48. Verboncoeur, J.P., 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.
49. Villasenor, J., O. Buneman, Rigorous charge conservation for local electromagnetic field solvers, Comput. Phys. Commun. 69(2–3) (1992) 306–316.
50. Vincenti, W.G. and C. H. J. Kruger, Introduction to Physical Gas Dynamics, Malabar: Krieger Publishing Company, 1967.
51. Wang, J., 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.
52. Wang, J., 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.
53. Wang, J., P. Liewer, V. Decyk, 3D electromagnetic plasma particle simulations on a MIMD parallel computer, Comput. Phys. Commun. 87(1) (1995) 35–53.

54. Wu, J. S, W.-J. Hsiao, Y.-Y. Lian and K.-C. Tseng, "Assessment of conservative weighting scheme in simulating chemical vapour deposition with trace species," International journal for numerical methods in fluids, vol. 43, no. 1, pp. 93-114, 2003.
55. Wu, J.S, 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.