Current Capabilities of AFRL's Spacecraft Simulation Tool

Samuel J. Araki [†], Robert S. Martin[‡] David L. Bilyeu[§]
Air Force Research Laboratory, Edwards Air Force Base, CA 93524, USA

Lubos Brieda[¶]
Particle In Cell Consulting LLC., Thousand Oaks, CA 91362, USA

Carrie Hill[‖]
Aurora Engineering, Orono, ME 04473, USA

Maria Choi[**]
NASA Glenn Research Center, Cleveland, OH 44135, USA

Assessment of spacecraft integration issues is typically accomplished with a combination of numerical tools developed to simulate different regions with different key physics. For instance, a detailed study of spacecraft with an electric propulsion device requires a device model, a plume model, and a spacecraft charging model. AFRL's in-house spacecraft simulation tool, TURF, is now capable of performing all of these calculations in one simulation. In addition, the plume simulation capability has been expanded significantly to improve speed and accuracy. Other upgrades in TURF include adaptive mesh refinement (AMR) and dynamic load balancing. All of these upgrades are covered in this paper by providing detailed descriptions or pointing to relevant references. This paper also presents three example simulations demonstrating its multi-physics multi-scale capability.

I. Introduction

Electric propulsion (EP) devices are highly efficient when compared to chemical propulsion devices. As solar power and battery technologies continue to advance, EP devices are becoming more viable alternatives for both primary and secondary propulsion systems. Nevertheless, EP devices can induce additional issues such as sputtering, contamination, and electric charging of spacecraft [1, 2]. Plumes of electric thruster consist of multiple plasma species including singly and multiply charged ions, electrons, and neutral atoms. While most of the charged particles remain within the EP plume, slow ions created by charge-exchange (CEX) reactions tend to accelerate at high angles due to electrostatic forces, potentially colliding and damaging the spacecraft and its components even in regions with no line of sight of the thruster. The sputtered material can then redeposit and cover other spacecraft surfaces (e.g. cameras, solar arrays, etc), which could lead to degradation or reduction of their functionality. Another concern is electrical charging of spacecraft resulting from fluxes of charged particles from a thruster, potentially causing electrical sparking and failure of instruments and solar panels.

Assessment of these spacecraft integration issues is typically accomplished with a combination of numerical tools that have been developed to simulate different regions with different key physics. More specifically, three different numerical tools are used for complete assessment: (1) a device model, (2) a plume model, and (3) a spacecraft charging model. The device model simply serves as a source model and provides boundary conditions to the plume model, so the complexity of the model varies from devices to devices and depends on how accurate it needs to be. For a Hall-effect thruster (HET), the model can be prescribed current distributions as functions of energy and location calibrated from an experiment [2–6] or obtained from a physics-based model such as HPHall [7–9], Hall2De [10, 11], HYPHEN [12, 13], and PICPluS [6]. Another approach is to directly couple a device model with a plume model by

Distribution A: Approved for Public Release; Distribution is Unlimited. PA No. XXXXX

[†] Computational Scientist, Jacobs Technology Inc., In-Space Propulsion Branch (AFRL/RQRS), samuel.araki.ctr@us.af.mil

[‡] Computational Scientist, In-Space Propulsion Branch (AFRL/RQRS). Currently, Program Manager at Army Research Office.

[§] Modeling & Simulation Group Lead, In-Space Propulsion Branch (AFRL/RQRS)

[¶] Researcher, Particle In Cell Consulting LLC.

[‖] Researcher, Aurora Engineering

[**] R&D Engineer, Electric Propulsion Systems Branch

running both simultaneously and transferring particle data on the fly [14, 15]. In a plume model, ions from the device model are propagated until reaching the computational domain boundary or impacting the spacecraft surface. The majority of plume codes employ a hybrid particle-fluid model where heavy species particles (ions and neutral atoms) are solved by the Particle-in-Cell (PIC) model and electrons are approximated as a fluid at Boltzmann equilibrium. Codes that use this approach include COLISEUM [14], SPIS [2, 4, 16], EP2PLUS [13, 17], and openPlumeEP [6]. Variants of the model exist, which use the isothermal hydrodynamic equation to solve for ions (HallPlume2D) [11] or solve the quasi-neutral electron fluid model (MPIC) [8, 9, 18]. There also exist other hybrid and fully kinetic codes that have been used to study EP plumes but only applied to small spacecraft [19–21]. Lastly, a spacecraft charging model uses ion fluxes from a plume code to predict the spacecraft potential. This potential then affects the ion flux in the plume model, so rigorous calculation involves iteration between the two models. Several codes are capable of performing a charging calculation, including Nascap-2k [22], SPIS [16], and MUSCAT [23].

Thermophysics Universal Research Framework (TURF) [24, 25] is the in-house research code developed at the In-Space Propulsion Branch of the Air Force Research Laboratory. This framework is a general purpose code, but the Spacecraft module is built upon the framework to include operations specifically for spacecraft and EP devices. Development of TURF started in 2011 after realizing needs for a common numerical framework within the research group. In 2015, the Spacecraft module of TURF was initially created, and capabilities in the AFRL's legacy plume simulation tool, COLISEUM, were ported over to TURF and the Spacecraft module with different levels of improvements in terms of computational efficiency and algorithm [15]. Beyond the 2016 release version of TURF [15], new models such as the fast and accurate charge-exchange (CEX) collision calculation [26, 27], the radiosity view factor (RVF) model [28–31], and the interspecies fractional collision model [32] have been incorporated into TURF. More recent development involves incorporation of the adaptive mesh refinement (AMR) and the dynamic load balancing capabilities [35]. TURF and the Spacecraft model have been expanded beyond the EP plume model with additions of a Hall-Effect Thruster (HET) device model and spacecraft charging models [33, 34]. The legacy HET model, HPHall, has been ported over and adapted to fit in TURF. Also, several simple charging models have been implemented to calculate migration of charges between spacecraft surface elements and components.

The purpose of this paper is to provide a summary of TURF's current capabilities for spacecraft simulations and point to published papers that contain pieces of the capabilities for more details. The three sub-modules (HET, plume, and spacecraft charging) involve multiple time- and length-scales that are governed by ions, electrons, grid resolution, and electron motion within materials. The AMR capability is the key to the multi-physics and multi-scale simulation in TURF, enabling a fully coupled simulation from a HET device to a plume and to spacecraft. In addition, coupling the plume and spacecraft charging models is accomplished by using different time-step sizes for different species and the implicit solver for the charge migration on the spacecraft surfaces. The rest of this paper is organized as follows. Section II provides an overview of TURF's code design and covers computational meshes and parallelization strategies. Section III describes available methods for various physical models relevant to spacecraft simulations. Section IV demonstrates coupling of different sub-modules in different simulation scenarios. Finally, this paper closes with conclusions and future work in Section V.

II. Thermophysics Universal Research Framework (TURF)

This section briefly explains important aspects of TURF's code design (Section II.A), computational mesh (Section II.B), and parallelization strategy (Section II.C).

A. Code Design

TURF has several important features to provide a unified system for different numerical models, making the framework highly modular and thus flexible and expandable [24, 25, 35]. In other words, new capabilities can be rapidly integrated and tested without compromising the integrity of the existing code base. The modularity of the framework also enhance coupling and hybridization of different models. The key in achieving this code design is TURF's General Service Object (GSObject), encapsulating objects to provide common instructions for managing objects. GSObjects are connected in a tree hierarchical structure with a common ancestor containing the simulation setup data (`World`), and data defined in GSObjects are accessed by traversing through the object tree. Underneath `World` is a GSObject containing the domain data (`Domain`), followed by `Operators` that provide instructions for advancing a simulation. Each operator contains an 'init' function to extract required input data independently. Since operators access data in other operators only through the common parent object, they are decoupled such that one can be replaced by another without changing any parts of the code. Furthermore, "OpFactory" builds a list of available operators during compilation, and these operators can be added and reordered to create a new simulation without recompilation.

B. Computational Meshes

TURF currently supports different types of computational meshes to discretize a computational domain and a geometry. For a computational domain, TURF primarily uses a uniform Cartesian mesh but has recently been extended to perform AMR if needed. TURF also has support for a tetrahedral mesh but has not been actively used to date. On the other hand, a geometry is represented by a triangulated surface mesh, which can be imported as various file formats including Abaqus, Cart3D, Exodus, Gambit, OFF, and Universal. Here, we refer to volume and surface meshes as meshes for the computational domain and the geometry surfaces, respectively. Also, for clarity, we refer to volume and surface mesh elements as cell and element, respectively.

Once geometry surface meshes are introduced in the computational domain, the “Sugarcubing" routine groups cells in different contiguous regions and patches the regions across different subdomains to determine free-space, boundary, and interior regions. Also, for each cell, a list of elements that intersect the cell is created, which is then used to create another list of elements that intersect all the neighboring cells. The second list of elements is used to determine particle-surface intersection.

C. Parallel Computing

TURF has been developed with heterogeneous systems in mind, using a hybrid MPI and OpenMP parallel programming.* The computational domain is partitioned into subdomains, and the subdomains are distributed to different MPI processes. Within each subdomain, loop-level parallelization is performed when looping through mesh cells, surface elements, or particles. TURF’s predecessor, COLISEUM, depended on a static domain decomposition strategy for parallelization, while work-load was often not balanced due to the highly nonuniform particle distribution throughout the computational region in EP plume simulations. TURF uses the over-decomposition strategy and allows multiple subdomains per MPI process, adding much flexibility when statically or dynamically balancing work-loads and mitigating unbalanced load seen in COLISEUM [35]. While the particle decomposition strategy is not used as extensively as the domain decomposition strategy, TURF has demonstrated its use in a full-PIC simulation [36], distributing particles to different MPI processes and performing an all-reduce operation on the field data.

III. Spacecraft Simulations Using TURF

Spacecraft simulations are performed using combinations of TURF operators consisting of generic algorithms and operators defined in the Spacecraft module. The Spacecraft module consists of operators that are written specifically for spacecraft simulations such as an electron fluid solver for a HET and EP device particle source models. This section describes models used to perform plume (Section III.A), HET (Section III.B), and spacecraft charging simulations (Section III.C).

A. Plume Model for Electric Propulsion Devices

The basic plume simulation in TURF closely follows the model implemented in COLISEUM and uses the hybrid PIC-fluid approach where heavy species and electrons are solved by the PIC model and the Boltzmann-electron fluid model as described in Sections III.A.1 and III.A.2, respectively. Extensive verification studies against the COLISEUM code have been performed in Ref. [15]. Since the release version in 2016, many new models have been implemented into TURF to improve efficiency and/or accuracy of the existing models. These new models are listed below along with sections and references covering some details:

- Various source and sputtering models (Section III.A.1)
- Fast and accurate calculation of CEX collisions (Section III.A.1 and Refs. [26] and [27])
- Interspecies fractional collisions to improve statistics of CEX ions (Section III.A.1 and Ref. [32])
- Unmagnetized and magnetized quasi-neutral (detailed) electron fluid models (Section III.A.2)
- Particle VDF compression and RVF model for calculation of sputtered atom fluxes (Section III.A.3 and Ref. [37])
- Thin sheath approximation for ions (Section III.C and [34])
- Dynamic load balancing with over-decomposition (Ref. [35])

*Parallelization using Graphics Processing Units (GPU) was supported during the initial development phase but was eventually dropped.

1. Particle-In-Cell Model for Ions

Ions are advanced in time through a standard PIC cycle [40], following the steps given in Table 1. While all the basic operations remain the same as in Ref. [15], a few steps are slightly reordered to fix some artifacts.[†] Notable upgrades are new particle source and sputtering models, a more efficient particle data movement across subdomains and MPI processes (Particle Patch), a fast and accurate calculation of CEX collisions, and improved CEX ion statistics through interspecies fractional collisions. Particle source models available in TURF and the Spacecraft module are listed in Table 1. For sputtering, various models are available for normal yield, angular yield, and angular and speed ejection distributions of sputtered atoms as shown in Table 3. Particle patching routine is detailed in Ref. [35]. The elastic collision routine has been upgraded since the 2016 release version. Previous implementation used a curve-fit equation for differential cross-section and employed an acceptance-rejection sampling technique to determine post-collision angles [41]. The new implementation uses the classical scattering equation with a quantum interatomic potential [26] so the differential cross-section can be computed for any energies as long as assumptions made in the model hold. Some improvements from Ref. [26] have been made by incorporating a look-up table and simply using the CEX probability of $P_{\mathrm{CEX}} = 0.5$ together with a total elastic collision cross-section of $\sigma_{\mathrm{total}} = 2\sigma_{\mathrm{CEX}}$ as implemented in Ref. [27]. Finally, CEX ion statistics has been improved by effectively splitting particles when undergoing CEX collisions. For a Monte Carlo Collision (MCC) routine, this is not difficult to implement. However, for a Direct Simulation Monte Carlo (DSMC) collision routine, this could not be accomplished without the method described in the recently published paper [32].

[†] When particles are injected, they are created on the same source area but are assigned different time stamps randomized within one time-step. Until those particles are advanced by the push routine, they are bunched on one plain such that the field calculation would be slightly wrong. Therefore, the charge deposition routine is moved before the inject routine.

Table 1 Order of operations in each time-step for the PIC model

	Operations	**Description**
1	Deposit charge	Weight particle charge to the grid.
2	Field	Compute electric potential and field. See Section III.A.2.
3	Inject	Inject particles. See Table 2.
4	Advance	Electrostatic push, surface intersection calculation, generate sputtered particles, sample surface data from particles
5	Patch	Copy particles to a correct subdomain if they cross boundary of the original subdomain
6	Sort	Sort particles according to cells and species
7	Sample	Accumulate particle mass, momentum, and energy to compute field data
8	Collision	Apply momentum exchange (MEX) and charge exchange (CEX) collisions

Table 2 Particle source models

Operator Name[1]	**Description**
`BeamletImportOp`	An ion thruster model using beamlet data from an ion optics code
`BoxICOp`	Maxwellian particles created in a box
`FieldSourceOp`	A HET model injecting particles based on HPHall particle data
`IonThrusterOp`	An ion thruster model injecting Maxwellian particles based on the input current density profile (see Ref. [38])
`NormalMaxwellianOp`	Maxwellian particles injected from a surface (see Ref. [39])
`SourceParticleOp`	Particle injection from a surface using particle data written in a VTK format file
`SourceRPAOp`	Particle injection from a hemisphere using RPA data (see Ref. [15])
`SourceStageSphericalOp`	Particle injection from a hemisphere using VDFs stored as particle data (see Ref. [30])
`SourceVDFOp`	Particle injection from a surface using VDF data stored in a 3D grid (see Ref. [31])
`SpawnHPHallOp`	A HET model that runs HPHall and collects particles (see Ref. [14, 15]).

[1] All of these operators start with `MSPDist` in their names to indicate that they operate on particle data, but `MSPDist` is omitted here.

Table 3 Sputtering models

Distributions	Available models
Sputter yield at normal incidence (Normal yield)	(1) constant, (2) linear, (3) Matsunami et al. [42], (4) Yamamura and Tawara [43], (5) Kannenberg et al. [44], (6) Roussel et al. [45], (7) Garnier et al. [46], (8) Pencil et al. [47], and (9) Eckstein and Preuss [48]
Angular dependency of sputter yield (Angular yield)	(1) 4th order polynomial fit, (2) Bermúdez et al.[5], (3) Yamamura [49], (4) Eckstein and Preuss [48], and (5) Wei et al. [50]
Angular distribution of sputtered atoms	(1) cosine and (2) modified Zhang[1] [51]
Speed distribution of sputtered atoms	(1) constant and (2) Thompson distribution[2] [52–54]

[1] $f(E,\alpha,\beta,\phi) = \frac{1}{1-\cos(\beta)\sqrt{E^*/E}}\frac{\cos(\alpha)}{\pi}\left[1-\frac{1}{4}\sqrt{\frac{E^*}{E}}\left(\cos(\beta)\gamma(\alpha)+\frac{3}{2}\pi\sin(\beta)\sin(\alpha)\cos(\phi)\right)\right]$
where $\gamma(\alpha) = \frac{3\sin^2(\alpha)-1}{\sin^2(\alpha)} + \frac{\cos^2(\alpha)(3\sin^2(\alpha)+1)}{2\sin^3(\alpha)}\ln\left(\frac{1+\sin(\alpha)}{1-\sin(\alpha)}\right)$
E^* is the characteristic energy, β is the incident angle of a bombarding ion, α is the ejection polar angle of a sputtered atom, and ϕ is the ejection azimuthal angle of the sputtered atom with respect to the tangent vector of the incident ion.

[2] $g(V) = C_m\frac{V^3}{(V^2+V_b^2)^{3-2m}}$ and $V_b = \sqrt{\frac{2E_b}{M}}$
where C_m is the normalizing coefficient, V is the velocity, E_b is the binding energy, M is the molecular mass of the target material, and m is an energy-dependent parameter that varies from 0.5 at medium energies to 0 at low energies. [54]

2. Electron Fluid Model

Step 2 in Table 1 involves calculation of electric potential and field before injecting and advancing particles through electric forces. Conventional PIC simulations solve Poisson's equation, but solely using this approach in an EP plume is infeasible due to a large simulation domain size for a real satellite geometry, a wide range of plasma density as the plasma expands in space, and the Debye length limitation on mesh resolution. Therefore, quasi-neutrality is assumed when a cell volume is larger than the Debye sphere volume [55], and electric potential is computed by assuming equal electron and ion densities and solving electron fluid equations. Three models based on the fluid equations with different assumptions are available in TURF, including (1) the Boltzmann relation with isothermal and polytropic electron closure [56] and (2) the unmagnetized [8, 18] and (3) the magnetized quasi-neutral electron fluid model [9]. These solvers are detailed below. The systems of linear equations can be solved in TURF by Gauss-Seidel, bi-conjugate gradient method, or using the MUMPS library. As moving away from the thruster exit, plasma density rapidly decreases, and the Debye length potentially exceeds the cell size. This situation often arises in a geostationary orbit but not so much in a low Earth orbit. In this case, the quasi-neutral assumption can no longer be used, and potential in the rest of cells is solved by the Poisson's equation with electron density obtained from the Boltzmann relation as described below.

Inverting Boltzmann relation The simplest model inverts the Boltzmann relation and computes electric potential, assuming quasi-neutrality in the scale of the computational cells such that electron density can be obtained directly from ion density calculated by the PIC model. In the absence of magnetic fields and interspecies collisions, only the electrostatic and pressure forces remain in the electron momentum equation, assuming spatially uniform velocity and instantaneous response of electrons with a change in potential. Assuming that electrons behave as a perfect gas ($p_e = n_e k_B T_e$), two different expressions for electric potential can be obtained when the electron temperature is isothermal or follows the polytropic relationship.

$$\text{Isothermal: } \phi = \phi^* + \frac{k_B T_e^*}{e}\ln\left(\frac{n_e}{n_e^*}\right), T_e = \text{Const} \tag{1}$$

$$\text{Polytropic: } \phi = \phi^* + \frac{k_B T_e^*}{e}\frac{\gamma}{\gamma-1}\left\{\left(\frac{n_e}{n_e^*}\right)^{\gamma-1}-1\right\}, T_e = T_e^*\left(\frac{n_e}{n_e^*}\right)^{\gamma-1} \tag{2}$$

where k_B is the Boltzmann constant, T_e is the electron temperature, e is the elementary charge, n_e is the electron density, $*$ denotes the reference values, and γ is the polytropic index of any real number. Here, $\gamma = 1$ yields the isothermal (constant temperature) model and $\gamma = 5/3$ implies an adiabatic condition. The reference temperature and density values are chosen at some location near the thruster exist and are assumed constant throughout the domain. Previous studies have shown that this assumption does not hold in reality [19, 21]

Unmagnetized Quasi-Neutral Electron Fluid Model Starting from the electron momentum equation, the generalized Ohm's law is obtained by assuming steady-state, negligible electron inertia, and the friction term expressed as $\mathbf{R} = en_e\mathbf{j_e}/\sigma$ where $\mathbf{j_e}$ is the electron current density and σ is the electrical conductivity. Assuming that ion current is much smaller than electron current and applying the charge continuity such that $\nabla \cdot \mathbf{j} = 0$, the final equation becomes as follows.

$$\nabla \cdot (\sigma \nabla \phi) = \nabla \cdot \left(\frac{\sigma}{en_e} \nabla (n_e k_B T_e) \right) \tag{3}$$

where

$$\sigma = \frac{e^2 n_e}{m_e \nu_e}, \quad \Lambda = \frac{4\pi\epsilon_0 m_e c_e^2 \lambda_D}{e^2 Z}, \quad \lambda_D = \sqrt{\frac{\epsilon_0 k_B T_e}{e^2 n_e}}$$

$$\nu_e = \nu_{en} + \sum_{s=1}^{N_I} \nu_{ei_s}, \quad \nu_{en} = \sigma_{en}(T_e) n_n \sqrt{\frac{8k_B T_e}{\pi m_e}}, \quad \nu_{ei} = n_i Z \frac{4\sqrt{2\pi}}{3} \left(\frac{m_e}{k_B T} \right)^{3/2} \left(\frac{e^2}{4\pi\epsilon_0 m_e} \right)^2 \ln \Lambda$$

Here, ν is the collision frequency, $\ln \Lambda$ is the Coulomb logarithm, c_e is the electron thermal velocity, Z is the ion charge number, N_I is the number of ion species, σ_{en} is the cross-section for electron-neutral collision, ϵ_e is the vacuum permittivity, and subscripts e, i, and n denote electron, ion, and neutral, respectively. The expressions for ν_{en} and ν_{ei} are obtained from Refs. [1] and [57] (Eq.3.6-12 and Eq.8.10), respectively. While the formulation herein assumes negligible ion current, the effect can easily be incorporated by adding into the charge continuity equation. Once electric potential is determined, electron current can be solved directly by plugging the solution back into the momentum equation.

$$\mathbf{j_e} = \sigma \left(-\nabla \phi + \frac{1}{en_e} \nabla P \right) \tag{4}$$

The electron temperature is computed by solving the electron energy equation.

$$\nabla \cdot \left(\frac{5}{2} n_e k_B T_e \mathbf{v_e} - \kappa_e \nabla T_e \right) = \mathbf{E} \cdot \mathbf{j} - \sum_{h=1}^{N_i} \frac{3m_e}{m_h} \nu_{eh} n_e k_B (T_e - T_h) - n_e n_h C_i \varepsilon_i \tag{5}$$

where

$$\kappa_e = \frac{2.4}{\nu_e + \nu_{ei}/\sqrt{2}} \frac{k_B^2 n_e T_e}{m_e}$$

Here ε_i is the ionization energy and is 12.13 eV for xenon. This model is based on Ref. [18], but formulation is slightly different. For example, the mass conservation equation is no longer used here, and instead, electron current is directly solved by Eq. (4). The two produce identical results if the boundary condition for the former method is set correctly. In addition, Eqs. (3) and (5) are solved by the finite volume method in three-dimensions. Finally, until the solution develops to steady-state, it can be unstable due to the large difference in density from the plume to regions with no ions, causing electron speed to overshoot to unphysical values. For this reason, a floor function is used to set the minimum ion density value.

Magnetized Quasi-Neutral Electron Fluid Model Magnetic field effect has been shown to be significant in the near-field plume region [9], affecting the downstream plume map. Therefore, natural extension of the unmagnetized model (Eqs. (3) to (5)) is to include the magnetic field. The model implemented in TURF closely follows the model by Choi [9] but includes some updates as for the unmagnetized model.

$$\nabla \cdot \left(\sigma \underline{\underline{\mu}}^{-1} \nabla \phi \right) = \nabla \cdot \left(\underline{\underline{\mu}}^{-1} \frac{\sigma}{en_e} \nabla (n_e k_B T_e) \right) \tag{6}$$

where $\underline{\underline{\mu}}^{-1}$ is the inverse of mobility tensor given as

$$\underline{\underline{\mu}}^{-1} = \begin{pmatrix} 1+\mu^2 B_x^2 & \mu^2 B_x B_y + \mu B_z & \mu^2 B_x B_z - \mu B_y \\ \mu^2 B_x B_y - \mu B_z & 1+\mu^2 B_y^2 & \mu^2 B_y B_z + \mu B_x \\ \mu^2 B_x B_z + \mu B_y & \mu^2 B_y B_z - \mu B_x & 1+\mu^2 B_z^2 \end{pmatrix} \tag{7}$$

and μ is the electron mobility where $\mu = e/(m_e \nu_e)$. Equation (6) is stiff because of the off-diagonal terms in the mobility tensor, yielding numerically unstable solutions in many cases. In order to mitigate the stiffness, monotonicity preserving discretization for anisotropic diffusion is used [58]. Even with this discretization scheme together with a floor function for electron velocity, the solution can still be unstable until it reaches steady-state, causing the steady-state solution to deviate from the expected solution. This has been improved by using the Boltzmann relation until reaching steady-state and switching to the quasi-neutral electron fluid solver.

Similar to the method in the unmagnetized case, the electric potential values from solution of Eq. (6) are used to compute the electron current.

$$\mathbf{j_e} = \sigma \underline{\underline{\mu}}^{-1} \left(-\nabla\phi + \frac{1}{e n_e} \nabla P \right) \tag{8}$$

The electron energy equation remains exactly the same as Eq. (5).

Non-linear solver in a non-quasi-neutral region When the quasi-neutral assumption becomes invalid, the Poisson's equation needs to be solved for electric potential. The potential values determined from the Boltzmann inversion model are used as the Dirichlet boundary condition, and the Poisson's equation is solved for the rest of non-quasi-neutral cells, while still approximating electrons as a fluid in Boltzmann equilibrium.

$$\nabla^2\phi = -\frac{\rho_i - \rho_e}{\varepsilon_0} = -\frac{\rho_i - \rho_e^* \exp\left(\frac{\phi-\phi^*}{T_e^*}\right)}{\varepsilon_0} \tag{9}$$

The reference values for temperature and potential are determined from far-field data. Equation (9) is a nonlinear equation with ϕ appearing in both the sides, which can be solved iteratively. This “quasi-neutral switch" is not compatible to the quasi-neutral electron fluid solvers in the current form, as a boundary condition for the fluid solver is poorly defined.

3. Radiosity View Factor Model for Sputtered Atoms

An EP plume code needs to also track sputtered atoms to study contamination of spacecraft, which is often done using computational particles and random sampling. However, tracking sputtered atoms together with the plume species would escalate the simulation run-time due to large differences in speed. Thompson distribution for ejection speed of sputtered atom has its peak near the binding velocity, V_b, which is often a few orders of magnitude smaller than the ion speed. Also, energy spread in Thompson distribution is large such that speed difference can be over a factor of 10 within the same species population. In order to mitigate this speed separation, recently, the non-Maxwellian RVF model has been successfully applied to sputtered atoms [37], significantly reducing the overall simulation run-time. The non-Maxwellian RVF model [31] requires velocity distribution functions (VDFs) of sputtered atoms at different spatial locations, which are created from the incident particle data accumulated throughout a simulation for sampling but compressed through applying particle merging technique [64]. The use of the particle merging scheme together with the non-Maxwellian RVF model for neutral atoms from a thruster has been demonstrated in the previously published paper [30].

B. Hall-Effect Thruster Model

Most of source models implemented in plume codes are static such that injection is assumed to remain the same in the course of a simulation. These models are very useful when changes in a long time-scale are only of interest, which is often the case when studying important metrics for spacecraft integration such as ion flux, erosion, and deposition rates. However, in order to gain a deeper understanding of facility effects, it is necessary to capture dynamic responses of diagnostics affected by the dynamic nature of a HET device, as these signals can inform about the degree of coupling between one another [60, 61]. Moreover, properly using HET data from an experiment or a simulation as a source model in a plume code is not always straight forward. For example, the RPA source model in TURF injects particles on a hemispherical virtual surface. This model assumes velocity directions normal to the surface, but in reality, there would be spread of velocity angles unless the injection radius is much larger compared to the thruster dimension such that the thruster can be approximated as a point source. Furthermore, injection of ions from a virtual surface leads to a numerical artifact of an empty region inside the hemisphere, resulting in electrostatic forces that slow down ions after being injected.

Coupling of HPHall and TURF has been accomplished by utilizing MPI to run both simultaneously and transfer particle data on the fly. This approach has solved the issues associated with the static device model. Nevertheless, further extending the degree of coupling and trying to simulate the entire facility are simply not practical with this approach. For this reason, a HET device model has been incorporated natively into TURF. The model closely follows HPHall, solving a quasi-1D electron fluid model on a mesh representing magnetic field lines while solving heavy species as discrete particles. The electron fluid solver in TURF is called HallTPM (Hall-effect thruster Thermalized Potential Model) and computes electron temperature and plasma potential in magnetized plasma. While numerous researchers have contributed additional models for mobility, near wall conductivity, and wall fluxes since the original code created by Fife in 1998 [7], HallTPM closely follows the original implementation presented in Fife's dissertation, with the major exception being the three-zone mobility model of Hofer et al. [62]. Another difference between the two models is that heavy species are solved in three-dimensions, reusing exactly the same routines as in the plume simulation.

C. Spacecraft Charging Model

Assessment of electrical damages on spacecraft is often done using software tools developed specifically for spacecraft charging calculations (e.g. Nascap-2k, SPIS, and MUSCAT). These tools can calculate fluxes of charged species due to various space environmental effects and predict migration of charges through spacecraft. SPIS is also capable of solving for an EP plume and predicting currents from an electric thruster. The built-in plume simulation capability is important as charged particle fluxes and spacecraft potential are highly coupled, and running the plume and charging calculations together will often lead to a faster solution when compared to running two models/codes alternately until convergence. Similar to SPIS, TURF includes a spacecraft charging model, enabling self-consistent solutions of the charged particle flux and spacecraft potential when combined with a plume simulation [33, 34].

In TURF, three different charging models are available depending on the material conductivity: (1) dielectric; (2) conductive; and (3) charge propagation. Different spacecraft components are solved by appropriate charging models and are coupled through a common bus with some capacitance. A solution from the combined plume and charging simulation was often unstable due to large differences in important timescales. For this reason, the charge propagation calculation is performed implicitly, and a smaller time-step is used to compute the total charge on each surface element. Using different time-step sizes in plume and charging calculations leads to incorrect transient solution, but this is often not a problem as the correct steady-state solution is attainable and is often only needed. Moreover, uncertainty is already large due to the assumptions used for capacitance.

In addition to the charge migration model, an accurate current collection model to spacecraft surfaces is necessary when predicting spacecraft potential. Currently, TURF's charging model only considers currents from an EP thruster and thermal population, but currents from photoelectrons and secondary electrons as well as other environmental effects can be easily added if needed. Ion currents from the plume simulation are directly obtained by the discrete particles, while ion velocity before impact is altered considering the potential difference between plasma and surfaces as described in Ref. [34]. When plasma density is sufficiently large such that quasi-neutral assumption holds, electron current can be obtained by using the thin sheath approximation. An electron current is calculated using an expression obtained by integrating the Maxwellian VDF between appropriate velocities. Assuming a bulk speed of zero, the electron current becomes as follows:

$$J_e = \begin{cases} J_{e0} \exp\left(-\frac{e(\phi-\phi_w)}{k_B T_e}\right) & \text{for } \phi_w \le \phi \\ J_{e0} & \text{for } \phi_w > \phi \end{cases} \tag{10}$$

where J_{e0} represents the one-way flux given by $J_{e0} = en_e v_{\text{th}} A/4$, v_{th} is the mean speed of thermal electrons given by $v_{\text{th}} = \sqrt{8k_B T_e/\pi m_e}$, A is the surface area, and ϕ_w is the wall potential. Unlike the Boltzmann inversion model for the electron fluid, the quasi-neutral electron fluid model can predict a finite electron bulk velocity based on potential and pressure gradients. The contribution to electron current can be incorporated by using the streaming Maxwellian equation and integrating from $v - v_n$ to ∞.

$$J_e = \int_{v-v_n}^{\infty} e(v+v_n)\sqrt{\frac{m}{2\pi k_B T_e}} A \exp\left(-\frac{m_e v^2}{2k_B T_e}\right) dv \tag{11}$$

where v is some velocity satisfying $m_e(v+v_n)^2/2 = e(\phi - \phi_w)$ and v_n is the bulk velocity normal to the surface. Also, setting $v = 0$, the saturation electron current can be calculated. The resultant expressions are as follow:

$$J_e = \begin{cases} J_{e0} \exp\left(-\frac{m_e}{2k_B T_e}(v-v_n)^2\right) + 2J_{e0}\frac{v_n}{v_{\text{th}}}\left[1 - \text{erf}\sqrt{\frac{m_e}{2k_B T_e}}(v-v_n)\right] & \text{for } \phi_w \le \phi - \frac{m_e v_n^2}{2e} \\ J_{e0} \exp\left(-\frac{m_e}{2k_B T_e}v_n^2\right) + 2J_{e0}\frac{v_n}{v_{\text{th}}}\left[1 - \text{erf}\left(-\sqrt{\frac{m_e}{2k_B T_e}}v_n\right)\right] & \text{for } \phi_w > \phi - \frac{m_e v_n^2}{2e} \end{cases} \tag{12}$$

Note that setting $v_n = 0$ in Eq. (12) reduces to Eq. (10).

IV. Demonstration of Multi-Physics Multi-Scale Simulations for Spacecraft Integration

Three simulations with different combinations of physical models and relevant time-scales are set up to demonstrate the multi-physics capability in TURF: (1) contamination in a chamber (Section IV.A), (2) EP TEMPEST simulation (Section IV.B), and (3) a full spacecraft simulation (Section IV.C). In all of these simulations, xenon is used as the propellant gas.

A. Contamination in a chamber

The first example demonstrates an EP plume simulation in a cage [63] with particle ions and a constant background gas and finds deposition rates of sputtered atoms caused by ion bombardment. In a plume of a HET device, the plume ion speed is on the order of 30 km/s while Thompson's distribution for sputtered atoms' ejection speed peaks near V_b, which can be significantly smaller than the ion speed. As the simulation time-step size is restricted by the ion speed such that ions do not travel more than one cell in one time-step, slow sputtered atoms need significantly more number of time advancement cycles to reach steady-state, greatly increasing the overall simulation run-time. For this reason, it is beneficial to split the simulation into two, solving for ions first and then for sputtered atoms with larger time-step sizes. However, this approach requires a storage of the incident particle distribution, which is not often trivial without using a large amount of memory or by sacrificing statistics. The approach taken here employs the particle merging

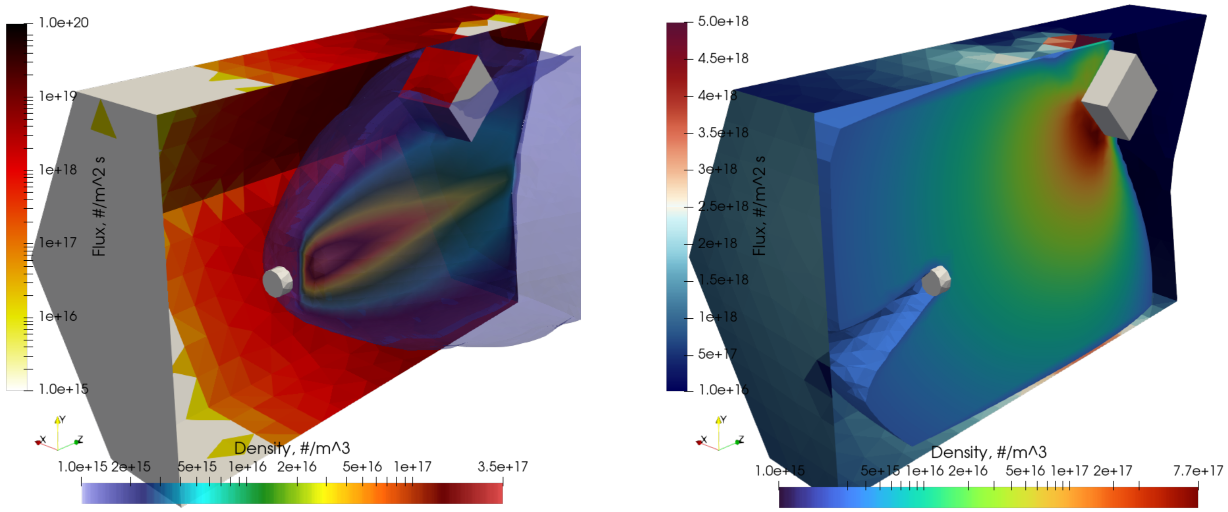

Fig. 1 (Left) Ion density and flux from a simulation of an EP plume in a chamber. In order to use the RVF model, VDFs of ions are sampled as particle data and compressed via the particle merging algorithm. (Right) Sputtered atom density and flux computed from the radiosity view factor model. Using the VDF data of incident ions, VDFs of sputtered atoms are constructed and used in the RVF model.

algorithm [64] to compress data on the fly without significantly perturbing conserved moments. Incident ions are grouped by surface element IDs and are merged within the same group.

The computation region spans from (−1.7, −1.6, −0.2) to (1.7, 1.6, 4.6) in units of meters with a grid spacing of 0.1 m in each direction. As described above, this example is split into two parts, subsequently solving for ions and then sputtered atoms. For the first part, a time step of 0.1 μs is used, and the simulation runs for 100, 000 time steps, leading to total simulation time of 10 ms. Using SourceRPAOp (see Table 2), xenon ions are injected on a hemisphere with a radius of 0.3 m with respect to the thruster center according to RPA data presented in Ref. [15] and a mass flow rate of 10^{-4} kg/s. All types of collisions are turned off in this example. Xenon neutral density is set to 10^{-5} Torr (3.22×10^{17} #/m^3). For the Boltzmann inversion model, reference values of $\gamma = 1.3$, $n_e^* = 3.11 \times 10^{17}$ #/m^3, $T_e^* = 8.016$ eV, and $\phi^* = 35.82$ V are used along with a floor plasma density of 10^{12} #/m^3. A box-shaped object is placed downstream of the thruster, and sputtering is only enabled from surfaces of this object. As ions impact these surfaces, they are moved to different particle data in order to construct ion VDFs and compress the data on the fly.

For the second part, the constant yield model with $Y = 5$ atoms/ions is assumed, and modified Zhang's angular distribution [51] and Thompson's speed distribution [54] are used along with parameters of $E^* = 72$ eV, $V_b = 316$ m/s, and $m = 0.23$. Once VDFs for sputtered atoms from different surface elements are constructed using the incident ion distributions, it can be used to either inject particles from SourceVDFOp or use in the non-Maxwellian RVF model [31]. Left and right figures in Fig. 1 show the ion density and flux from the first simulation and the sputterant density and flux from the second simulation using the RVF model, respectively. The distribution of sputtered atoms from this simulation is nearly identical when compared to the result from the sputtered atoms simulated as particles. Nonetheless, the simulation run-time was about 10 times faster.

B. EP TEMPEST Simulation

The second example simulates the EP TEMPEST set-up [63] with the plume model and the electron current collection model. The unmagnetized quasi-neutral electron fluid model is used in place of the Boltzmann inversion model. The cage is assumed to be grounded, so no charge migration calculation is actually performed. Instead, total current to each segment of the cage are recorded throughout the simulation. The same particle source model is used as in the previous example, but the injection radius is reduced to 0.01 m to eliminate numerical artifacts from an empty region between the injection surface and the thruster. The background neutral density is set to 3.22×10^{17} #/m^3 throughout the computational domain. For the electron fluid model, a floor plasma density of 10^{15} #/m^3 is used to prevent solutions from numerically overshooting due to the large difference in density near the edge of a plume. As boundary conditions for Eq. (3), electric potential of 0 V is assigned throughout all the geometry elements except for the thruster face (set to 10 V). Similarly, electron temperature of 1 eV is assumed throughout the cage surfaces, while $T_e = 3$ eV is assumed on the thruster face and the gradient is set to zero on the other thruster surfaces.

Since the quasi-neutral electron fluid model implemented in TURF has never been rigorously verified and validated against any experimental data, simulation results presented herein are still preliminary and are not meant to accurately represent the physics in any case. Figure 2 shows ion density, potential, electron temperature, and magnitude of electron current. While the ion density and the potential plots look reasonable, there are a few unexpected features in electron temperature and current. Along the center line, electron temperature decreases as expected until it hits a local minima

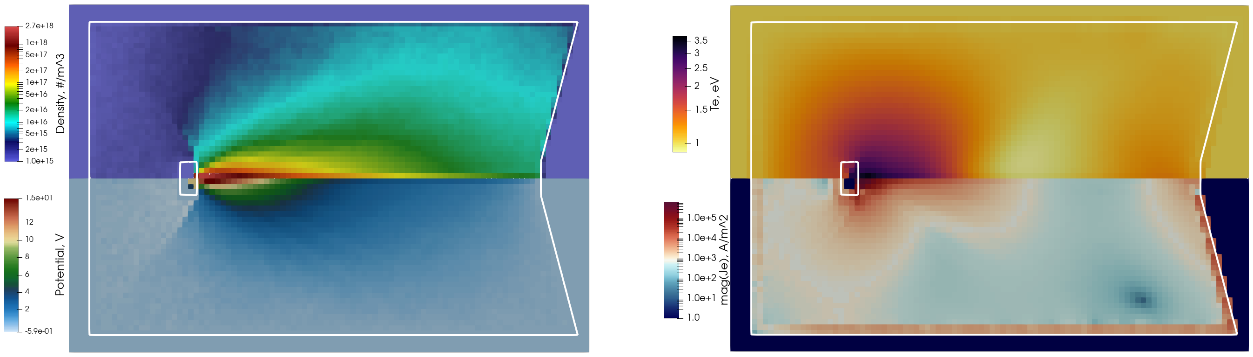

Fig. 2 (Left) 2D slice of ion density and electric potential. (Right) 2D slice of electron temperature and magnitude of electron current. White lines represent geometry boundaries.

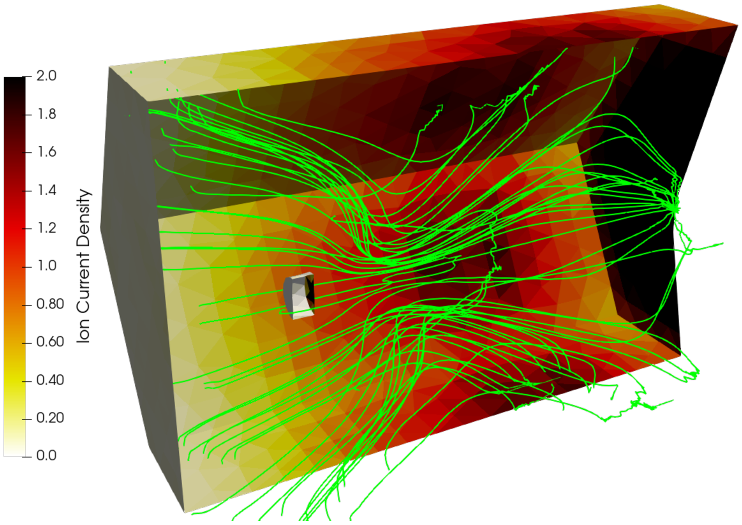

Fig. 3 Green lines represent the electron current streamlines. The colors on the surfaces represent the ion current density in A/m^2.

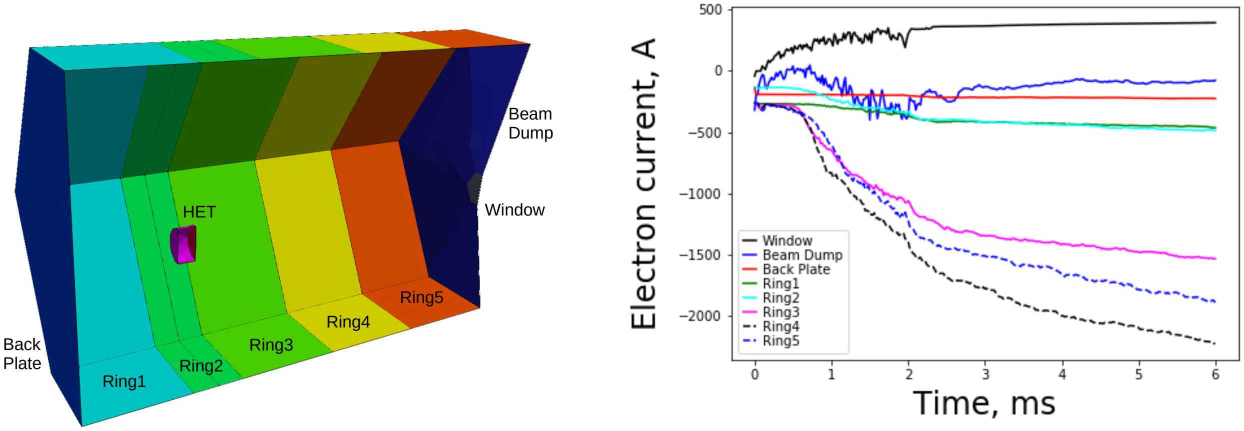

Fig. 4 Integrated electron currents on different components as a function of time.

and starts to increase towards the downstream walls. It is unclear if the local minima is physical, but it can well be a numerical artifact due to an incorrect boundary condition for electron temperature. Unlike electric potential, an appropriate boundary condition for electron temperature is not readily available. The boundary conditions might have also caused electron current to be discontinuous near the geometry as shown in Fig. 2. Figure 3 shows the streamlines of electrons and ion current density on the walls. Furthermore, Fig. 4 shows the cage geometry with different component names (left) and the total current on these components (right). Because of the inflated electron current near the boundary, the result is unlikely correct, but a validation against experimental data is subject to future work.

C. Full Spacecraft Simulation

The last example demonstrates native coupling of three different spacecraft models (HET, plume, and spacecraft charging models), which have traditionally been handled by three independent codes. In a plume simulation, plasma density varies by several orders of magnitude, as the slow ions created in the plume ($n \sim 10^{17}$ #/m^3) damage spacecraft components in regions with no line of sight of a HET channel ($n < 10^{14}$ #/m^3). However, by using the quasi-neutral assumption, Debye length no longer drives the cell size unlike a full PIC model. Instead, the cell size is driven by many other factors including the computational domain size, the allowable simulation length, and the accuracy requirement. Ideally, the cell size should be set as large as possible to minimize the simulation run-time, but pushing too far leads to a under-resolved plasma plume and thus an incorrect ion flux distribution to spacecraft surfaces. A rule of thumb from

Ref. [65] was to use at least four cells across the HET exit diameter. Then, the simulation time-step size is set by the Courant-Friedrichs-Lewy (CFL) number such that one or more time-steps are required for high-energy primary ions to travel through one cell. The larger the spacecraft is, the larger the computational domain becomes, and using the same cell size across the whole domain is not often easy when the computational domain spans over 10 m. Moreover, the cell size in the HET model is even much smaller than the rule-of-thumb cell size for a plume, so a constant mesh resolution is simply not feasible if the plume model is to be natively coupled to the HET model. Therefore, the AMR capability in TURF is the enabler of the native coupling of the HET and plume models. The coupling of EP plume and charging models was already demonstrated in Ref. [33, 34], which was enabled by the use of non-matching time-step sizes in different models and an implicit solver for charge migration. Although inconsistent time-step sizes led to inaccurate time-dependent solution, it was shown to produce the correct steady-state solution.

The simulation performed herein uses a mock-up of the SMART-1 satellite with dimensions obtained from Ref. [2]. The satellite has one SPT-100 thruster, and the parameters used in HallTPM are based on the HPHall inputs calibrated against some throttle condition. [‡] The simulation domain only covers one solar array and the entire spacecraft bus. Figure 5 shows a slice of the entire computational mesh with additional zoomed up views of the mesh near the thruster region. Mesh refinement is performed for up to six levels, and the finest mesh resolution is applied to all computational cells used in the HallTPM calculations, while the other mesh resolutions are set based on spatial coordinates.

Figure 6 shows a snapshot of the electric potential and electron temperature computed by the HallTPM model. Electron temperature is unrealistically high, but properly calibrating the mobility coefficients might bring the electron temperature result down to a reasonable range. Also, the mesh resolution is too coarse for the HallTPM simulation. For some reasons, further increasing the mesh refinement level caused the simulation to crash, which still needs to be investigated. For electric potential, the magnitude seems reasonable, but the values along the inner wall of the channel raise an issue with the boundary treatment near walls. In any event, this effect should be mitigated when a higher mesh resolution is achieved in this region.

Figure 7 shows a 2D slice of the ion density computed by the HallTPM and plume models, which is obtained by time-averaging ten snapshots in ParaView. The large problem size did not permit the simulation to complete, so the particle count for neutral atoms was still developing. Also, the simulation was terminated before time-averaging starts

[‡] Due to differences in models and implementations in HallTPM and HPHall, they do not predict the same thruster behavior when using the same mobility coefficients and model. Instead, both the codes have to be calibrated independently to the same throttle condition.

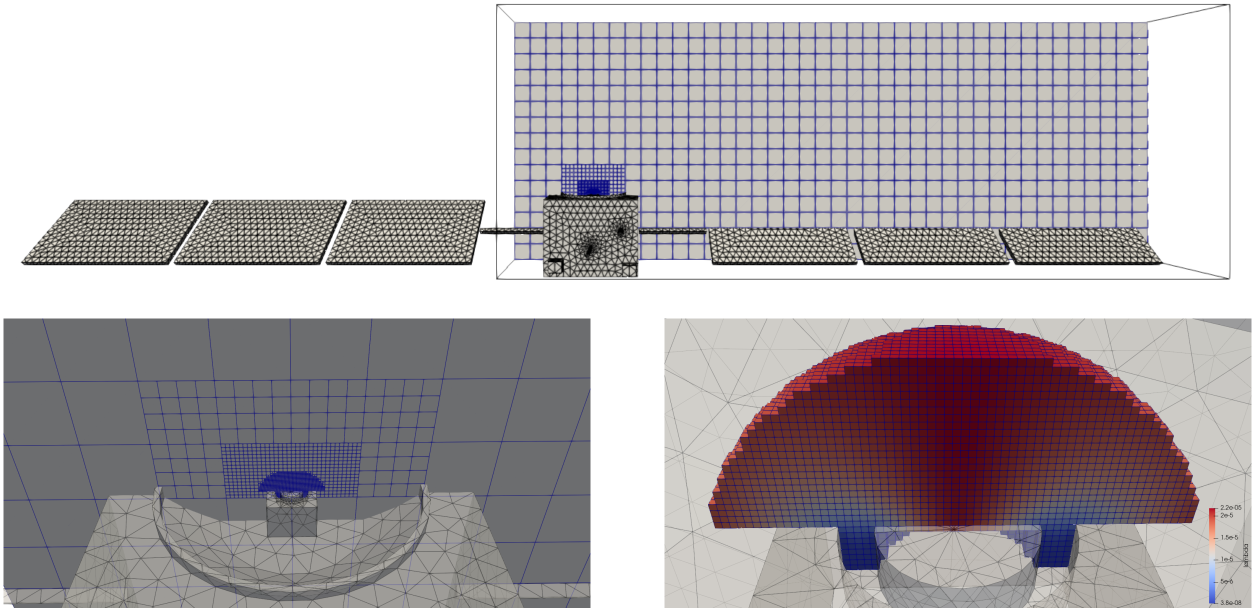

Fig. 5 (Top) A surface mesh of SMART-1 satellite, and a volume mesh with AMR. The box represents the computational domain. (Bottom left) Zoomed up view of the finer mesh region. (Bottom right) Finest mesh resolution is used in HallTPM.

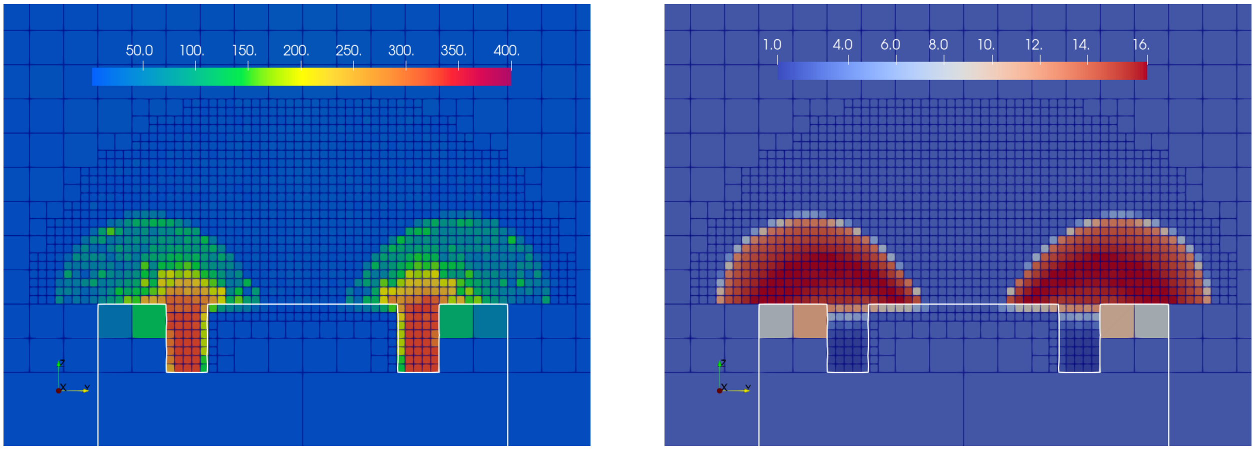

Fig. 6 Electric potential (Left) and electron temperature (Right) data computed by the HallTPM model.

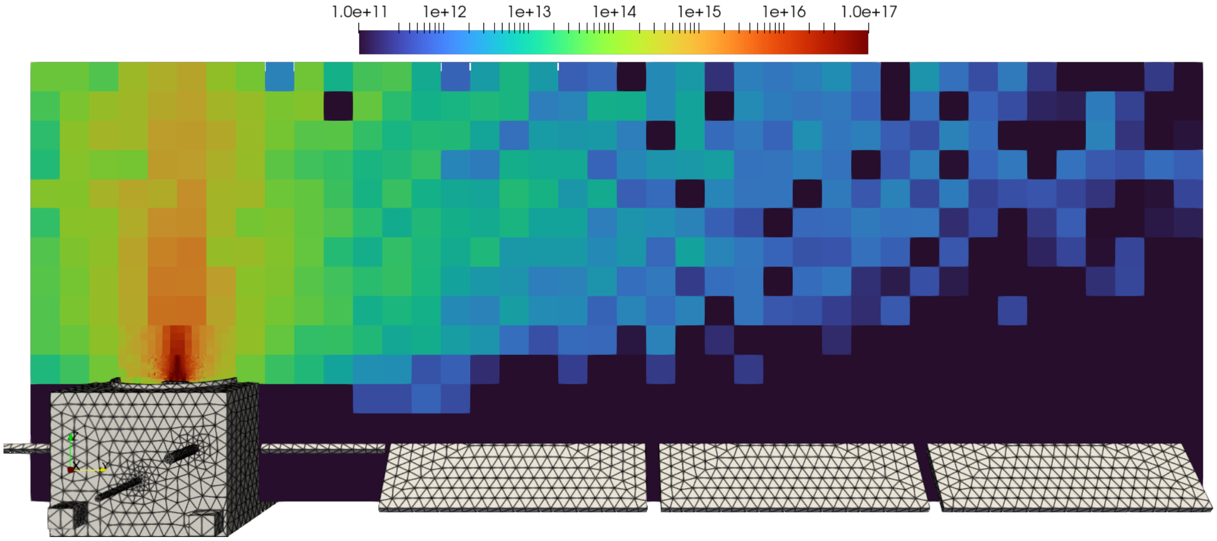

Fig. 7 Ion density (#/m^2· s) in the plume simulation.

so that the CEX ions behind the thruster exit plane are not captured in these snapshots. For this reason, no charging calculation has been performed. The results presented here will be updated before the conference.

V. Conclusion

This paper presented the current capabilities of TURF and its spacecraft simulation module. After the first release version of TURF in 2016, various improvements in speed and accuracy over the standard plume simulation have been incorporated into TURF: (1) additional particle source models and sputtering models, (2) the fast and accurate elastic collision calculation, (3) the fractional collision method to improved CEX ion statistics, (4) the RVF model for accelerated calculations of sputtered atoms, (5) the unmagnetized and magnetized quasi-neutral (detailed) electron fluid models, (6) the dynamic load balancing method using over-decomposition, and (7) particle data transfer between MPI processes. Beyond the plume capability, TURF now includes an HET model (HallTPM) and spacecraft charging models, enabling a simulation of the entire spacecraft without relying on other codes. The recently added AMR capability is the key for this, as the full spacecraft simulation is hardly feasible without reducing grid resolution away from a thruster.

This paper also demonstrated multi-physics multi-scale coupling of models in TURF through three different examples. These include (1) a plume simulation with sputtered atoms solved via the RVF model, (2) a plume simulation with the unmagnetized quasi-neutral electron fluid model and the surface charging model, and (3) a full spacecraft simulation with HallTPM, the plume model, and the spacecraft charging model. Although the results presented herein have not been rigorously validated, they have successfully proven the extensive capabilities of TURF.

Many improvements still have to be made before fully utilizing TURF in production simulation runs. These include, but not limited to these are, (a) enabling higher refinement levels in the AMR routine, (b) implementing the first-order weighting and interpolation near the grid boundary in AMR, (c) upgrading the quasi-neutral electron fluid model to work with AMR, (d) performing validation studies of the plume potential solvers, (e) coupling HallTPM and the plume with the quasi-neutral electron fluid model, (f) improving HallTPM and the spacecraft charging model, (g) resolving inability to resize particle buffer size after the first iteration, and (h) implementing a circuit model and a cathode model.

References

[1] Goebel, D. M., and Katz, I., *Fundamentals of Electric Propulsion: Ion and Hall Thrusters*, John Wiley & Sons, New York, 2008. https://doi.org/10.1002/9780470436448.

[2] Tajmar, M., Sedmik, R., and Scharlemann, C., "Numerical Simulation of SMART-1 Hall-Thruster Plasma Interactions," *Journal of Propulsion and Power*, Vol. 25, No. 6, 2009, pp. 1178–1188. https://doi.org/10.2514/1.36654.

[3] Passaro, A., Vicini, A., Nania, F., and Biagioni, L., "Numerical Rebuilding of SMART-1 Hall Effect Thruster Plasma Plume," *Journal of Propulsion and Power*, Vol. 26, No. 1, 2010, pp. 149–158. https://doi.org/10.2514/1.36821.

[4] Wartelski, M., Ardura, C., Theroude, C., Reissner, A., Tajmar, M., and Gengembre, E., "The Assessment of Interactions between Spacecraft and Electric Propulsion Systems Project," *32nd International Electric Propulsion Conference*, Wiesbaden, Germany, 2011, pp. 1–13. IEPC-2011-028.

[5] Bermúdez, L. M., Barnhart, K. J., Agathon-Burton, C. A., Basak, D., and Glogowski, M. J., "A Comprehensive Numerical Approach to the Modeling and Simulation of Plume Interaction Effects on Solar Electric Propulsion Spacecraft," *35th International Electric Propulsion Conference*, Georgia Institute of Technology, USA, 2017, pp. 1–14. IEPC-2017-238.

[6] Zitouni, B., Laube, J., Massaccesi, N., Sonneveld, V., and Peukert, M., "Thrusters Modelling, Propellant Choice and Plume Expansion: openPlumeEP Capabilities," *36th International Electric Propulsion Conference*, University of Vienna, Austria, 2019, pp. 1–12. IEPC 2019–211.

[7] Fife, J. M., "Hybrid-PIC Modeling of Electrostatic Probe Survey of Hall Thrusters," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 1998.

[8] Huismann, T. D., "Improving Hall Thruster Plume Simulation through Refined Characterization of Near-Field Plasma Properties," Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, 2011.

[9] Choi, M., "Improved Hall Thruster Plume Simulation by Including Magnetic Field Effects," Ph.D. thesis, University of Michigan, Ann Arbor, Michigan, 2016.

[10] Mikellides, I. G., and Katz, I., "Numerical simulations of Hall-effect plasma accelerators on a magnetic-field-aligned mesh," *Phys. Rev. E*, Vol. 86, 2012, p. 046703. https://doi.org/10.1103/PhysRevE.86.046703.

[11] Lopez Ortega, A., Katz, I., Mikellides, I. G., and Goebel, D. M., "Self-Consistent Model of a High-Power Hall Thruster Plume," *IEEE Transactions on Plasma Science*, Vol. 43, No. 9, 2015, pp. 2875–2886. https://doi.org/10.1109/TPS.2015.2446411.

[12] Domínguez-Vázquez, A., Zhou, J., Fajardo, P., and Ahedo, E., "Analysis of the Plasma Discharge in a Hall Truster via a Hybrid 2D Code," *36th International Electric Propulsion Conference*, University of Vienna, Austria, 2019, pp. 1–24. IEPC 2019–579.

[13] Cichoski, F., Domínguez-Vázquez, A., Merino, M., Fajardo, P., and Ahedo, E., "3D Simulations of a Magnetized Hall Effect Thruster Plume," *36th International Electric Propulsion Conference*, University of Vienna, Austria, 2019, pp. 1–16. IEPC 2019–460.

[14] Brieda, L., and Keidar, M., "Multiscale Modeling of Hall Thrusters," *32nd International Electric Propulsion Conference*, Kurhaus, Wiesbaden, Germany, 2011, pp. 1–13. IEPC 2011–101.

[15] Araki, S. J., Martin, R. S., Bilyeu, D. L., and Koo, J. W., "SM/MURF: Current Capabilities and Verification as a Replacement of AFRL Plume Simulation Tool COLISEUM," *52nd AIAA/SAE/ASEE Joint Propulsion Conference*, Salt Lake City, Utah, 2016, pp. 1–29. https://doi.org/10.2514/6.2016-4939.

[16] Roussel, J.-F., Rogier, F., Dufour, G., Mateo-Velez, J.-C., Forest, J., Hilgers, A., Rodgers, D., Girard, L., and Payan, D., "SPIS Open-Source Code: Methods, Capabilities, Achievements, and Prospects," *IEEE Transactions on Plasma Science*, Vol. 36, No. 5, 2008, pp. 2360–2368. https://doi.org/10.1109/TPS.2008.2002327.

[17] Cichocki, F., Domínguez-Vázquez, A., Merino, M., and Ahedo, E., "Hybrid 3D Model for the Interaction of Plasma Thruster Plumes with Nearby Objects," *Plasma Sources Science and Technology*, Vol. 26, No. 125008, 2017, pp. 1–19. https://doi.org/10.1088/1361-6595/aa986e.

[18] Boyd, I. D., and Yim, J. T., "Modeling of the near field plume of a Hall thruster," *Journal of Applied Physics*, Vol. 95, No. 9, 2004, pp. 4575–4584. https://doi.org/10.1063/1.1688444.

[19] Wang, J., and Hu, Y., "On the limitations of hybrid particle-in-cell for ion thruster plume simulations," *Physics of Plasmas*, Vol. 26, No. 10, 2019, p. 103502. https://doi.org/10.1063/1.5111791.

[20] Korkut, B., Li, Z., and Levin, D. A., "3-D Simulation of Ion Thruster Plumes Using Octree Adaptive Mesh Refinement," *IEEE Transactions on Plasma Science*, Vol. 43, No. 5, 2015, pp. 1706–1721. https://doi.org/10.1109/TPS.2015.2415458.

[21] Jambunathan, R., and Levin, D. A., "Kinetic, 3-D, PIC-DSMC Simulations of Ion Thruster Plumes and the Backflow Region," *IEEE Transactions on Plasma Science*, Vol. 48, No. 6, 2020, pp. 2017–2034. https://doi.org/10.1109/TPS.2020.2988010.

[22] Mandell, M. J., Davis, V. A., Cooke, D. L., Wheelock, A. T., and Roth, C. J., "Nascap-2k Spacecraft Charging Code Overview," *IEEE Transactions on Plasma Science*, Vol. 34, No. 5, 2006, pp. 2084–2093. https://doi.org/10.1109/TPS.2006.881934.

[23] Muranaka, T., Hosoda, S., Kim, J.-H., Hatta, S., Ikeda, K., Hamanaga, T., Cho, M., Usui, H., Ueda, H. O., Koga, K., and Goka, T., "Development of Multi-Utility Spacecraft Charging Analysis Tool (MUSCAT)," *IEEE Transactions on Plasma Science*, Vol. 36, No. 5, 2008, pp. 2336–2349. https://doi.org/10.1109/TPS.2008.2003974.

[24] Martin, R. S., and Koo, J. W., *Thermophysics Universal Research Framework - Infrastructure Release*, Air Force Research Laboratory (AFRL/RQRS), April 2015.

[25] Araki, S. J., Martin, R. S., Bilyeu, D., Koo, J. W., Tran, J., and Kawashima, K. A., *Tutorial for Thermophysics Universal Research Framework –TURF-IR 2017a–*, Air Force Research Laboratory (AFRL/RQRS), May 2017.

[26] Araki, S. J., and Wirz, R. E., "Ion-Neutral Collision Modeling Using Classical Scattering With Spin-Orbit Free Interaction Potential," *IEEE Transactions on Plasma Science*, Vol. 41, No. 3, 2013, pp. 470–480. https://doi.org/10.1109/TPS.2013.2241457.

[27] Araki, S. J., "Fast computation of high energy elastic collision scattering angle for electric propulsion plume simulation," *AIP Conference Proceedings*, Vol. 1786, No. 1, 2016, p. 170004. https://doi.org/10.1063/1.4967668.

[28] Araki, S. J., "Sensitivity of Ion Flux on Spacecraft to Neutral Density in Hall Thruster Plume," *35th International Electric Propulsion Conference*, Georgia Institute of Technology, USA, 2017, pp. 1–12. IEPC-2017-539.

[29] Araki, S. J., "Extension of view factor model to free molecular flow with non-Maxwellian inflow condition," *AIP Conference Proceedings*, Vol. 2132, No. 1, 2019, p. 180010. https://doi.org/10.1063/1.5119668.

[30] Martin, R. S., and Araki, S. J., "Coupling Non-Maxwellian View Factor Model to Octree Based Particle VDF Compression for Accelerated Spacecraft-Plume Simulation," *36th International Electric Propulsion Conference*, University of Vienna, Austria, 2019, pp. 1–9. IEPC 2019-733.

[31] Araki, S. J., "Radiosity view factor model for sources with general distribution," *Journal of Computational Physics*, Vol. 406, 2020, p. 109146. https://doi.org/https://doi.org/10.1016/j.jcp.2019.109146.

[32] Araki, S. J., and Martin, R. S., "Interspecies Fractional Collisions," *Physics of Plasmas*, Vol. 27, No. 033504, 2020. https://doi.org/10.1063/1.5143145.

[33] Araki, S. J., and Barrie, A. C., "Electric Propulsion Plume Simulation Coupled with Spacecraft Charging," *54th AIAA/SAE/ASEE Joint Propulsion Conference*, Cincinnati, Ohio, 2018, pp. 1–11. https://doi.org/10.2514/6.2018-4906, aIAA 2018-4906.

[34] Araki, S. J., "Multiscale Coupling of Spacecraft Charging Model with Electric Propulsion Plume Simulation," *IEEE Transactions on Plasma Science*, Vol. 47, No. 11, 2019, pp. 4898–4908. https://doi.org/10.1109/TPS.2019.2945534.

[35] Araki, S. J., and Martin, R. S., "Dynamic load balancing with over decomposition in plasma plume simulations," *Journal of Parallel and Distributed Computing*, Vol. 163, 2022, pp. 136–146. https://doi.org/https://doi.org/10.1016/j.jpdc.2022.01.023.

[36] Tran, J., “Numerical Study of Current Driven Instabilities and Anomalous Electron Transport in Halleffect Thrusters,” Master’s thesis, University of California, Los Angeles, 2017.

[37] Araki, S. J., and Martin, R. S., “Calculation of Sputtered Atom Deposition via Non-Maxwellian Radiosity View Factor Model and Particle VDF Compression,” 2022. In Preparation.

[38] Brieda, L., “Development of the DRACO ES-PIC Code and Fully-Kinetic Simulation of Ion Beam Neutralization,” Master’s thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2005.

[39] Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, 1994.

[40] Birdsall, C. K., and Langdon, A. B., *Plasma Physics via Computer Simulation*, IOP Publishing Ltd., 1991. https://doi.org/https://doi.org/10.1201/9781315275048.

[41] Scharfe, M. K., Koo, J., and Azarnia, G., “DSMC Implementation of Experimentally-Based Xe^{+}+ Xe Differential Cross Sections for Electric Propulsion Modeling,” 2011, pp. 1085–1090. https://doi.org/10.1063/1.3562789.

[42] Matsunami, N., Yamamura, Y., Itikawa, Y., Itoh, N., Kazumata, Y., Miyagawa, S., Morita, K., Shimizu, R., and Tawara, H., “Energy dependence of the ion-induced sputtering yields of monatomic solids,” *Nuclear Data Tables*, Vol. 31, 1984, pp. 1–80. https://doi.org/https://doi.org/10.1016/0092-640X(84)90016-0.

[43] Yamamura, Y., and Tawara, H., “Energy dependence of the ion-induced sputtering yields of monatomic solids at normal incidence,” *Nuclear Data Tables*, Vol. 62, 1996, pp. 149–253. https://doi.org/https://doi.org/10.1006/adnd.1996.0005.

[44] Kannenberg, K., Khayms, V., Emgushov, B., Werthman, L., and Pollard, J., “Validation of Hall thruster plume sputter model,” *37th Joint Propulsion Conference and Exhibit*, Salt Lake City, Utah, 2001, pp. 1–10. https://doi.org/10.2514/6.2001-3986.

[45] Roussel, J.-F., Bernard, J., and Garnier, Y., “Numerical simulation of induced environment, sputtering, and contamination of satellite due to electric propulsion,” *Proceedings Second European Spacecraft Propulsion Conference*, 1997, pp. 517–522.

[46] Garnier, Y., Viel, V., Roussel, J.-F., and Bernard, J., “Low-Energy Xenon Ion Sputtering of Ceramics Investigated for Stationary Plasma Thrusters,” *Journal of Vacuum Science and Technology A*, Vol. 17, No. 6, 1990, pp. 3246–3254. https://doi.org/10.1116/1.582050.

[47] Pencil, E., Randolph, T., and Manzella, D., “End-of-life Stationary Plasma Thruster far-field plume characterization,” *32nd Joint Propulsion Conference and Exhibit*, Lake Buena Vista, Florida, 1996, pp. 1–28. https://doi.org/10.2514/6.1996-2709, URL https://arc.aiaa.org/doi/abs/10.2514/6.1996-2709.

[48] Eckstein, W., and Preuss, R., “New fit formulae for the sputtering yield,” *Journal of Nuclear Materials*, Vol. 320, No. 3, 2003, pp. 209–213. https://doi.org/https://doi.org/10.1016/S0022-3115(03)00192-2.

[49] Yamamura, Y., “An empirical formula for angular dependence of sputtering yields,” *Radiation Effects*, Vol. 80, No. 1-2, 1984, pp. 57–72. https://doi.org/10.1080/00337578408222489.

[50] Wei, Q., Li, K.-D., Lian, J., and Wang, L., “Angular dependence of sputtering yield of amorphous and polycrystalline materials,” *Journal of Physics D: Applied Physics*, Vol. 41, No. 17, 2008, p. 172002. https://doi.org/10.1088/0022-3727/41/17/172002.

[51] Yalin, A. P., Williams, J. D., Surla, V., and Zoerb, K. A., “Differential sputter yield profiles of molybdenum due to bombardment by low energy xenon ions at normal and oblique incidence,” *Journal of Physics D: Applied Physics*, Vol. 40, No. 10, 2007, pp. 3194–3202. https://doi.org/10.1088/0022-3727/40/10/025.

[52] Thompson, M. W., “II. The energy spectrum of ejected atoms during the high energy sputtering of gold,” *The Philosophical Magazine: A Journal of Theoretical Experimental and Applied Physics*, Vol. 18, No. 152, 1968, pp. 377–414. https://doi.org/10.1080/14786436808227358.

[53] Thompson, M., “The velocity distribution of sputtered atoms,” *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, Vol. 18, No. 1, 1986, pp. 411–429. https://doi.org/https://doi.org/10.1016/S0168-583X(86)80067-2.

[54] Betz, G., and Wien, K., “Energy and angular distributions of sputtered particles,” *International Journal of Mass Spectrometry and Ion Processes*, Vol. 140, No. 1, 1994, pp. 1–110. https://doi.org/https://doi.org/10.1016/0168-1176(94)04052-4.

[55] Brieda, L., “Multiscale Modeling of Hall Thrusters,” Ph.D. thesis, The George Washington University, Washington D. C., 2012.

[56] Oh, D. Y., "Computational Modeling of Expanding Plasma Plumes in Space Using a PIC-DSMC Algorithm," Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, 1996.

[57] Mitchner, M., and Kruger, C. H., *Partially ionized gases*, Wiley, New York, 1973.

[58] Sharma, P., and Hammett, G. W., "Preserving monotonicity in anisotropic diffusion," *Journal of Computational Physics*, Vol. 227, No. 1, 2007, pp. 123–142. https://doi.org/https://doi.org/10.1016/j.jcp.2007.07.026.

[59] Martin, R., "Conservative bin-to-bin fractional collisions," *AIP Conference Proceedings*, Vol. 1786, No. 1, 2016, p. 090003. https://doi.org/10.1063/1.4967609.

[60] Sugihara, G., May, R., Ye, H., hao Hsieh, C., Deyle, E., Fogarty, M., and Munch, S., "Detecting Causality in Complex Ecosystems," *Science*, Vol. 338, No. 6106, 2012, pp. 496–500. https://doi.org/10.1126/science.1227079.

[61] Eckhardt, D., Koo, J., Martin, R., Holmes, M., and Hara, K., "Spatiotemporal data fusion and manifold reconstruction in Hall thrusters," *Plasma Sources Science and Technology*, Vol. 28, No. 4, 2019, p. 045005. https://doi.org/10.1088/1361-6595/ab0b1f.

[62] Hofer, R., Katz, I., Goebel, D., Jameson, K., Sullivan, R., Johnson, L., and Mikellides, I., "Efficacy of Electron Mobility Models in Hybrid-PIC Hall Thruster Simulations," *44th AIAA/ASME/SAE/ASEE Joint Propulsion Conference & Exhibit*, 2008, pp. 1–29. https://doi.org/10.2514/6.2008-4924.

[63] MacDonald, N. A., "Electric Propulsion Test and Evaluation Methodologies for Plasma in the Environments of Space and Testing (EP TEMPEST)," Tech. Rep. AFRL-RQ-EDVG-2016-074, Air Force Research Laboratory, Edwards AFB, CA, USA, April 2016.

[64] Martin, R. S., and Cambier, J.-L., "Octree particle management for DSMC and PIC simulations," *Journal of Computational Physics*, Vol. 327, 2016, pp. 943–966. https://doi.org/https://doi.org/10.1016/j.jcp.2016.01.020.

[65] Araki, S. J., Bilyeu, D. L., Koo, J. W., Martin, R. S., Kubota, K., Koga, K., Cho, S., Kinefuchi, K., Hatta, S., and Ueda, H. O., "Convergence of Stochastic Models for Electric Propulsion Plume Simulation," *36th International Electric Propulsion Conference*, University of Vienna, Austria, 2019, pp. 1–15. IEPC 2019-156.