# An Iterative Tracer Particle Approach for Steady-State Simulation of Electrospray Thruster Plumes

**IEPC-2024-114**

*Presented at the 38th International Electric Propulsion Conference, Toulouse, France*
*June 23-28, 2024*

Yethinder Ragav Lakshmi kumar* and Oliver Jia-Richards†
*University of Michigan, Ann Arbor, Michigan, USA*

**Current methods for simulating the evolution of an electrospray thruster plume in the field-free region, while effective, require substantial computational resources. This paper introduces a potentially computationally-efficient alternative, termed the iterative tracer particle method, for simulating the plume of an electrospray thruster in steady state. This method could drastically reduce computational time while providing an approximate solution. Detailed within this work are the method's design, its validation against the traditional $n$-body method, and the integration of the fragmentation process. For a 2D test case simulation, the iterative tracer particle method required only 30 seconds, achieving a speed that is 1233 times faster than a simplified $n$-body approach, which took 10.3 hours. The memory-efficient architecture of this method, which resulted in using 3514 times less memory in this specific test case, allows for simulations to be run on personal computers. The efficiency offered by this method could enable the solving of inverse problems to infer thruster emission properties from measurements taken in a laboratory environment.**

## I. Introduction

Electrospray thrusters are a form of electrostatic propulsion that generate thrust in the range of 1 $\mu$N–1 mN, and are gaining traction for modern space missions due to their compact size and scalability [1–3]. The principle of operation for electrospray thrusters involves extracting charged particles from a conductive liquid and then accelerating the charged particles through an electric field. The two main classes of electrospray thruster differ by their propellant type: field-emission electric propulsion (FEEP) systems use liquid metals while ionic-liquid ion sources (ILIS) use ionic liquids. There are also two characteristic emission modes. In the droplet mode, a continuous jet is formed which eventually breaks up into a collection of droplets, whereas the ion mode involves individual ions and ion clusters being "evaporated" directly from a static meniscus. Most practical devices operate in a mixed mode where both droplets and ions are emitted.

A basic electrospray thruster consists of an emitter and an extractor, biased with respect to each other by a potential difference of $\sim$1–10 kV. The electric field that forms between the emitter and extractor extracts charged particles from the emitter and accelerates them out of the thruster through an orifice in the extractor. The emitter and extractor are typically separated by a few tens of micrometers depending on the type of thruster. The spatial aspects of the electrospray thruster's operation can be segmented into three distinct domains: the emission region where the charged particles are evaporated from the liquid propellant ($\sim$1 $\mu$m); the acceleration region, generally between the electrodes, where the charged particles are accelerated out of the thruster ($\sim$1 mm); and the field-free region beyond the thruster exit plane where the charged particles may interact with the surrounding environment ($\gtrsim$1 mm). The small scale of the emission and acceleration regions makes it nearly impossible to take direct measurements of the acceleration process without interfering with the thruster's normal operation. This limitation restricts the types of measurements that can be taken and our understanding of the physics at play in these regions. Measurements are typically taken at least a few

*Ph.D. Candidate, Department of Aerospace Engineering, yethi@umich.edu.
†Assistant Professor, Department of Aerospace Engineering, oliverjr@umich.edu.

centimeters away from the thruster, in the field-free region, and in the case of time-of-flight measurements, sometimes even more than a meter away.

The requirement that any measurements (e.g., spatial distribution of the emitted current density) be taken in the field-free region means that the measurements may be impacted by post-emission processes (e.g., inter-particle forces or plume-surface interactions). Some of these impacts would not influence the performance of the thruster, but could influence indirect estimates of thruster performance characteristics derived from the measurements. For instance, inter-particle Coulomb forces would tend to increase the angular spreading of the emitted beam, reducing the angular efficiency of the beam estimated from the angular distribution of current density. Similarly, inter-particle Coulomb forces would alter the angular distribution of different species in the emitted beam, potentially creating an artificial deficit of lighter species along the beam centerline [4]. Since time-of-flight measurements are commonly taken along the beam centerline, the mass flow rate of the thruster as estimated from integrating time-of-flight data would be artificially inflated.

In order to eliminate the influence of post-emission processes on indirect estimates of the thruster performance characteristics, a method is required for inferring the beam parameters immediately after emission from measurements taken downstream of the thruster. Solving this inverse problem (i.e., determining parameters from measurements) requires the ability to solve the forward problem (i.e., determining measurements from parameters) multiple times for a variety of different parameter settings. Therefore, the ability to simulate the evolution of an electrospray thruster plume after emission in a computationally-simple manner is essential. This paper introduces a new simulation framework, named iterative tracer particle (ITP), aimed at simulating the electrospray plume in the field-free region at significantly lower computational times than current alternatives. The fundamental steps of the algorithm are explained and validated against a widely adopted simulation method, the $n$-body approach, through a simple test case. Subsequently, the phenomenon of fragmentation of ion clusters is also implemented. The implications and limitations of this method are discussed, and future work is outlined. This study signifies the initial phase in the development of this innovative approach.

## II. Background

In 1994, Ganán-Calvo et al. introduced a significant advancement in electrospray simulation by conducting the first in-depth analysis of droplet dynamics. The simulation results were validated by comparing them with actual experimental data, establishing a foundational framework for future research in this field [5]. Since then, several algorithms have been developed and tested to simulate the electrospray plume with a focus on the different domains of the electrospray process. Starting from the emission region, two major models currently used are molecular dynamics (MD) [5–9] and electrohydrodynamics (EHD) [10–12], which concentrate on the physics of Taylor cone emission. These methods are applied to the emitter and emission regions extending no greater than 5 $\mu$m.

For the acceleration region, various methods are utilized, with Lagrangian $n$-body approach being particularly notable. This technique is widely used as a baseline due to its accuracy in capturing critical phenomena like particle-particle Coulombic interactions and ion cluster fragmentation [5, 10, 13, 14]. Its applicability is most effective just after the emission region, where particle density is high and inter-particle forces are significant, beyond which ($\gtrsim$5 $\mu$m) they are assumed to be negligible [12]. The computation time and memory requirements scale as $O(N^2)$, where $N$ is the number of particles being simulated, which imposes practical constraints on the approach due to limited time and memory. Alternatives to this method have been developed to handle the $\sim 10^8$ particles needed for simulation of an electrospray thruster operating in the ion mode, aiming to reduce computation time while maintaining accuracy. The Barnes and Hut octree method (BHT) [15] is one such alternative, that uses an octree data structure and approximates distant forces, reducing the computation time order to $O(N\log N)$. Multiple adaptations of this method, such as the Fast Multipole (FMM), Lumped Space Charge (LSC), and Zonal Time Step (ZTS) methods, were tested and compared; the results show improvement in speed under certain test conditions [12, 16].

Another popular method, known as particle-in-cell (PIC), traditionally involves particles moving through a gridded structure where the charge of macroparticles (which represent multiple actual particles) are distributed to a background mesh in order to compute electrostatic fields using Poisson's equation and update particle positions using Lorentz forces through integration at each time step [6–9, 17–19]. The traditional PIC methods focus on interactions between particles and fields, capturing inter-particle forces collectively through the charge distributed to the background mesh, rather than calculating direct Coulombic interac-



*The 38th International Electric Propulsion Conference, P. Baudis Convention Center, Toulouse, France, June 23-28, 2024*

tions. Using a single-level uniform grid (i.e. one grid resolution for the entire domain) for the acceleration region is considered inefficient due to varying densities, forcing a choice of fine resolution, increasing computational time, or coarse resolution, reducing accuracy. To tackle this, a multi-level PIC technique featuring higher resolution near the emitter was explored by Wang et al. [6], yielding results in good agreement with experimental data. Although this method is faster than the $n$-body and can practically handle a large number of particles ($N > 10^5$), it could still take several hours to simulate the the acceleration region of an electrospray thruster as noted in Ref. [7]. Narayanan et al. further optimized computational performance by implementing parallelization of processes in PIC, a technique known as ES-PIC [7]. This method is slightly different in how it handles the inflow of particles and performs a shared-memory parallelization of all the fundamental steps of traditional PIC. The reported results show a significant reduction in computation time, with the lowest time being ∼36 minutes using 8 cores for the same test case as examined by Wang et al.

All of the above-listed methods for acceleration region can be applied to the field-free region as the fundamental physics are not drastically different. Like in the later part of the acceleration region, inter-particle Coulomb forces are frequently considered negligible for individual ionic-liquid ion sources, but cannot be considered negligible for an array-scale system, in which case these effects have some impact on plume divergence [20]. The upper bound for the field-free region simulations depends on the goal of the study but usually are on the millimeter scale [12, 13, 21], for which a baseline $n$-body approach would not be feasible or efficient depending on the time step. To get a good time resolution (low time step), other methods such as BHT and their derivatives are more attractive options. If the region of interest is in the centimeter-to-meter scale, PIC seems to be preferred [8, 9, 19, 20, 22].

Asher et al. used their version of PIC as a part of a larger multi-scale model to simulate the neutralization process of a bipolar thruster on the meter scale, for a test case with $13 \times 10^6$ macroparticles simulated over 2100 steps, with different time steps for the three spatial regions. In total, four species of macro particles were injected into this simulation, with each macroparticle representing different quantities of ions for different species, on the order of $10^4$. Smith et al. implemented a shared-memory parallel multi-grid Poisson solver, based on Narayanan et al., into PIC on the Air Force's TURF framework [20, 22]. This method enables the simulation of an entire thruster array with fragmentation effects, by using macroparticles—each representing 100 individual ions—in a reasonably quick (∼22 minutes) time. If approximating the high-level plume shape is sufficient an even faster and simpler method exists, developed by Deng et al., that resolves the overall plume shape in a fraction of a second for a single electrospray [23]. However, this method does not adequately capture the nuances within the plume due to its foundational assumptions. It models the entire plume as one continuous line-of-charge oriented perpendicular to the extractor, characterized by a uniform charge density. This results in a singular electric field at the center line, which prevents accurate prediction of particle behavior in its vicinity. Furthermore, the simplification to a uniform charge density fails to account for dynamic changes in charge distribution over time. While the assumption of constant axial velocity may be valid in specific, localized regions of the plume, these assumptions are not broadly applicable to the charge distribution within electrospray setups, particularly concerning phenomena such as fragmentation.

As it stands, PIC and its variations are among the most efficient techniques for simulating electrospray thruster plumes in the field-free region at large scales, capturing a considerable amount of detail with decent accuracy. However, their computational times—generally ranging from a few minutes to hours on well-equipped computing machines—still present a limitation in a few cases [6, 7, 22]. This is particularly true when simulating a very high number of particles, as this requires substantial memory to track each particle. Additionally, when attempting to solve inverse problems to deduce the emission properties of thrusters based on measurements in the field-free region, these computational times become a significant bottleneck. In these scenarios, the quest for methods that drastically reduce computation times to the order of seconds is essential to enable more comprehensive and efficient evaluations.

In the pursuit of advancing simulation techniques for electrospray thruster plumes in the field-free region, this paper introduces a novel method termed iterative tracer particle (ITP). This approach is capable of capturing essential effects such as fragmentation and surface interactions, as well as properties like plume shape and charge distribution, on par with conventional methods, yet it achieves significantly higher computational efficiency. As a result, the iterative tracer particle method demonstrates a significant potential to reduce simulation times from minutes to mere seconds, thereby enabling comprehensive surveys across various operating conditions. This advancement is particularly pertinent for the inference of thruster emission properties from downstream current measurements, which could revolutionize methods for indirect thrust estimation of electrospray thrusters.

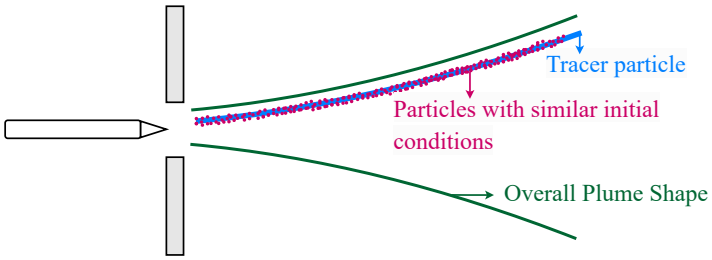


Figure 1: Principle of the iterative tracer particle approach.

## III. Approach

The iterative tracer particle method is a particle-mesh approach, similar to the particle-in-cell (PIC) technique, designed for tracing the steady-state trajectories of charged particles. The effectiveness of this method is grounded in the principle that, in a steady state, particles of a given species emitted with similar initial conditions will follow similar trajectories, forming a particle stream. As depicted in Figure 1, the trajectory of a single tracer particle serves as a reliable approximation for the particle stream. Therefore, instead of simulating each individual particle, simulating a single tracer particle that represents the combined contribution of the particle stream is a viable approach. Leveraging this principle allows for a significant reduction in the number of particles required to be simulated, multiple individual particles down to a single tracer particle, while still achieving comparable plume simulation results. However, it's important to note that due to this principle, this method does not capture the transient phase, as particles originating from the same point may diverge in behavior considerably during this initial phase.

The primary work involved in the iterative tracer particle approach is the determination of the steady-state charge distribution such that the steady-state trajectories of the tracer particles can be simulated. The plume charge distribution is represented as a mesh, similar to the mesh-based representation used in PIC, which is updated after each iteration until it converges to the steady-state distribution. In each iteration, the current guess for the plume charge distribution defines a background electric field, and the trajectory of each tracer particle is propagated according to the background electric field. The trajectory of each tracer particle is then treated as a line of charge, and is distributed to the charge distribution mesh. After every tracer particle has been simulated, the new charge distribution mesh is passed to the next iteration to serve as the new guess for the steady-state distribution. By repeating this process until the background charge distribution converges, the steady-state charge distribution for the plume can be determined.

It is important to note that during each iteration, the electric force acting on each tracer particle is solely determined from the background electric field, and individual tracer particles do not directly interact with each other. Interaction between tracer particles is indirect, through distribution of their charge to the charge distribution mesh and re-calculation of the background electric field at the following iteration. Therefore, the propagation of each tracer particle within an iteration is completely independent of the other tracer particles. Particles can then be simulated sequentially, significantly reducing the memory requirement of the approach as it is not necessary to maintain the tracer particle's trajectory once its charge has been distributed. This particle propagation architecture also naturally suits parallelization, potentially offering additional gains in computational time. However, the most computationally demanding aspect of the current implementation of this method is in the calculation of the electric field.

The overall iterative tracer particle algorithm is detailed in Algorithm 1. The algorithm starts by initializing the charge distribution mesh, defining any external electric fields, and defining the initial conditions for each tracer particle. The tracer particle initial conditions could come from the output of an acceleration region simulation, defining the distributions of particle species, position, and velocity when entering the field-free region. In each iteration, the charge distribution mesh is used to determine the internal electric field due to inter-particle Coulomb forces before being reset. The internal field is added to the external field to define the total electric field that will act on the tracer particles. Each tracer particle is then propagated through the domain using a fourth-order Runge-Kutta (RK4) method, with the dynamics of the particle governed by the equation of motion derived from Lorentz force, where the self-induced magnetic field is

**Algorithm 1** Iterative Tracer Particle

```
Initialize charge mesh
Initialize external electric field
Initialize tracer particles
while number of iterations is less than the maximum number of iterations do
    Calculate internal electric field from charge mesh
    Calculate total electric field from internal and external fields
    Reset charge mesh
    for each tracer particle do
        Propagate tracer particle trajectory
        Define line of charge from tracer particle trajectory
        Distribute charge to charge mesh
    if steady-state criterion is satisfied then
        Exit while loop
return steady-state charge distribution
```

considered to be negligible. The equations of motion are therefore

$$m\frac{\mathrm{d}^2\boldsymbol{r}}{\mathrm{d}t^2} = q\boldsymbol{E}(\boldsymbol{r}) \tag{1}$$

where $\boldsymbol{r}$ is the particle position, $m$ is the particle mass, $q$ is the particle charge, and $\boldsymbol{E}(\boldsymbol{r})$ is the local electric field. Since the electric field is defined over a discrete mesh, the components of $\boldsymbol{E}(\boldsymbol{r})$ are approximated using bilinear interpolation [24].

After each tracer particle has been simulated through the domain, its charge is distributed to the charge mesh using the line-of-charge approximation. While ideally a continuous line of charge would be used to update the charge mesh, computational efficiency necessitates using the midpoint between consecutive time steps along the tracer particle trajectory as a pragmatic approximation for preliminary analyses. This method performs bilinear assignment, the inverse of bilinear interpolation, to assign the total charge along each segment of the tracer particle's trajectory to the neighboring charge mesh grid points. Once every tracer particle has been simulated, and charge distributed to the charge mesh, the algorithm checks two exit criteria. The first is if the number of iterations has reached a user-defined maximum number of iterations. The second is based on the change in the charge distribution; once the change in the charge distribution between iterations falls below a user-defined tolerance, the algorithm returns the charge distribution mesh as the steady-state charge distribution. Alternative convergence criteria could be used such as the change in the tracer particle trajectories between iterations.

### A. Fragmentation

Many electrospray thrusters use ionic-liquids as their propellant. For an ionic liquid composed of a cation $C^+$ and anion $A^-$, the charged particles emitted from an ionic-liquid ion source, which form the basis for ionic-liquid electrospray thrusters, are assumed to come from a discrete set of possible particles defined by $(C^+A^-)_nC^+$ in the positive mode and $(C^+A^-)_nA^-$ in the negative mode, where $n$ is a non-negative integer. Particles are typically classified by $n$, their degree of solvation, with the most common species being monomers ($n = 0$) and dimers ($n = 1$) for an ionic-liquid ion source operating in the ion mode [25]. Over time, ion clusters ($n > 0$) may fragment, typically breaking apart into a neutral particle, $(C^+A^-)$, and an ion cluster with degree of solvation $n' = n - 1$ [26].

Fragmentation could occur in the acceleration or field-free regions, but this work is only concerned with particle fragmentation in the field-free region. The rates of dimer-to-monomer fragmentation in the field-free region have been experimentally characterized for a variety of ionic liquids, and are consistent with an constant rate process model [26, 27]. As a simple baseline implementation to demonstrate the incorporation of fragmentation into the iterative tracer particle method, this work assumes that the electrospray thruster emits a beam composed of 60% monomers and 40% dimers by current fraction, in line with experimental measurements of an ionic-liquid ion source beam composition [28]. It is assumed that there is no fragmentation in the acceleration-region such that the emitted beam is mono-energetic. All tracer particles associated



*The 38th International Electric Propulsion Conference, P. Baudis Convention Center, Toulouse, France, June 23-28, 2024*

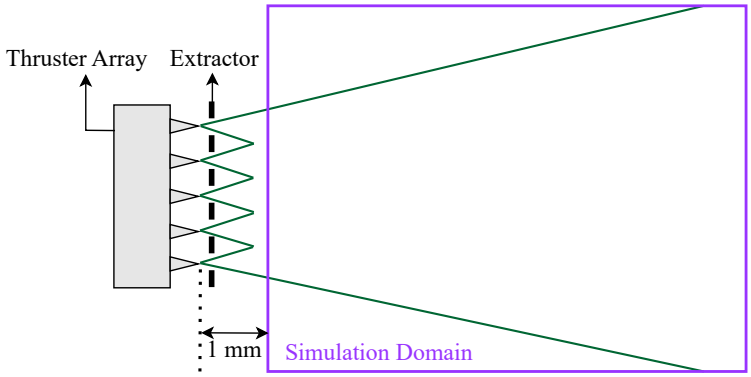


Figure 2: Notional diagram of the simulation domain.

with dimers are assigned a lifetime according to a constant rate equation [26]

$$t = -\tau \ln(1 - p) \tag{2}$$

where $\tau$ is the species mean lifetime and $p$ is a uniformly-distributed random number.

During each iteration, tracer particles associated with dimers are assigned a random lifetime according to Eq. 2 and then simulated, as dimers, up to their lifetime. Once the dimer lifetime is reached, the original tracer particle is replaced by two new tracer particles to represent the monomer and neutral fragmentation products. Both of the new tracer particles assume the position and velocity of the dimer tracer particle, with the only modification to the particle parameters being the particle masses and lack of charge for the neutral tracer particle. Both new tracer particles are simulated as any other tracer particle. However, only the trajectory of the monomer tracer particle is of interest since the neutral tracer particle will follow a straight-line trajectory and does not contribute to the charge distribution.

## IV. Implementation

For validating the simulation results of the ITP method, an $n$-body simulation algorithm was developed and implemented. This $n$-body algorithm also employs the RK4 method for particle propagation, with the dynamics driven by the direct interaction between the charged species in the beam

$$m_p \frac{\mathrm{d}^2 \boldsymbol{r}_p}{\mathrm{d}t^2} = \sum_{i \neq p}^{n} \frac{q_p q_i}{4\pi\epsilon_0} \frac{\boldsymbol{r}_p - \boldsymbol{r}_i}{\|\boldsymbol{r}_p - \boldsymbol{r}_i\|^3} \tag{3}$$

where $\boldsymbol{r}$ is the particle position, $m$ is the particle mass, and $q$ is the particle charge. Subscript $p$ represents the particle under consideration while subscript $i$ represents all other particles in the domain that contribute towards the force acting on particle $p$. The simulations are executed in MATLAB, with the dynamics calculations of the $n$-body algorithm being parallelized through MATLAB's Parallel Computing Toolbox, specifically using the Threads Parallel Environment setting. Likewise, the $\boldsymbol{E}$-calculation phase of the iterative tracer particle algorithm is also parallelized. The computations were run on a computer which was configured with an AMD Ryzen 7 5800H processor (3.2 GHz), 8 CPU cores, and 16 GB of RAM.

The test case presented in this paper is a two-dimensional array-level simulation, using simplified initial conditions, characterized by a collimated beam of uniformly-spaced particles. The collimated and uniformly-spaced beam are idealized representations of the beam emitted by an electrospray thruster rather than accurate depictions of the complexities observed in experiments or high-fidelity simulations (e.g., Ref. [10]). This approach was selected to streamline the validation processes with $n$-body simulations, acknowledging the limitation due to the absence of realistic distribution data as well as low particle counts. Realistic simulations will be presented in future works.

Both the $n$-body and iterative tracer particle simulations were conducted using identical initial conditions and parameters, as detailed in Table 1. The simulation setup is depicted in Figure 2, where the simulated plume is of a thruster array, in the field-free region, starting at approximately 1 mm from the emitter tip. The simulation domain is a 6 cm square with axis ranges of $x \in [0, 6]$ cm and $y \in [-3, 3]$ cm. The simulations were

Table 1: Simulation parameters comparison case.

| Parameter | Value |
|---|---|
| Domain side length | 6 cm |
| Time step | 5 ns |
| Total simulation time ($n$-body) | 5500 ns |
| Total emitted particle count ($n$-body) | 55000 |
| Peak particle count ($n$-body) | 15010 |
| Number of tracer particles | 50 |
| Number of grid cells along $x$-axis | 80 |
| Number of grid cells along $y$-axis | 160 |
| Grid cell resolution d$x$ | 750 $\mu$m |
| Grid cell resolution d$y$ | 375 $\mu$m |

| Emitter Design Parameter | Value |
|---|---|
| Array length | 1 cm |
| Array acceleration potential | 1 kV |
| Array emitter current | 200 $\mu$A |
| Monomer current fraction | 1.0 |
| Monomer mass | 111.2 amu |
| Monomer initial speed | 41.507 km/s |

carried out with a time step of 5 ns, over a duration of 5.5 $\mu$s. The requirement for a large simulation time to reach steady-state constrained the choice of time step given the available resources. At each time step, 50 particles are added to the domain with $x = 0$ and uniformly distributed over a range of $y \in [-0.5, 0.5]$ cm to represent a small electrospray thruster. All particles in the simulation represent monomers of EMI-$BF_4$ propellant, operating in positive pure-ion mode. The charge of each particle in both the $n$-body and iterative tracer particle simulations is calculated according to

$$q = \frac{I\Delta t}{n_{\Delta t}} \tag{4}$$

where $I$ is the total array current, $\Delta t$ is the simulation time step, and $n_{\Delta t}$ is the number of particles added to the domain in each time step. The initial speed of each particle is calculated according to

$$v = \sqrt{2(q/m)\phi} \tag{5}$$

where $(q/m)$ is the charge-to-mass ratio of the species and $\phi$ is the thruster acceleration potential. Since the emitted beam is assumed to be collimated, the velocity direction for all particles is assumed to be the positive $x$ direction such that $\boldsymbol{v} = v\hat{\boldsymbol{x}}$.

In the $n$-body simulation, 50 particles are released at each time step ($n_{\Delta t} = 50$) to emulate a continuous flow. This approach results in a total of 55,000 emitted particles, and peak count of 15010 particles in the domain in a given time step. On the other hand, only 50 particles are simulated during each iteration of the iterative tracer particle approach. The grid mesh resolution, a critical component of the iterative tracer particle method, selected for this comparison consists of 80 cells along the $x$-axis and 160 cells along the $y$-axis ($80 \times 160$). The rationale behind this selection is addressed in the subsequent section. The convergence criterion for the iterative tracer particle approach is based on the maximum change in charge value at any grid point across the entire charge mesh. For the test cases in this work, the charge mesh is assumed to be near steady state when the maximum change in charge between iterations is less than 1/10 of the charge of an individual tracer particle.

## V. Validation

This section delves into the comparison and analysis of the simulation results. The charge distribution at the final time step of the $n$-body simulation is shown in Figure 3. Given that the iterative tracer particle simulation inherently yields the plume in steady state, it is imperative to compare it with the $n$-body simulation only when the latter has also attained a steady-state condition. Every charge distribution plot presented in this section, such as the one shown in Figure 3, was calculated with a mesh resolution of 30 cells along the $x$-axis and 60 cells along the $y$-axis ($30 \times 60$).

For the $n$-body simulation, steady-state conditions are identified by examining emitted particles. Particles emitted at each time step are considered a discrete set. The determination of steady state relies on the stabilization of two metrics: the total residence time of particles in the simulation domain and the maximum

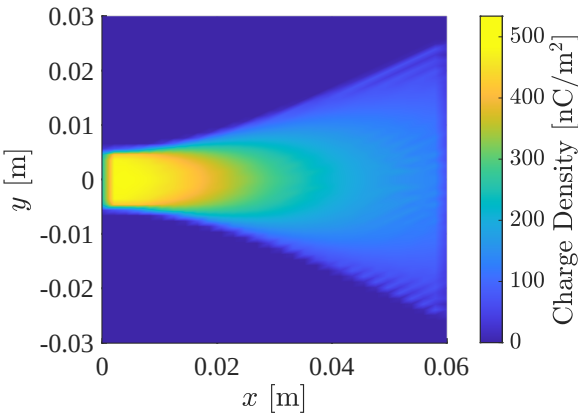


Figure 3: Charge distribution at the final time step of the $n$-body simulation.

change in charge across the simulation domain. The total residence time is assessed by tracking the amount of time before all of the particles in each set have left the simulation domain. The maximum change in charge is evaluated by first calculating the charge distribution of every discrete particle set separately. This calculation is similar to distributing the charge of tracer particles in the iterative tracer particle approach, using bilinear assignment. Here, the complete progression (time history) of particles in the set within the simulation domain is accounted for and distributed to a $(80 \times 160)$ grid, the same grid used in the iterative tracer particle approach. Subsequently, the maximum change in charge values across all grid points between consecutive particle sets is computed. This method of comparing the plume of particle sets, rather than the plume at a given time step, enables direct comparison with the iterative tracer particle simulation, which outputs a single particle set, fundamentally different from the time snapshot that the $n$-body simulation naturally produces.

Steady-state is considered to be achieved when the maximum residence time for particles across sets becomes consistent and the maximum change in charge between sets diminishes below a user-defined threshold, as shown in Figure 4. In Figure 4 the maximum change in charge has been normalized by the particle charge ($2\times10^{-14}$ C). Particle sets 229–793 have identical maximum residence times, indicating the potential for steady-state conditions starting from the 229$^{\text{th}}$ time step. The plot of maximum change in charge supports the claim of the simulation approaching steady state as it shows a continuous downward trend. The spikes in the maximum change in charge are numerical artifacts due to slight changes in the positions of particles at the edge of the simulation domain, causing them to flip between being inside or outside the simulation domain between different sets. The change in charge between sets stabilizes at the 793$^{\text{rd}}$ set, and so the charge distribution of the 793$^{\text{rd}}$ particle set is assumed to represent the steady-state distribution. For particle sets beyond 793, there is not enough time for the particles to fully traverse the simulation domain due to the pre-defined simulation duration, which creates the downward trend in residence time in Figure 4.

The convergence criterion for the iterative tracer particle approach is that the maximum change in charge across the charge mesh between iterations falls below 1/10 of the charge of a tracer particle. The progression of this maximum charge change metric is depicted in Figure 5, where the $y$-axis is normalized by the particle charge. The maximum residence time for tracer particles in the final iteration of the iterative tracer particle approach is 1.535 $\mu$s, which matches that of the steady-state $n$-body sets. To further compare the iterative tracer particle solution and the steady-state $n$-body solution, Euclidean distances are calculated between corresponding $n$-body particles and tracer particles at each time step, where particles exist in both solutions. For each particle in a set, these distances are first averaged across all time steps, yielding a mean Euclidean distance for each individual particle. These mean distances for all particles in the set are then aggregated and normalized by the initial particle count, resulting in the mean Euclidean distance per particle per set. The results of these calculations depicted in Figure 5, where the values are shown to be consistently minimal, maintained at the micrometer level for the steady-state sets, with clear stabilization evident beyond set 400. This consistent minimal variation among these sets suggests convergence in system dynamics, thereby affirming the attainment of steady-state conditions in the $n$-body sets selected for comparison.

The 2D charge distribution at the final iteration of the iterative tracer particle simulation is shown in

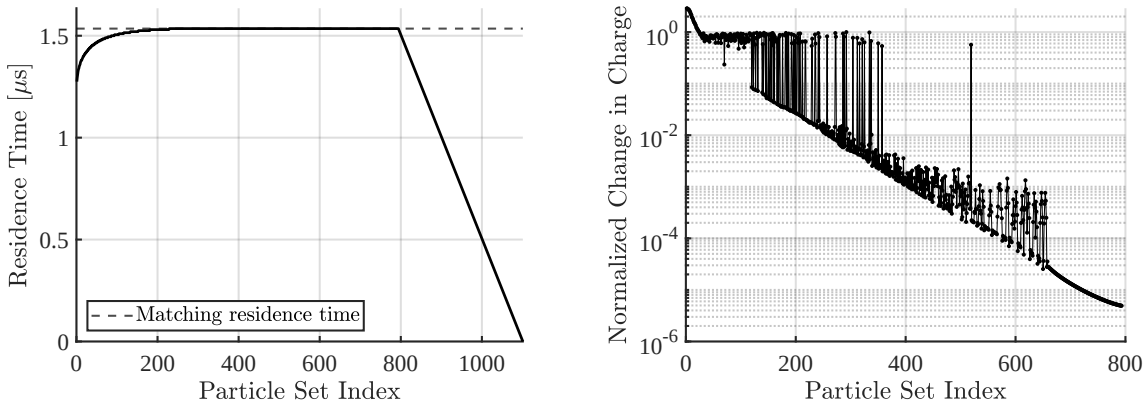


Figure 4: Maximum particle residence time versus particle set (left) and maximum change in charge distribution versus simulation time (right) for the $n$-body simulation.

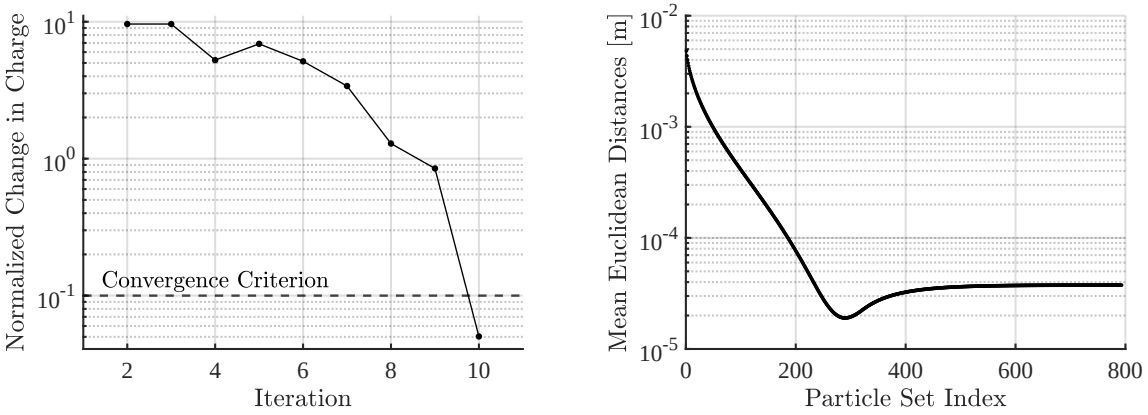


Figure 5: Maximum change in charge between iterations of the iterative tracer particle approach (left) and normalized distance error between the iterative tracer particle final solution and different particle sets from the n-body approach (right).

Figure 6, with an overlay of the trajectories of the outer-most tracer particles. Direct comparison between the $n$-body and iterative tracer particle results is also shown in Figure 6, by plotting the difference in the 2D charge distributions. The primary differences between the two solutions are at the right end of the domain, where tracer particles from the iterative tracer particle simulation tend to exit the domain slightly earlier than particles from the $n$-body simulation, and at the outer edge of the plume. Resolving these differences will be considered in future work, but is likely due to the approximate nature of the electric field for the iterative tracer particle approach due to the discrete mesh. However, it is worth noting that the charge distribution for the inner region of the plume show consistent behavior.

To better understand the observed discrepancies, Figure 7 presents the root mean square error (RMSE) analysis for the $x$-axis, $y$-axis, and Euclidean distance, comparing the final results of the iterative tracer particle and $n$-body approaches. Each time step of every particle in the n-body data is compared to its corresponding point in the iterative tracer particle data, considering only the time steps where particles exist in both simulations. Here, the particle indices correspond to their initial $y$-axis position, starting with index 1 at the top, $y(0) = 0.5$ cm, and progressing to index 50 at the bottom, $y(0) = -0.5$ cm. A lower RMSE value signifies a closer match between the two trajectories, indicating higher accuracy in the iterative tracer particle simulation. The RMSE plot highlights that particles at the edges of the plume are the most mismatched, and those at the center of the plume also exhibit slightly higher errors than the surrounding particles. The overall RMSE values are 42.8829 $\mu$m in the $x$-axis, 23.3241 $\mu$m in the $y$-axis, and 48.8155 $\mu$m

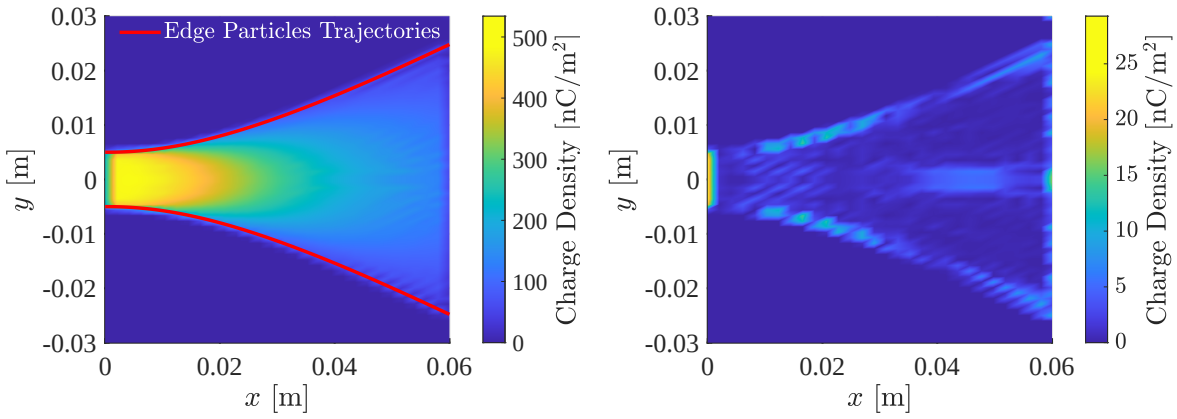


Figure 6: Charge distribution from the iterative tracer particle final solution (left) and difference in charge distribution between the n-body approach and iterative tracer particle approach (right).

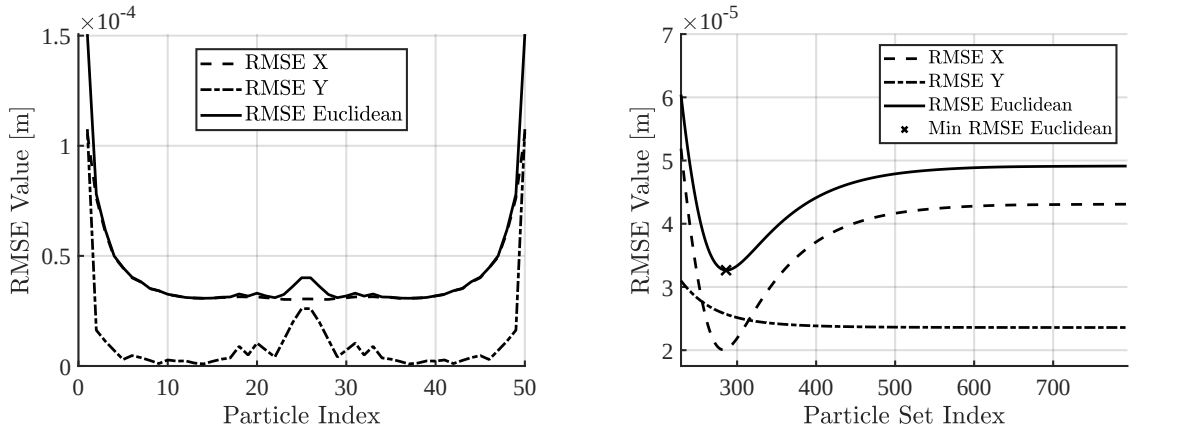


Figure 7: Root mean square error between the final solutions of the iterative tracer particle and $n$-body approaches (left) and root mean square error for different $n$-body particle sets versus the final solution of the iterative tracer particle approach (right).

for the Euclidean distance, which are low in the micrometer range compared to the 6 cm simulation domain, indicating good agreement. The evolution of RMSE between the last iteration of the iterative tracer particle and all $n$-body sets beyond the 229$^{\text{th}}$ set is plotted in the right part of Figure 7. This plot reveals a higher error in the $x$-direction (along the thruster axis), with the trends stabilizing beyond the 500$^{\text{th}}$ time step.

Further validation of the simulations' alignment involves an in-depth cross-sectional charge distribution analysis at various $x$ locations. To eliminate numerical noise, the charge distributions are evaluated over a lower-resolution mesh ($30 \times 60$) relative to the mesh used during the iterative tracer particle simulations. Figures 8, and 9 show a comparison of the charge distribution cross sections at downstream locations of $x = 3$ cm and $x = 5.5$ cm, respectively. The plots demonstrate a general agreement between the two simulations, further affirming the iterative tracer particle technique's validity against the $n$-body standard. Due to the unrealistically low number of particles simulated, these charge distribution plots may seem impractical. Therefore, a more realistic simulation using the iterative tracer particle method was conducted with 1000 initial particles and a mesh resolution of $50 \times 100$, completing in just 109 seconds. The charge distributions at each of these locations from the new simulation are also shown in Figures 8, and 9. The results with 1000 tracer particles reveal some impractical spikes in the charge distribution at the edge of the plume. Consequently, the mesh resolution was increased to $150 \times 300$, and the simulation was conducted again with the remaining parameters unchanged. The results plotted in Figure 10 show a more realistic distribution. This simulation took 523 seconds to complete, a 4.8× increase in computation time due to a 9× increase

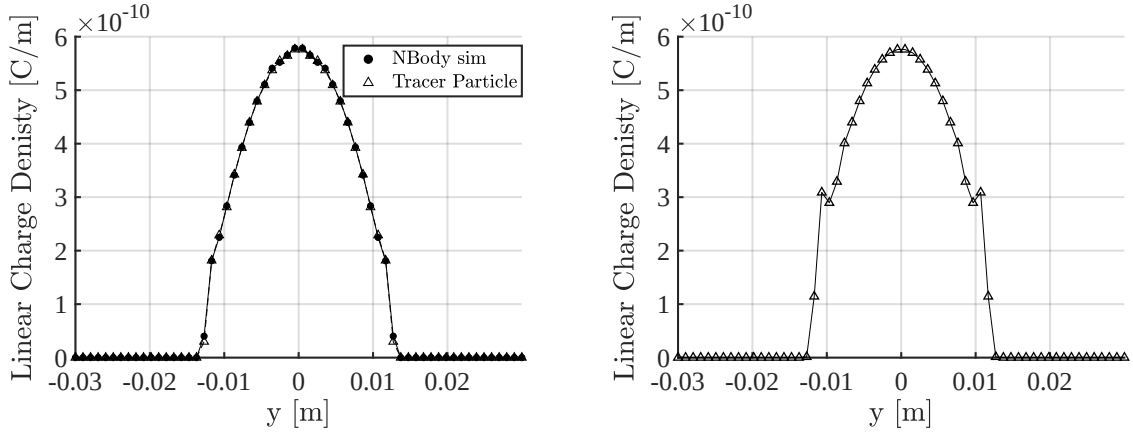


Figure 8: Comparison of charge distribution from both the iterative tracer particle and $n$-body solutions at 3 cm downstream of the thruster for 50 tracer particles (left) and iterative tracer particle solution at the same downstream location for 1000 tracer particles (right).

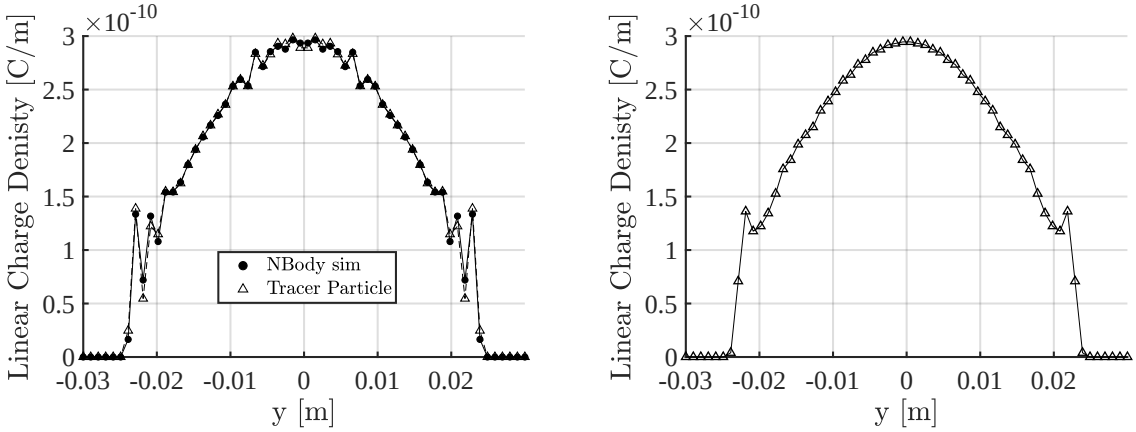


Figure 9: Comparison of charge distribution from both the iterative tracer particle and $n$-body solutions at 5.5 cm downstream of the thruster for 50 tracer particles (left) and iterative tracer particle solution at the same downstream location for 1000 tracer particles (right).

in the charge distribution mesh grid points. The optimal resolution, in terms of providing accurate results within a reasonable time frame, could lie between the two tested parameters presented here.

This comprehensive analysis, encompassing particle time progression, RMSE, and charge distribution comparisons, demonstrates the iterative tracer particle simulation's alignment with the $n$-body simulation outputs. The evidence from these comparison metrics supports the validation of the iterative tracer particle simulation technique against the $n$-body method. Notably, the iterative tracer particle simulation demonstrates a significant advantage in computational efficiency—completing in 30.09 seconds for the 50 tracer particles test case compared to the 10.31 hours required by the $n$-body simulation—while producing a comparable result. It also has superior memory efficiency, using just $\sim$3.5 MB while the n-body simulation needed $\sim$12.3 GB.

With the iterative tracer particle method being validated, the incorporation of particle fragmentation was also tested with the test case parameters listed in Table 2 and the implementation logic described in Section III A. The resulting plume's 2D charge distribution is plotted in Figure 11. It can be seen that monomers are dispersed more widely from the central axis of the beam compared to dimers. This observation aligns with theoretical expectations, as monomers will experience a larger acceleration than dimers from inter-particle Coulomb forces due to their larger charge-to-mass ratio. This simulation took 169 seconds to complete.

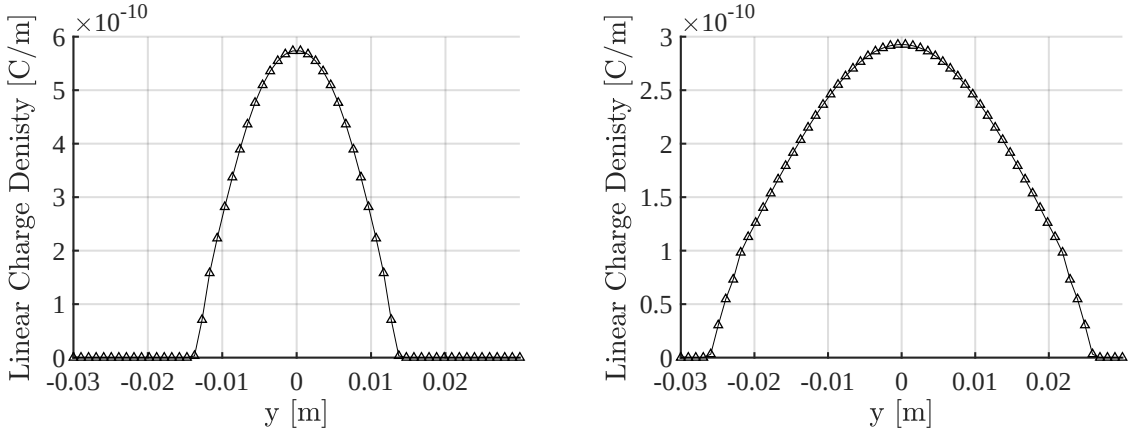


Figure 10: Charge distribution at 3 cm (left) and 5.5 cm (right) downstream of the thruster using an iterative tracer particle approach with 1000 tracer particles and high-resolution charge mesh.

Table 2: Simulation parameters for fragmentation test case.

| Parameter | Value |
|---|---|
| Domain side length | 6 cm |
| Time step | 5 ns |
| Number of tracer particles | 1000 |
| Number of grid cells along $x$-axis | 50 |
| Number of grid cells along $y$-axis | 100 |
| Grid cell resolution d$x$ | 1200 $\mu$m |
| Grid cell resolution d$y$ | 600 $\mu$m |

| Emitter Design Parameter | Value |
|---|---|
| Array length | 1 cm |
| Array acceleration potential | 1 kV |
| Array emitter current | 200 $\mu A$ |
| Monomer current fraction | 0.6 |
| Dimer current fraction | 0.4 |
| Monomer mass | 111.2 amu |
| Dimer mass | 309.2 amu |
| Monomer initial speed | 41.507 km/s |
| Dimer initial speed | 29.347 km/s |

### A. Impact of Mesh Resolution

The grid mesh resolution for the test case presented earlier was selected to satisfy the Courant–Friedrichs–Lewy (CFL) condition, defined by

$$v_{\max}\Delta t \leq L \tag{6}$$

where $v_{\max}$ is the maximum speed of a particle, $\Delta t$ is the simulation time step, and $L$ is the length of each grid cell. This condition aids in determining the minimum cell length or the maximum number of grid points for a given domain size such that no particle traverses more than one grid cell in a single time step. In the test case under consideration, the upper limit of $L$, determined by this condition, is $2.0753 \times 10^{-4}$ m which translates to an upper limit of 289 cells in the $x$ direction. The selected mesh resolution in the $x$ direction, 80 cells, is ∼70% below the limit. Given that the particles are initialized with a $y$ velocity component of zero, the CFL condition cannot be used to provide an upper bound for the $y$ direction prior to initializing the mesh resolution. It is expected that even as the $y$ velocity component builds up, it will remain considerably lower than the $x$ velocity, resulting in relatively small displacements in the $y$ direction. To accurately capture these small $y$ movements, a finer resolution is generally more effective. Thus, based on an educated guess informed by the simulation domain size, the $y$ resolution was set at double the $x$ resolution, with 160 cells. This value turned out to be ∼75% below the CFL limit of 625 cells, based on the maximum $y$ velocity attained by particles during the simulation.

This $x$ direction resolution selection, being ∼70% below the limit, is based on findings from multiple test simulations where the computational time was the shortest for this resolution while matching total residence time with that of $n$-body. It is important to note that this resolution selection strategy may not be universally applicable. There are several other factors apart from the CFL condition that influence the

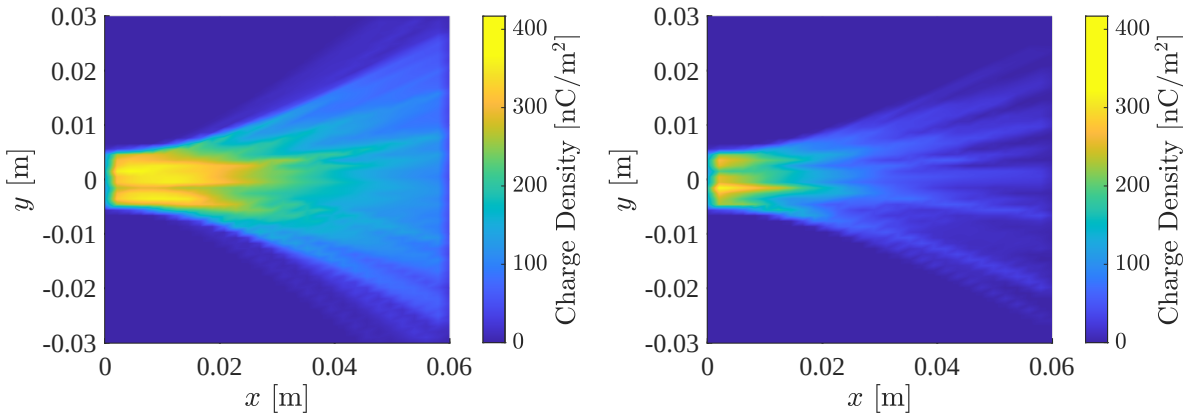


Figure 11: Charge distribution of monomers (left) and dimers (right) from the final iteration of an iterative tracer particle approach.

accuracy of the simulation. Therefore, a more detailed and systematic process for choosing the appropriate mesh resolution for a given set of simulation parameters is necessary. This is a topic of ongoing research and will be addressed in future work. The general trend observed indicates that for a given set of initial conditions and simulation parameters, increasing the mesh resolution reduces the mismatch error between the $n$-body and iterative tracer particle simulations up to a certain resolution, beyond which the error starts to increase again. Another observed trend is that, with an increase in the initial number of particles for the same set of initial conditions or simulation parameters, the optimal match—or the least error—between the iterative tracer particle simulation and its $n$-body reference occurs at higher resolutions. This indicates that an increase in the number of particles necessitates an increase in mesh resolution. Additionally, the choice of mesh resolution for a given domain size is influenced by the simulation time step. Ultimately, the choice of mesh resolution hinges on a balance of computational speed and accuracy, representing a critical trade-off.

## VI. Conclusion

A computationally-efficient approach for simulating the plume of an electrospray thruster is essential for solving inverse problems such as inferring thruster emission properties from measurements taken in the field-free region. The ability to solve these problems would enhance the understanding of thruster operation and enables the refinement of electrospray thruster design based on experimental feedback. The iterative tracer particle approach introduced in this work is aimed at doing exactly that, potentially enabling computationally-simple simulation of electrospray plumes in the field-free region while maintaining decent accuracy and capturing important phenomena such as ion cluster fragmentation. The iterative tracer particle method relies on the fact that in steady-state operation, particles emitted from the thruster with similar initial conditions will follow similar trajectories. Hence, a set of particles with similar initial conditions is approximated as one tracer particle and propagated using a particle-mesh technique.

This method has been validated against the $n$-body approach for a simple 2D array-level test case. The simulation results, compared using 2D charge distribution difference, charge distribution overlay, and root mean square analysis, show a good agreement between the two methods. The small discrepancies, mainly on the outer edge of the plume, could be attributed primarily to the chosen mesh resolution. The tuning of this mesh resolution, for a given set of parameters, will be analyzed in future work. These simulations, when run on a personal computer, took 10.31 hours for the $n$-body method (parallelized) and 30.09 seconds for the iterative tracer particle approach, which is faster by a factor of 1233. The memory utilization during the simulation is also significantly lower for this method, at $\sim$3.5 MB compared to $\sim$12.3 GB for the $n$-body simulation. This computational efficiency factor would become even more pronounced in a realistic simulation with a significantly larger number of particles. The computational efficiency of the iterative tracer particle approach could also be improved through the adoption of more efficient sub-processes such as the use of a multigrid Poisson solver for calculating the electric field rather than the current brute-force calculation of the electric field given the charge distribution mesh at each iteration.

## References


[1] Natisin, M., Zamora, H., McGehee, W., Arnold, N., Holley, Z., Holmes, M., and Eckhardt, D., "Fabrication and Characterization of a Fully Conventionally Machined, High-Performance Porous-Media Electrospray Thruster," *Journal of Micromechanics and Microengineering*, Vol. 30, No. 11, 2020, p. 115021. https://doi.org/10.1088/1361-6439/abb8c3.

[2] Krejci, D., Mier-Hicks, F., Thomas, R., Haag, T., and Lozano, P., "Emission Characteristics of Passively Fed Electrospray Microthrusters with Propellant Reservoirs," *Journal of Spacecraft and Rockets*, Vol. 54, No. 2, 2017, pp. 447–458. https://doi.org/10.2514/1.A33531.

[3] Cisquella-Serra, A., Galobardes-Esteban, M., and Gamero-Castaño, M., "Scalable Microfabrication of Multi-Emitter Arrays in Silicon for a Compact Microfluidic Electrospray Propulsion System," *ACS Applied Materials & Interfaces*, Vol. 14, No. 38, 2022, pp. 43527–43537. https://doi.org/10.1021/acsami.2c12716.

[4] Schroeder, M. R., Gallud, X., Bruno, A. R., Jia-Richards, O., and Lozano, P. C., "Angular Properties of Ionic-Liquid Electrospray Emitters," *AIAA SciTech Forum*, American Institute of Aeronautics and Astronautics, National Harbor, MD, USA, 2023. https://doi.org/10.2514/6.2023-1408.

[5] Gañán-Calvo, A., Lasheras, J., Dávila, J., and Barrero, A., "The Electrostatic Spray Emitted from an Electrified Conical Meniscus," *Journal of Aerosol Science*, Vol. 25, No. 6, 1994, pp. 1121–1142. https://doi.org/10.1016/0021-8502(94)90205-4.

[6] Wang, P., Borner, A., Korkut, B., Li, Z., and Levin, D. A., "Simulations of Electrospray in a Colloid Thruster with a High Resolution Three-Dimensional Particle-in-Cell Model," *44th AIAA Plasmadynamics and Lasers Conference*, American Institute of Aeronautics and Astronautics, 2013, p. 2629. https://doi.org/10.2514/6.2013-2629.

[7] Narayanan, R. K. and Madduri, K., "Parallel Particle-in-Cell Performance Optimization: A Case Study of Electrospray Simulation," *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, IEEE, 2017, pp. 1158–1167. https://doi.org/10.1109/IPDPSW.2017.160.

[8] Asher, J., Huang, Z., Cui, C., and Wang, J., "Multi-Scale Modeling of Ionic Electrospray Emission," *Journal of Applied Physics*, Vol. 131, No. 1, 2022, p. 014902. https://doi.org/10.1063/5.0071483.

[9] Asher, J. S. and Wang, J., "Three-Dimensional Particle-in-Cell Simulations of Bipolar Ionic Electrospray Thruster Plume," *Journal of Propulsion and Power*, Vol. 38, No. 4, 2022, pp. 573–580. https://doi.org/10.2514/1.B38610.

[10] Petro, E. M., Gallud, X., Hampl, S. K., Schroeder, M., Geiger, C., and Lozano, P. C., "Multiscale Modeling of Electrospray Ion Emission," *Journal of Applied Physics*, Vol. 131, 2022, p. 193301. https://doi.org/10.1063/5.0065615.

[11] Gallud, X. and Lozano, P. C., "The Emission Properties, Structre and Stability of Ionic Liquid Menisci Undergoing Electrically Assisted Ion Evaporation," *Journal of Fluid Mechanics*, Vol. 933, 2021, p. A43. https://doi.org/10.1017/jfm.2021.988.

[12] Hampl, S. K., Waggoner, M. T., Gallud Cidoncha, X., Petro, E. M., and Lozano, P. C., "Comparison of Computational Algorithms for Simulating an Electrospray Plume with a N-Body Approach," *Journal of Electric Propulsion*, Vol. 1, No. 1, 2022, p. 17. https://doi.org/10.1007/s44205-022-00015-w.

[13] Breddan, M. J., Curry, D. R., Sharma, M., Richmond, M. O., Collins, A. L., Brieda, L., and Wirz, R. E., "Electrospray Plume Modeling: Study on Drag Influence," *AIAA Science and Technology Forum and Exposition*, 2022, p. 1358. https://doi.org/10.2514/6.2022-1358.

[14] Zhao, Y. and Wang, J. J., "A Particle-Particle Simulation Model for Droplet Acceleration in Colloid Thrusters," *36th International Electric Propulsion Conference*, Electric Rocket Propulsion Society, University of Vienna, Austria, 2019.

[15] Barnes, J. and Hut, P., "A Hierarchical O (N log N) Force-Calculation Algorithm," *Nature*, Vol. 324, No. 6096, 1986, pp. 446–449. https://doi.org/10.1038/324446a0.

[16] Grifoll, J. and Rosell-Llompart, J., "Efficient Lagrangian Simulation of Electrospray Droplets Dynamics," *Journal of aerosol science*, Vol. 47, 2012, pp. 78–93. https://doi.org/10.1016/j.jaerosci.2012.01.001.

[17] Tskhakaya, D., Matyash, K., Schneider, R., and Taccogna, F., "The Particle-in-Cell Method," *Contributions to Plasma Physics*, Vol. 47, No. 8-9, 2007, pp. 563–594. https://doi.org/10.1002/ctpp.200710072.

[18] Kumar, R., Bauner, A., Li, Z., and Levin, D., "Electrospray Simulation in a Colloid Thruster Using Particle-in-Cell (PIC) Approach," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, American Institute of Aeronautics and Astronautics, 2012, p. 788. https://doi.org/10.2514/6.2012-788.

[19] Asher, J., Acarregui, O., and Wang, J., "Numerical Simulation of Ionic Electrospray Contamination for Small Satellite Formation Flight," *IEEE Transactions on Plasma Science*, Vol. 51, No. 9, 2023, pp. 2508–2514. https://doi.org/10.1109/TPS.2023.3295400.

[20] Smith, A. G. and Petro, E. M., "Propagating an Electrospray Emission Model to Array Scales Using Particle-in-Cell," *Journal of Propulsion and Power*, 2024, pp. 1–14. https://doi.org/10.2514/1.B39400.

[21] Petro, E. M., Miller, C. E., Schmidt, J., and Lozano, P. C., "Development of an Electrospray Fragmentation Model for Kinetic Plume Modeling," *36th International Electric Propulsion Conference*, Electric Rocket Propulsion Society, University of Vienna, Austria, 2019.

[22] Smith, A. G. and Petro, E. M., "Array-Scale Modeling of Electrospray Ion Plumes within AFRL Plume Simulation Tool TURF," *AIAA Science and Technology Forum and Exposition*, 2023, p. 63. https://doi.org/10.2514/6.2023-0063.

[23] Deng, W. and Gomez, A., "Influence of Space Charge on the Scale-Up of Multiplexed Electrosprays," *Journal of Aerosol Science*, Vol. 38, No. 10, 2007, pp. 1062–1078. https://doi.org/10.1016/j.jaerosci.2007.08.005.

[24] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes 3rd Edition: The Art of Scientific Computing*, Cambridge University Press, 2007.

[25] Lozano, P. and Martínez-Sánchez, M., "Ionic Liquid Ion Sources: Characterization of Externally Wetted Emitters," *Journal of Colloid and Interface Science*, Vol. 282, No. 2, 2005, pp. 415–421. https://doi.org/10.1016/j.jcis.2004.08.132.

[26] Miller, C. E. and Lozano, P. C., "Measurement of the Dissociation Rates of Ion Clusters in Ionic Liquid Ion Sources," *Applied Physics Letters*, Vol. 116, No. 25, 2020, p. 254101. https://doi.org/10.1063/5.0006529.

[27] Miller, C. E., "Characterization of Ion Cluster Fragmentation in Ionic Liquid Ion Sources," Ph.D. thesis, Massachusetts Institute of Technology, 2019.

[28] Lozano, P. C., "Energy Properties of an EMI-Im Ionic Liquid Ion Source," *Journal of Physics D: Applied Physics*, Vol. 39, 2006, pp. 126–134. https://doi.org/10.1088/0022-3727/39/1/020.