Proceedings of the ASME 2024
International Design Engineering Technical Conferences
and Computers and Information in Engineering Conference
IDETC/CIE 2024
August 25-28, 2024, Washington, DC

DETC2024-142145

PERFORMANCE ANALYSIS OF AUTOMATASCALES TO SUPPORT EARLY DESIGN DECISIONS

Pongchalat Chaisiriroj, Robert B. Stone
Design Engineering Lab
School of Mechanical, Industrial and Manufacturing Engineering
Oregon State University
Corvallis, OR, USA

ABSTRACT

In this paper, we present a comprehensive study and performance analysis on the AutomataScales simulations method focusing on electric propulsion systems for deep space missions. These applications require precise and time efficient simulations. However, traditional simulation methods such as Particle-In-Cell (PIC) method facing challenges from computationally intensive (2.5-21 days), memory demands (random-access memory or RAM and CPU), and steep learning curve for researchers. These limitations reduce their effectiveness in resource-constrained environments. For instance, each GB of RAM consumes approximately 0.1875 watts which resulting in more power consumption ranging from 87.1 to 145.2 MW per simulation run. The AutomataScales method combines discretization techniques with cellular automata and a multi-layer, multi-resolution approaches. This method offers a powerful tool to model complex multiphysics interactions and utilizing hybrid numerical scheme (discrete and continuous) with lower computational time and memory usage. The method depicts intricate and accurate behaviors in various types of particle trajectory (ionized particles, primary and secondary electrons) and plasma physics (particle collision and ionization). It provides a scalable and adaptable framework for multiphysics simulations with almost real-time simulation (0.1 second per time step). A key aspect of our research is the computational efficiency of AutomataScales. Our results show that the method can achieve up to 36.9 times faster, and 2.1 times less physical memory (RAM) compared to commercial simulation tools such as COMSOL Multiphysics® software. This substantial reduction in computational resources make AutomataScales more efficient and accessible for researchers to explore broader design variables in their early design process with or without computational constraints.

Keywords: AutomataScales, Cellular automata, Multiphysics Simulations, Multi-Resolution Analysis.

1. INTRODUCTION

The advancement of computational power has revolutionized the way we approach the design of complex engineering systems. Instead of relying solely on rapid prototyping, we now heavily rely on physics-based simulations. While some of these methods can be time-intensive, the integration of machine learning with computer graphics offers a promising horizon. This enhances user interaction and visualization of data, while also increasing simulation speeds.

Physics-based dynamic simulation modeling in 2D or 3D is essential in understanding and designing complex systems such as electric propulsion thrusters. Types of electrical thrusters can be classified as electrothermal systems for Resistojets and Arcjets, electrostatic systems for ion thrusters, and electromagnetic systems for Pulsed Plasma thrusters (PPT) or Hall Effect thrusters (HET) [1]. Information about the 2D and 3D responses of engineering properties in electric propulsion like temperature, density, viscosity, pressure, and velocity enhances design by allowing lower-cost testing and flexible inspection. The cost of experimental studies is high, and traditional measurement techniques are limited in spatial or temporal resolution, making comprehensive studies challenging. For example, when attempting to develop an ion thruster, experimentation is limited due to the size, equipment, and conditions of the vacuum chamber, making it difficult to conduct tests in a wide range of operating scenarios [2]. In addition, more complicated systems are usually composed of several sub-systems (e.g., hollow cathode, ionization chamber [3–5], anode, magnetic properties, and neutralizer [6]) with multi-physics interactions between each sub-system such as mechanical interactions, electrical interactions (e.g., surface charging, electric forces), magnetic interactions (e.g., magnetic interactions [7]) and physical interactions (e.g., erosion [8], particle-particle interactions and collision [9]). Thus, preliminary numerical simulations enable designers to study accurate complex dynamical systems (e.g., nonequilibrium plasma dynamics [2,10], plasma turbulence [11–13]) to obtain the distributions of plasma parameters for various system designs [14].

To analyze how fluids flow, most computational fluid dynamics (CFD) software tools require a numerical method and a mathematical model of the physical case; the underlying equations can vary significantly by flow regime. The Navier-Stokes (NS) equation is recognized as a mathematical representation of the fluid-related physical model. In fluid dynamics, gas dynamics, and thermodynamics, the method may be applied to describe changes in all physical properties. This includes mass transfer, phase change, heat transfer, and chemical reactions. Because the Navier-Stokes equation can be used in various fields, assumptions for each simulation model are required to obtain accurate results such as conservation of mass, conservation of momentum, inviscid fluids, and Newtonian fluid. In a fluid that conducts electricity, there is a possibility that an electric body force will occur, which can significantly alter the trajectory of the fluid flow. Ionized gas is one such fluid. It consists of free electrons, neutral components, and ionized components. Therefore, addition equations are needed such as the kinetic-energy balance equation for the streamlines, the internal energy of a molecule, the matter-energy equation, the Joule effect, caused by the flowing of a conduction current in the electromagnetic field, and the electromagnetic energy equation [15]. By using the modified NS equation for the hypersonic flow with an electromagnetic field, complex phenomena can be solved by keeping most of the parameters constant and varying only a single variable of interest such as velocity of a particle. The accuracy of numerical simulation studies focusing on a single parameter of interest from a single subsystem must therefore be balanced with the complexity of the model [16,17]. Therefore, it is extremely difficult to develop adequate assumptions and equations for each subsystem and integrate them to simulate multi-physics interactions for the whole system.

While machine learning promises rapid predictions and detailed insights into real-time users' interactions such as those between fluids and objects, its complex neural networks and data dependency present significant challenges. It is often difficult to comprehend and only capable to accurately model the physics behavior within the range of available data [18]. Recent research into Cellular Automata has shown the potential to integrate with Multiphysics interaction simulation models, presenting a revolutionary avenue to support early design decisions. This integration promises to offer both depth and speed, enabling designers to anticipate a multitude of physical phenomena at the foundational stages of design.

Traditional simulation methods for particle trajectory in electric propulsion such as the Particle-In-Cell (PIC) method facing challenges due to its computational intensity and high memory demands. The simulation using this method can take substantial computing resources from 2.5 to 21 days to compete for each model, varying by the complexity of each case. Additionally, these methods have a steep learning curve for researchers to adopt quickly and effectively. The high computational demands also lead to increased power consumption. For instance, gigabyte (GB) of RAM consumes approximately 0.1875 watts. This translates to total power consumption ranging from 87.1 to 145.2 megawatts per simulation case [19–21]. These limitations from both computing resource and time reduce their computational efficiency in resource-constrained environment [22]. The AutomataScales method addresses these issues by integrating discretization techniques with cellular automata and utilizing a multi-layer, multi-resolution approaches. This method models complex multiphysics interactions using a hybrid numerical scheme that combines discrete and continuous techniques, thereby reducing computational time and memory usage while maintain the simulation accuracy. AutomtaScales able to capture intricate behavior in various particle trajectory (ionized particles and primart and secondary electrons) by using transition rules from the cellular automata framework as well as plasma physics phenomena (particle collisions and ionization). It provides a scalable and adaptable framework for multiphysics simulations with almost real-time simulation (up to 0.1 second per time step). Our research demonstrates that AutomataScales can perform up to 36.9 time faster and use 2.1 times less memory (RAM) compared to commercial tools such as COMSOL Multiphysics. This significant reduction in computational resources enables AutomataScales more efficient and accessible which allow researchers to explore broader range of design spaces in the early design process with or without computational constraints.

This research aims to comprehensively elaborate on the theory behind the AutomataScales method (previously referred to as Layered Automata [23]). Moreover, we conduct extensive performance analysis based on the prior case study [23] to validate the efficiency of the approach. This paper contributes to three main areas as follows: 1) demonstrating the robustness of the AutomataScales computational grid that ensure stable and consistent simulations across various scenarios. 2) analyzing the influence of different layer resolutions on simulation accuracy and computational efficiency, and 3) evaluating the accuracy of the AutomataScales method in modeling complex system with lower computational costs. These contributions support researchers by offering a low-fidelity simulation tool that can promptly visualize complex physics phenomena. Additionally, the AutomataScales method could accelerates the learning process for researchers to set up higher fidelity simulations accurately and efficiently.

The structure of this paper is delineated as follows: Section 2 delves into established computational meshing techniques, electric propulsion models, cellular automata methodologies, and multi-resolution theory for each field. Section 3 elaborates on a novel Multiphysics interaction method, "AutomataScales: Integrating Scales in Multiphysics Modeling." Section 4 illustrates the research design underpinning this study. Computational performance insights are elaborated upon in Section 5 along with conclusions on the efficacy of the proposed approach.

2. BACKGROUND

This section provides a review of four main methodologies and theoretical foundations that support AutomataScales simulation as follows: 1) elaborating on discretization techniques used in physics simulation, 2) exploring the applications and challenges of ring cusp discharge type ion thrusters simulations, 3) examining the current development of cellular automata in multiphysics simulations, and 4) discussing the importance of multi-resolution techniques across various fields to enhance simulation accuracy and efficiency.

2.1 Discretization Techniques in Physics Simulations

Discretization techniques are foundational in computational physics simulations, serving as a bridge between the mathematical underpinnings and their practical computational applications. In areas like computational fluid dynamics (CFD) and plasma physics, these techniques are indispensable. Both fields often rely on complex differential equations to describe fluid motions or plasma behaviors, but many real-world scenarios and geometries elude exact analytical solutions. Hence, an approximate yet precise solution is paramount, and this is where discretization comes into play. By converting the continuous mathematical descriptions into computable models, discretization techniques ensure that simulations in CFD and plasma physics are not only accurate but also computationally efficient, capturing the intricate dynamics of fluids and plasmas. The four main categories of discretization techniques are mesh-based, cell-based, particle-based, and particle-in-cell or PIC method (Fig. 1).

2.1.1 Mesh-based Techniques

Finite Element Method (FEM) stands out in the realm of mesh-based methodologies for solving fields related equation such as Maxwell's equations (the electric and magnetic fields). This technique is paramount for tackling boundary and initial value problems described by PDEs. Two pivotal considerations underpin CFD meshing: mesh density and mesh geometry, which includes structured, unstructured, or hybrid meshing patterns. With the evolution of CFD, adaptive meshing, which adjusts in alignment with flow gradients and complex geometrical nuances, has gained prominence [24].

2.1.2 Cell-based Techniques

Cellular Automata (CA) is known for using the cell-based method to model complex systems (e.g., fluid dynamics and biology). One of the advantages of this method is that it can mimic system performance based on only local interactions governed by local update rules (transition rules). The transition rules can be divided into three categories: direct rules, multi-step rules (e.g., in plasma physics or multi-physics, the cell can be influenced by a particle velocity, electromagnetic fields), and probabilistic rules. The Lattice Boltzmann Method (LBM) is derived from the lattice gas automata or cellular automata methods. Fluid density on a lattice is simulated as a result of streaming and collision processes over a discrete lattice rather than solving the NS equations (FEM). As a result of the method, fluid behavior can be mimicked in controlled and complex environments, whereas other CFD methods are unable to do so. Additionally, the model can be parallelized due to its local dynamics feature obtained from the CA method, which is crucial to execute on the graphics processing unit (GPU). Combining or coupling it with other methods could be beneficial, such as using heat transfer in order to overcome its thermo-specific solution limitation and using the Galilean or Newtonian transformation technique to overcome its high-speed fluid flow limitation [25].

2.1.3 Particle-based Techniques

In CFD, the particle-based approach revolutionizes fluid dynamic equations by substituting fluid continuum with a discrete particle set. Its prowess lies in accurately modeling pure advection, facilitating solutions for multi-material challenges, and enabling seamless interface detection. Smoothed Particle Hydrodynamics (SPH) introduces a Lagrangian framework, allowing continuum equations' discretization directly at designated discrete points, bypassing the conventional spatial mesh systems like FEM. The fluid dynamics isn't spatially fixed but flows with the current. The variable values at each particle can be estimated by aggregating contributions from neighboring particles. Crucial determinants for fluid motion encompass external forces, fluid viscosity, and shifts in the pressure field.

2.1.4 Particle-in-Cell (PIC)

Particle-in-cell (PIC) is a popular in plasma and laser dynamics as the method can use to solve plasma dynamics problem mainly by treating plasma as particles so that the plasma can consider as a continuum fluid and be able to apply Navier Stoke equation with the minimum assumptions [26] by treating all physical and chemical progress as the consequence of collisions. Therefore, it is obeying the statistical laws and gives the accurate kinetic information of plasma parameters [27–31]. The first micro-DC ion thruster PIC simulation developed by [5,32] is able to predict the performance of ion thruster and achieve 59% of total efficiency at low power consumption with the efficient miniature discharge configuration. Recently, there are studies that aim to design more efficient ion thruster by combining the PIC method with other approach such as the Monte Carlo, or develop the model so that it is parallelizable and could be run on high-performance computing clusters (HPCs) or cloud computing (e.g., Amazon Web Services, AWS) [33]. As a result, the method is able to predict similar outcomes compared to [5] with broader configuration (e.g., mass flow rates, discharge voltages.)

In addition, the embodiment of sparse-grid (SG) techniques in PIC has led to significant advancements in complex system simulations. These techniques optimize simulation models' efficiency by employing coarser sub-grids to reduce the computational cost for high resolution models [34–36]. This approach aligns with the AutomataScales multilayers concept by using coarser grid resolution at low transition layer for calculation. These approaches represent a leap in computational physics, enabling more effective and accurate simulations of multi-scale and multi-physics scenarios.

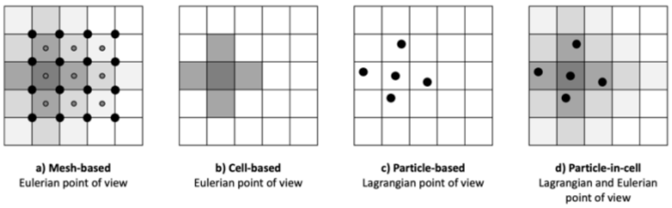

Figure 1: Discretization technique in physics simulations. a) Mesh-based; b) Cell-based; c) Particle-based; d) Particle-in-cell

2.2 Ring Cusp Discharge Type Ion Thruster

For mission and spacecraft designers, ion thrusters offer a significant and unique capability due to their ability to provide desired thrust levels, thrust control, propellant efficiency, and high total efficiency (up to 70%). In recent years, technologies and approaches used by spacecraft have evolved and improved in order to enhance their capabilities and efficiency. There are generally two types of successful ion thruster discharges, namely those generated by direct current electron bombardment (such as Kaufman or ring cusp) and those generated via electromagnetic fields, such as radio frequency (RF) or microwaves. In direct current discharges, electrons are typically discharged through a hollow cathode. It has been observed that ring-cusp discharges and thrusters are the most effective for conventionally sized ion thrusters [37]. An ion thruster working mechanism can be divided into three processes. First of all, a plasma is created in the discharge chamber, the ions are then accelerated through two (possibly three or four) ion optics grids, and then, a neutralizer emits electrons to provide system charge neutralization.

A discharge chamber model (DCM) that includes dynamic electromagnetic fields was developed by [5] in 2005. It proposed a two-dimensional hybrid diffusion model that treated ions, secondary electrons, neutrals, and primary electrons with diffusion models, zonal models, and particle tracking models to study the plasma inside the discharge chamber. However, it is not capable of capturing all aspects of the physics that occur inside the discharge chamber. To simulate the plasma inside the discharge chamber in a detailed manner, [38] developed an axisymmetric two-dimensional PIC with Monte Carlo Collisions (MCC) model that tracked the major particles in the chamber individually. It should be noted, however, that most numerical simulations of the ion thruster discharge chamber are based on the electrostatic model, which disregards the time-varying electromagnetic characteristics. Due to the time-varying electromagnetic characteristics of the discharge chamber, the model proposed by [39] may provide a more accurate and detailed description of the chamber. Rather than using the hybrid model [38], this method separates the physics model into three sub-models: Using the PIC method to track particles, electromagnetic fields are solving by Maxwell equations, and collision processes are described using MCC. In addition, high fidelity simulations using this method can require substantial computing resources ranging from 2.5 to 21 days with 32 to 448 CPU cores, and 9 to 14 days with 1 to 2 graphics processing units (GPUs), depending on the complexity of the model [22]. Consequently, the amount of RAM needed depend on CPU core. For High-Performance Computing (HPC) systems, the system typically requires 8 GB of RAM per core [21]. This results in a total of 256 to 3,584 GB. Consequently, power consumption increases as 8 GB of RAM consumes approximately 1.5 watts [19,20]. Therefore, traditional PIC methods could consume between 87.1 and 145.2 MW, assuming the systems uses the same computing architecture and only varies the amount of RAM.

Due to the significant demands of computing resource, there is a clear need for innovative approaches to better handle complex system. Addressing these challenges is crucial for supporting researchers in exploring complex multiphysics interactions with wide range of design parameters, improving simulation accessibility, and reducing costs during the early development process.

2.3 Physics-based Cellular Automata

Automata with probability is a generalization of finite automata with non-determinism. Only the probability is used to create the transition matrix for its transition function. Thus, all states have a weighted set of next states where the weights must sum to 1. These weights must also be reflected in the notions of states and acceptance. The state of the machine at a given step must now also be represented by a stochastic vector of states, and a state is accepted if its total probability of being in an acceptance state exceeds some cut-off. Phase or state transition of matter will be easier to inform the relevant calculations when the state of matter is explicitly defined by and appropriate state variable. An investigation by [40] illuminated the harmonization of CA with the Lattice Boltzmann method (LBM), elucidating accurate outcomes for multiphysics interactions during the solidification phase of forced convection processes. [41] proposed a multi-physics interaction model by using the CA coupling with LBM to successfully determine energy transfer and state of matter such as solid, liquid, and gas. The state of matter also includes the mixture between each state such as solid and liquid, liquid and gas, or gas and solid. Building on the foundational work in multi-physics modeling using the Cellular Automaton (CA) approach, as delineated by [42], the research presents a groundbreaking methodology tailored to simulate thermo-fluid dynamics within complex systems.

Traditional simulation methodologies often find themselves encumbered by computational inefficiencies and a lack of adaptability, especially when navigating systems marked by intricate geometries and nuanced boundary conditions. In contrast, the CA methodology, characterized by its grid-centric architecture and rule-driven evolution, emerges as a potent alternative. Its inherent streamlined nature, harmonized with an ability to depict complex behaviors, earmarks CA as an optimal tool for multi-physics simulations. Beyond mere computational efficiency, the CA framework offers unparalleled adaptability across diverse scenarios, positioning itself as a game-changer in the realm of complex physics simulations. The culmination of the author's research not only underscores this potential but also pioneers a method adept at handling intricate multi-physics challenges encompassing diffusion, advection, reaction kinetics, and even external interactions such as gravity and tension forces.

2.4 Multi-Resolution Simulation

Mechanical systems often exhibit intricate behaviors across multiple length and time scales. Such systems constitute a significant challenge for engineers and researchers to identify the increment and time step for each simulation model so that it satisfies the Courant-Friedrichs-Lewy (CFL) condition. Multi-resolution or multi-scale simulation has emerged as a compelling approach to address these challenges by providing a unified framework for modeling diverse scales within a single simulation.

In this section, we will delve into three multi-resolution concepts from different areas: 1) the adaptive refinement from the finite element method, 2) the multi-scale techniques from machine learning, and 3) the multi-resolution concepts from computer graphics.

2.4.1 Adaptive Refinement in the Finite Element Method

The Finite Element Method (FEM) is a fundamental method in engineering simulations. However, traditional FEM implementations often employ uniform mesh resolutions that usually become computationally intensive and inefficient when dealing with high fidelity models. To overcome this challenge, adaptive refinement in FEM presents a solution by adjusting mesh resolution based on local error estimations. This adaptive approach optimizes computational resources by allocating finer mesh at high transition or turbulence region [43]. While adaptive refinement offers a powerful tool for simulation, it may not be suitable for all problems. For instance, an airfoil modeling in often require mesh divisions into four distinct regions with different resolutions in order to capture different transitional areas and accurately predictions lift and drag coefficients [44]. These approaches require substantial effort during the pre-processing stage which tailor the technique to specific scenarios and boundary conditions.

2.4.2 Multi-Scale in Machine Learning

The advancement of machine learning in data-driven modeling allows researchers to gain insights from complex systems without knowing all variables encapsulated in the systems. Multi-scale machine learning takes this a step further by leveraging data generated at smaller scales to make predictions at larger scales while maintaining the same accuracy. Recently, there is studies that attempt to apply ML to various types of simulation domains such as NS fluid flow, stress analysis, electromagnetic, and grid interpolation (applied data generated from smaller-scale training grids to larger-scale grid resolution) [45]. The approach can precisely predict the outcomes of the four simulation models (The root-mean-squared error, RMSE varies between 0.00214 to 91.57) with minimal computational time (between 0.09-14.71 seconds). The idea of grid interpolation or multi-grid resolution has been shown to reduce the computational cost and but also reduce the predicted outcome errors between 20% and 70% [46].

2.4.3 Multi-Resolution Concepts in Computer Graphics

The field of computer graphics needs realistic simulations and visualizations particularly in control of light and texture. Multi-resolution concepts are pivotal in achieving these goals, enabling the rendering of immense complex scenes with varying levels of detail [47]. By hierarchically representing objects and textures at different resolutions [48], the techniques allocate computational resources accordingly and result in realistic and interactive experiences with the same precision. Understanding these principles in computer graphics provides valuable insights into visualizing the behavior of high-fidelity complex systems from lower fidelity data.

3. METHODOLOGY

The AutomataScales method transforms complex physics phenomena into a discrete computational model based on the Cellular Automata (CA) architecture. CA's cell-based structure also allows AutomataScales to encapsulate physical interactions within local or transition rules and enables the integration with other approaches, such as multi-resolution and multilayer strategies. These techniques significantly enhance the AutomataScales computational capability by modeling different physics interactions with varying detail levels that potentially reduce the computational cost for each time step while reduce the complex boundary conditions within the model.

This section will elaborate on related theories and assumptions for the AutomataScales simulation model for electric propulsion applications.

3.1 AutomataScales: Level of Reality

In the landscape of mechanical engineering simulations, selecting an appropriate scale (microscopic, mesoscopic, or macroscopic) is pivotal for accurately capturing the desired physics phenomena. Microscopic simulations illuminate the quantum behaviors foundational to material properties; mesoscopic simulations reveal the statistical behaviors of particle assemblies; macroscopic simulations encompass the global behavior of the large-scale system (e.g., averaged velocity of the particle trajectory in electric propulsion).

The AutomataScales approach, grounded in the mesoscopic perspective, adeptly captures the dynamics that need to be fully observable at the macroscopic level and as detailed as at the microscopic level. Within this intermediate domain, AutomataScales offers a versatile simulation platform. Each layer within the AutomataScales model can be tuned to a specific scale of reality, allowing for a granular approach to precision and time optimization in simulations. Thus, it facilitates the balance between detail and computational efficiency necessary for engineering design to determine a broad range of variables to be optimized, especially during the early stage of the design process.

3.2 AutomataScales: Research Design

In this study, AutomataScales and COMSOL Multiphysics® software were utilized in parallel to simulate electric propulsion dynamics. Both platforms adhered to a standardized parameter set: particle numbers, computational grid resolution, simulation geometry, and boundary conditions as illustrated in Table 1 and Fig. 2. The simulations conducted on a system with an Intel Xeon CPU (E3 1585L v5, 3.00 GHz) and 16 GB of RAM. The AutomataScales model was implemented using MATLAB.

COMSOL was chosen as the benchmark for this study due to its established reputation as a leading commercial simulation software in both academia and industry field. COMSOL provides a comprehensive suite of tools for simulating various physics phenomena through its unique particle tracing approaches to solve for discrete trajectory instead of continuous field [49]. This unique method aligns with the foundational theories of AutomataScales, making COMSOL an ideal reference point for evaluating the performance of the AutomataScales simulation.

Table 1: Simulation parameters

Module	Parameters	Value	Unit
Mesh	Mesh geometry	Quadrilateral	-
	Number of mesh	10,000	-
	Computational grid resolution	0.1	mm
Electrostatic field	Anode	10,000	Voltage (V)
	Screen grid	10,000	Voltage (V)
	Accelerator grid voltage	-10,000	Voltage (V)
	Discharge cathode	-5,000	Voltage (V)
Particle properties	Particle mass	2.18×10^{-25}	Kilogram (kg)
	Electron mass	9.109×10^{-31}	Kilogram (kg)
	Particle charge	1.6022×10^{-19}	Coulomb (C)
	Initial energy	1000	Electron volts (eV)
Collision and ionization	Cross sectional radius	3×10^{-19}	m^2
	Background number density	1×10^{20}	m^{-3}
	Background gas molar mass	0.131	kg/mol
	Avogadro constant	6.022×10^{23}	mol^{-1}
	Ionization energy of Xenon	1.943×10^{-19}	Joules (J)
	Maximum number of primary particles	20,000	-
	Maximum number of secondary particles	20,000	-
Time dependent study	Initial time step	0	second
	Time step	10^{-10}	second
	Final time step	10^{-8}	second

The evaluation of performance metrics in this study encompassed computational time and accuracy in predicting physical phenomena. Specifically, for the AutomataScales method, in-depth analyses included grid convergence and layer resolution assessment. Our COMSOL model incorporated modules for electrostatics, charged particle trajectory analysis, and electric-particle interactions.

The study of particle trajectory in electric propulsion is critical as it correlates with the probability of particle collisions and ionization events. Given the stochastic nature of particle trajectory, it is imperative to derive results from multiple simulations for reliability. In this context, data from COMSOL were derived from 10 simulation runs to calculate both the average and maximum particle velocities (primary electrons, secondary electron and ionized Xenon particles) at the engine exit (45-55 mm from the bottom), specifically 97 mm from the left, after 100 time steps (Fig. 3). Similarly, data from AutomataScales were obtained from 100 simulation runs at the same position and time steps as those in the COMSOL simulations.

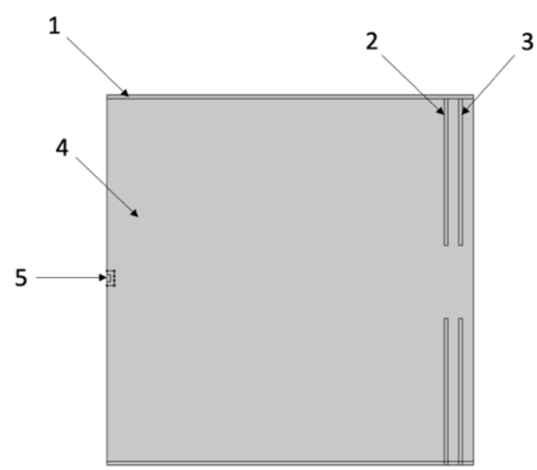

Figure 2: Electric propulsion model components: (1) anode, (2) screen grid, (3) accelerator grid, (4) ionization chamber, (5) discharge cathode

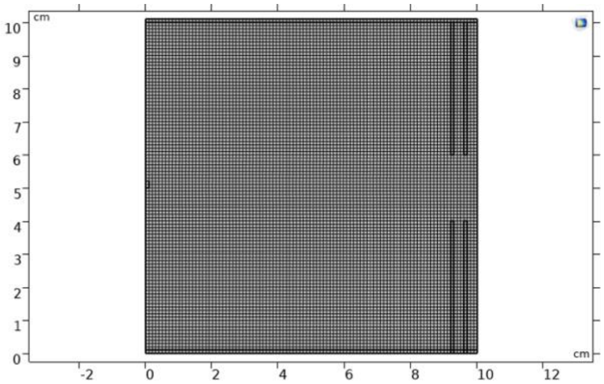

Figure 3: Quadrilateral computational grids optimized for plasma physics simulations

3.3 AutomataScales: Numerical Scheme

The following section provides details of three numerical schemes for electric propulsion dynamics, including a fields solver, particle trajectory and collision, and ionization.

3.3.1 Fields Solver

The electromagnetic fields are divided into two parts: one is the initial electric field when the voltage first applied on the Anode wall (Fig. 2) and the initial magnetic field generated from the permanent magnets. The electrostatic field is obtained from the Poisson's equation (Eq. 1) where V is the electric potential, ρ is the charged density, ε_0 is the vacuum permittivity ($\varepsilon_0 = 8.854 \times 10^{-12} \mathrm{F} \cdot \mathrm{m}^{-1}$). The electric field ($E$) is calculated from is the negative gradient of the potential from Eq. (1).

$$\nabla^2 V = -\frac{\rho}{\varepsilon_0} \tag{1}$$

$$\vec{E} = -\nabla V \tag{2}$$

In this case study, each computational grid cell is assigned a single particle. Consequently, the estimated impact of ion and electron densities on each cell's potential is approximately 18.09 µV. This value exerts a negligible influence on the overall electrostatic field of the model, particularly when contrasted with the boundary conditions at the walls or electrodes, which are set at -5,000 V, -10,000 V and 10,000 V, respectively. Given this minimal impact, it is reasonable to postulate that the charge density does not significantly alter the electrostatic field. Therefore, for the purposes of this study, Poisson's equation simplifies to Laplace's equation (Eq. 3).

$$\nabla^2 V = 0 \tag{3}$$

3.3.2 Particle Trajectory

Particle trajectory of the model is driven by the Newton-Lorentz equation of motion in Eq. (4), where $\vec{v}$ is the particle velocity, q is the electric charge of particle, $\vec{E}$ is the magnitude of electric field, and $\vec{B}$ is the magnitude of magnetic field. In addition, the velocity of each particle could be calculated based on Newton's second law in Eq. (5).

$$F_L = q\left(\vec{E} + \vec{v} \times \vec{B}\right) \tag{4}$$

$$\sum F = \frac{d}{dt} m\vec{v} \tag{5}$$

3.3.3 Collision and Ionization

The collision model is driven by calculating the collision probability ($P_{collision}$) in Eq. (6), where ν is the chance of collision which depends on the cross-sectional area of particle (σ), background particle density (N_d) and corrected velocity in Eq. (7). For cold gas approximation collision, the relative background velocity (g) is equal to the particle velocity, v.

$$P_{collision} = 1 - \exp\left(-\nu\Delta t\right) \tag{6}$$

$$\nu = N_d \sigma(v)|g| \tag{7}$$

The ionization process is initiated upon the occurrence of a collision, necessitating a specific energy input (ΔE) to ionize a Xenon atom, quantified as 12.13 eV. Subsequently, the velocity of the electron colliding with a neutral Xenon atom will be adjusted in accordance with Eq. (8) & (9), where g' is the post-collision relative velocity between a particle and background particles, m_p is the mass of particle (primary electron), m_g is the background mass, v' is the post-collision velocity of a particle.

$$|g'| = \sqrt{g \cdot g - \frac{2\Delta E\left(m_p - m_g\right)}{m_p m_g}} \tag{8}$$

$$v' = v - \frac{m_g}{m_p + m_g}(g - g') \tag{9}$$

3.4 AutomataScales: Structure

The following characteristics define the AutomataScales simulation architectures for the electric propulsion dynamics.

3.4.1 Cell Types

The AutomataScales model is typically structured in 2D or 3D spatial simulations. Cell type will influence the type of cellular spaces, and computational intensity of the model including the interaction between layers and transition rules for each cell properties or states. In this research, 2D cell type was selected.

3.4.2 Cell Spaces

The model prioritizes a quadrilateral arrangement, as it is supported by MATLAB architecture, and aligns with COMSOL's optimized meshing for plasma physics (Fig. 2 & 3). It is crucial for simulating the behavior of various particles in electric propulsion with the same cell or grid arrangement, where the conservation of momentum might be changed based on different grid arrangements.

3.4.3 Cell States

In the AutomataScales framework, cell states are created to represent a spectrum of physical conditions such as kinetic behavior of particles. Primary states signal the presence or absence of physical quantities within a cell, while secondary states offer a detailed enumeration of these quantities (e.g., velocity and energy levels) necessary for portraying the thermodynamics of propulsion systems. The sub-states are meticulously formulated to capture the ionization stages of Xenon, a common propellant in electric propulsion. By employing a probabilistic CA approach [41], transitions between these states are modeled to reflect the stochastic nature of particle collisions and ionizations, thus providing a comprehensive representation of plasma behavior.

3.4.4 Neighbor Types

The neighbor type in the AutomataScales method is selected with a focus on computational efficiency and mesoscopic fidelity for electric propulsion problems. The Von Neumann neighbor type is employed due to its less dense configuration, which only requires four neighbors around each cell (8 neighbors for the Moore neighbor type). The number of neighbors becomes increasingly significant when the neighborhood radius extends beyond a single cell [44]. The approach allows for the detailed analysis of local interactions, particularly at chamber walls and electrodes (e.g., discharge cathode, screen, and accelerator grid), managing the overall behavior of the simulation.

3.4.5 Layers

The AutomataScales model consolidates a multi-layered architecture that is crucial for modeling multiphysics interactions. Each layer represents different physical quantities, such as voltage, velocity, energy (eV), or electric field. Alternatively, the layer could be created based on the associated simulation domains (e.g., fluid dynamics and electromagnetics). In this research, the way electric fields influence particle acceleration is modeled through these interactive layers to observe the exhaust velocity of the propulsion system. The innovation of this method is to maintain optimal resolution for calculation across layers while ensuring that simulations predict system behavior.

3.4.6 Transformation or Transition Rules

The AutomataScales model's transformation rules are the engines that drive the evolution of cell states over time based on their current state and the states of neighboring cells. These rules considering the type of neighborhood to ensure that each cell's behavior is accurately represented. The implications of these rules are vast such as determine the collision rate when ionized particles collide with the engine walls. The sophistication of these rules allows for simulations that can predict the operational performance of propulsion systems under a variety of conditions (e.g., power level, component, geometry).

The transition rules for typical particle behavior in our simulation framework was implemented in this research where each particle has the same probability to advance in six directions. In addition, our study acknowledges that high-energy particles in electric propulsion systems exhibit complex behaviors, such as moving through multiple grid cells in one time step. We've established additional rules where these high-energy particles' movements are influenced by the average velocity of all particles.

3.5 AutomataScales: Time Complexity

Time complexity, usually described as the big O notation, $O(N)$, is vital for evaluating and developing each simulation model, as it represents the computational cost of running a simulation relative to the total number of grid points.

When applied the AutomataScales simulation to electric propulsion models, we anticipated that a time complexity would be approximately equal to the cellular automata (CA) framework. For a CA with the total number of cells of N and a fixed set of rules, the time complexity for a single update across the simulation domain is typically $O(N)$, as each cell's state is updated once per time step. Given L layers where L is the subset of all positive integers ($L \subseteq \mathbb{Z}^+$), each potentially operating at a different resolution, the time complexity for updating the entire multilayer system once can be classified as $O(N)$ if the layer are independent; $O(L \cdot N)$, if the layers are updated sequentially; and $O(L^2 \cdot N)$ if the layers are interdependent where the state update of each layer depends linearly on the states of all other layers.

$$\text{Time Complexity} \approx \begin{cases} O(N), & \text{if layers are independent,} \\ O(L \cdot N), & \text{if layers are updated sequentially,} \\ O(L^2 \cdot N), & \text{if layers are interdependent.} \end{cases}$$

In our model, we deal with 10 sequentially updated layers: electric field; particle force, particle acceleration, and particle velocity for primary electron, secondary electron and secondary ionized particles. Presumably, the time complexity can be estimated as $O(10N)$. However, constants are omitted from the notation due to their non-significant impact on scalability. Therefore, the overall time complexity of the AutomataScales model in our study is equal to $O(N)$.

4. COMPUTATIONAL PERFORMANCE ANALYSIS

This section delves into the computational performance of the AutomataScales method, focusing on its application in simulating electric propulsion systems. The objective of this section is to evaluate the model's efficiency, scalability, and accuracy through a series of comprehensive analyses.

4.1 Grid Convergence Analysis

In this study, we assessed simulation accuracy across various refined computational grid sizes. The goal of this analysis was to verify the convergence of the model. Thus, ensuring model’s accuracy and reliability.

To facilitate this analysis, we employed three distinct grid resolutions with quadrilateral grid type: coarse (2 mm), medium

(1 mm), and fine (0.5 mm). These grids were utilized to evaluate the grid convergence index (GCI) using the AutomataScales method. The analysis was based on the average velocity of the primary electron particle, derived from an ensemble of 100 simulation runs. The observed average velocities were 2.61×10^7 m/s for the fine grid (ϕ_1), 2.75×10^7 m/s for the medium grid (ϕ_2), and 6.84×10^7 m/s for the coarse grid (ϕ_3), derived from 100 simulation runs, compared to the exact solution obtained from COMSOL (2.47×10^7 m/s).

The computation of the grid convergence index incorporated four parameters: the Quantity of Interest (ϕ), represented by the averaged velocity; the Relative Change in quality of interest (e) obtained from Eq. (10); the grid refinement ratio (r) calculated from Eq. (11); and the observed order of accuracy (p) computed from Eq. (12). These factors and their interrelations are comprehensively detailed in Table 2. This methodical approach to grid convergence analysis in Eq. (13), where F_s is equal to 1.25 based on 3 grid point data, ensures that our numerical model accurately captures the intricate dynamics of the physical phenomena under investigation. [50–52]

$$e_{1,2} = \frac{|\phi_1 - \phi_2|}{\phi_1},\ e_{2,3} = \frac{|\phi_2 - \phi_3|}{\phi_1} \tag{10}$$

$$r_{1,2} = r_{2,3} = \frac{h_3}{h_2} = \frac{h_2}{h_1} \tag{11}$$

$$p = \frac{\ln(e_{1,2}/e_{2,3})}{\ln(r)} \tag{12}$$

$$\mathrm{GCI}_{e_{1,2}} = \frac{F_S \times e_{1,2}}{(r^p - 1)} \tag{13}$$

Table 2: Grid convergence analysis results for different grid size

Grid size (mm)	Mean particle velocity (m/s)	Percentage different from exact solution	Total computed time	GCI
2	6.84×10^7	176.9 %	2.48 s	-
1	2.75×10^7	11.34 %	10.73 s	6.59 %
0.5	2.61×10^7	5.67 %	121 s	0.24 %

4.2 Layer Resolution Analysis

In our study, we analyzed various layers within the AutomataScales model to investigate their influence on the overall simulation resolution, explicitly focusing on the particle trajectory and electric field layers. This comprehensive analysis was crucial to optimizing resolution for each layer and verifying the multi-resolution approach for AutomataScales.

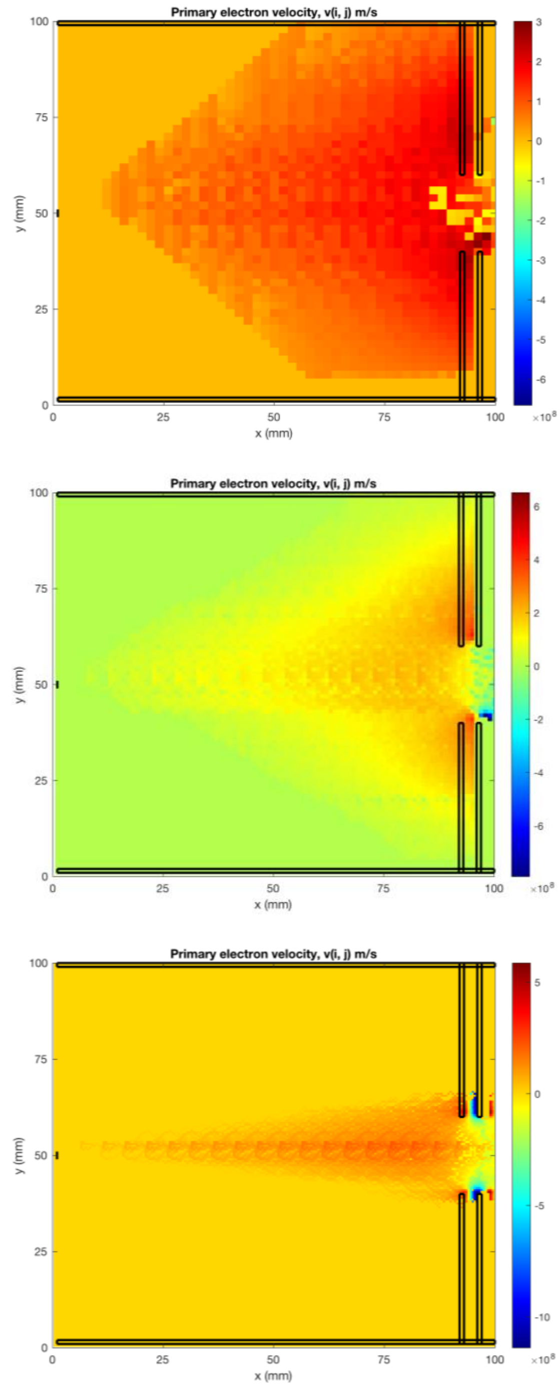

Figure 4: Multi resolution of primary particle velocity with grid size 2 mm (top), 1 mm (middle), and 0.5 mm (bottom)

Fig. 4 graphically depicts the particle velocities obtained at different grid resolutions: 2 mm, 1 mm, and 0.5 mm. The corresponding velocity data for each grid resolution combination are presented in Table 3. A remarkable observation is that the simulation model with a 1 mm grid size for the electrostatic field coupled with a 0.5 mm grid size with particle trajectory effectively demonstrates the efficacy of our multi-resolution approach. Additionally, the averaged velocity from the AutomataScales converged to the solution obtained from COMSOL, and the results from finer resolution when both layers' resolution equal to 0.5 mm.

Table 3: GCI results and particle velocity relative to different grid resolution.

Particle trajectory grid size (mm)	Electrostatic field grid size (m/s)	Percentage different from exact solution	Total computed time	GCI
2	5.77 x 10^7	133.6 %	2.22 s	-
1	2.75 x 10^7	11.34 %	10.73 s	1.87%
0.5	2.71 x 10^7	9.72 %	48.89 s	0.02%

4.3 Level of Reality, Time and Space Complexity

The model's ability to accurately capture the microscopic, mesoscopic, or macroscopic reality was scrutinized at the screen and accelerator grid region, the particle behavior at the exit including secondary electron and ionized Xenon particles was illustrated in Fig. 5 & 6. Additionally, we analyzed the time complexity of the AutomataScales method in various simulations to evaluate its scalability and performance in practical applications. The error between average computed particle velocity from COMSOL and AutomataScales equal to 11.34% and 4.08% for the peak velocity (Table 4).

Table 4: Comparative analysis of the proposed method and the COMSOL method for 100 time step

Method	COMSOL	AutomataScales
Total mesh	9,999	10,000
Total particles	40,000	40,000
Mean calculated peak velocity of primary particles (m/s)	3.92 x 10^7	3.76 x 10^7
Percentage variation in peak primary particle velocity	4.08 %	
Mean calculated velocity of primary particles (m/s)	2.47 x 10^7	2.75 x 10^7
Percentage deviation in mean primary particle velocity	11.34 %	
Total computed time (s)	161 – 394	10.73
Relative performance (particles per second)	101 – 248	3,727
Physical memory usage (GB)	2.18 – 3.88	1.84

The time complexity analysis of our simulation model, which involves the manipulation of 10 matrices each of size N by N, demonstrated a more complex computational profile. In contrast to initial estimations of $O(N)$, empirical observations from our study pointed to a more sophisticated relationship within the simulation model. Notably, when the input size was doubled from N = 50 to N = 100, and then to N = 200, the increase in computation time did not adhere to the expected linear scaling of $O(10N)$ or quadratic scaling of $O(10N^2)$. Instead, the observed ratios of time increase were significantly lower (4.33 and 11.28). Therefore, the ratios of time increase suggesting a computational demand growth that is less steep than quadratic but more than linear scaling.

This outcome was further supported by our multi-resolution analysis, where the electrostatic field layer was kept constant at N = 100 (1 mm grid size) coupled with the particle trajectory layer varied from N = 50 (2 mm grid size) to N = 100, and then to N = 200 (0.5 mm grid size). The computation times recorded for these variations (2.22s, 10.73s, and 48.89s respectively) also did not conform to the $O(10N^2)$ complexity model, further emphasizing a less than quadratic scaling.

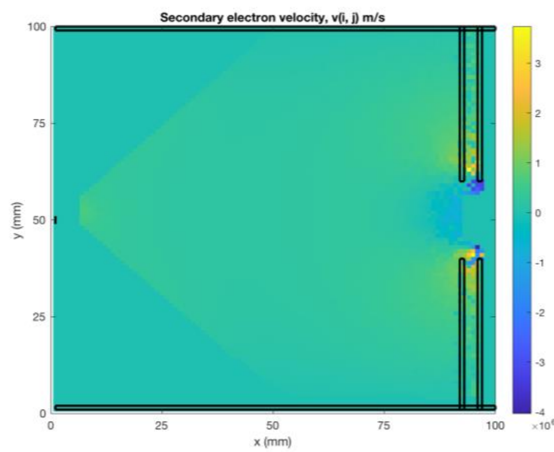

Figure 5: Particles behavior and velocity of secondary electron

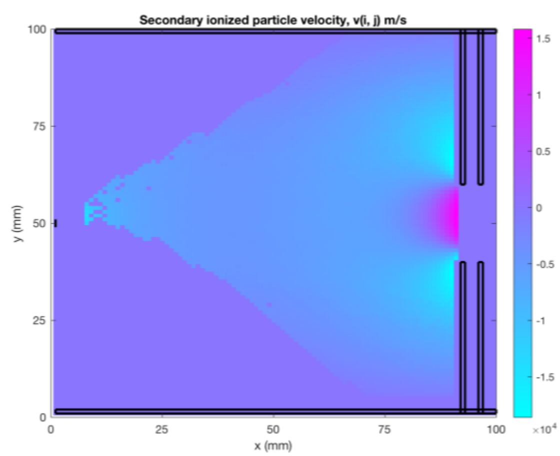

Figure 6: Particles behavior and velocity of ionized xenon particles

5. DISCUSSION AND CONCLUSION

The grid convergence analysis conducted in this study showed that a grid resolution of 1 mm is adequate for achieving stable results in our simulations as well as in our COMSOL model. Specifically, this resolution yields an error margin below 12% for the average particle velocity, and under 5% for the average peak velocity of particles. Furthermore, in our layer resolution analysis, it was determined that an electric field layer grid resolution of 1 mm, in conjunction with a particle trajectory layer grid resolution of 1 mm and 0.5 mm, resulted in a Grid Convergence Index (GCI) of less than 0.05% and an error margin below 10%. This configuration reduced the simulation time by approximately 2.47 times (48.89 seconds compared to 121 seconds).

Our comparative analysis with COMSOL Multiphysics software indicated that the AutomataScales method accurately models particle trajectories in electric propulsion systems. The AutomataScales simulation results not only align with the results from COMSOL, but also offer significant computational benefits, being up to 36.9 times faster and consuming up to 2.11 times less physical memory. The faster computing time and reduced physical memory usage of the AutomataScales method offer significant benefit beyond computational efficiency. As detailed in the background section 2.2, these improvements could substantially decrease simulation durations from 2.5 to 21 days to approximately 1.63 to 13.66 hours and the memory usage drop from 256-3,584 GB to 121.33-1,698.58 GB of RAM. If these reductions are achieved, power consumption could potentially drop from 87.1-148.2 MW to 0.133-15.66 MW. These potential enhancements would not only make AutomataScales simulation more feasible and accessible but also minimizing power consumption required to simulate each model.

Our time complexity analysis initially suggested an $O(N)$ complexity for our model as linear time growth. However, the computational times reveal some influences of additional factors beyond simulation domains (e.g., total cells or grid points) and boundary conditions. Despite our initial time complexity assumptions, the data indicates a more complex computational behavior as the simulation time growth between linear and quadratic rate (being equal to 4.33 and 11.28). Still, our findings affirm the model's effectiveness in managing large-scale simulations efficiently.

This study acknowledges several limitations. First of all, the particle trajectory model in our simulations is based on a stochastic process. This randomness is inherent in the nature of particle behavior in electric propulsion, where collisions occur unpredictably with probability from the Monte Carlo Collision method. As a result, both the direction and velocity of each particle may change with each simulation run. To counteract these stochastic variations and enhance the reliability of our results, it would be ideal to conduct a larger number of simulations (more than 100,000 runs) under the same boundary conditions. However, due to limited computational resources, we were unable to perform a sufficient number of simulations to thoroughly analyze the AutomataScales method and ensure statistical stability in the outcomes. Another limitation is the complexity of modeling high-energy particles in electric propulsion systems. In order to accurately capture the full spectrum of high-energy particle behavior requires the implementation of additional set of transition rules.

Future research will focus on the integration of refined physics-based numerical schemes, aimed at enhancing the precision in predicting high-energy particle phenomena. A detailed sensitivity analysis is also planned to evaluate the influence of various transition rules on particle velocity at the exhaust and the ionization rate within the simulation domain. This analysis is expected to yield critical insights for refining particle dynamics modeling in electric propulsion. Moreover, the scope of future studies will include the formulation of a broader spectrum of simulation conditions, showcasing the flexibility and adaptability of the AutomataScales method across a range of designs.

For researchers, these findings offer a comprehensive understanding of the AutomataScales method's computational capability. The analyses guide users in selecting appropriate grid resolutions, layer settings and resolution, and parameters for their specific simulation needs. Moreover, this research could support researchers, scientists, teachers and students who have limited computing resources to create and investigate their physics phenomena at the early stage of system design.

REFERENCES

[1] Nguyen van Sang, D., “Numerical Simulation of the Electric Propulsion’s Effects on the Spacecraft.”

[2] Miller, S. A., and Martinez-Sanchez, M., “Two-Fluid Nonequilibrium Simulation of Hydrogen Arcjet Thrusters,” *Journal of Propulsion and Power*, Vol. 12, No. 1, 1996, pp. 112–119. https://doi.org/10.2514/3.23998

[3] Wirz, R., and Katz, I., “2-D Discharge Chamber Model for Ion Thrusters,” 2004. https://doi.org/10.2514/6.2004-4107

[4] Jugroot, M., and Harvey, J. K., “Numerical Modeling of Neutral and Charged Particles within a Gridded Ion Thruster,” 2001.

[5] Wirz, R., “Computational Modeling of a Miniature Ion Thruster Discharge,” 2005. https://doi.org/10.2514/6.2005-3887

[6] Boyd, I. D., “Simulation of Electric Propulsion Thrusters,” MICHIGAN UNIV ANN ARBOR, 2011.

[7] Mahalingam, S., and Menart, J. A., “Computational Study of Primary Electron Confinement by Magnetic Fields in the Discharge Chamber of an Ion Engine,” *Journal of Propulsion and Power*, Vol. 23, No. 1, 2007, pp. 69–72. https://doi.org/10.2514/1.18366

[8] Cheng, S. Y., and Martinez-Sanchez, M., “Hybrid Particle-in-Cell Erosion Modeling of Two Hall Thrusters,” *Journal of Propulsion and Power*, Vol. 24, No. 5, 2008, pp. 987–998. https://doi.org/10.2514/1.36179

[9] Devaux, S., and Manfredi, G., “Vlasov Simulations of Plasma-Wall Interactions in a Magnetized and Weakly Collisional Plasma,” *Physics of Plasmas*, Vol. 13, No. 8, 2006, p. 083504. https://doi.org/10.1063/1.2244533

Decisions," 2023. https://doi.org/10.1115/DETC2023-112481

[10] Megli, T. W., Krier, H., and Burton, R. L., "Plasmadynamics Model for Nonequilibrium Processes in N2/H2 Arcjets," *Journal of Thermophysics and Heat Transfer*, Vol. 10, No. 4, 1996, pp. 554–562. https://doi.org/10.2514/3.829

[11] Pezzi, O., Servidio, S., Perrone, D., Valentini, F., Sorriso-Valvo, L., Greco, A., Matthaeus, W. H., and Veltri, P., "Velocity-Space Cascade in Magnetized Plasmas: Numerical Simulations," *Physics of Plasmas*, Vol. 25, No. 6, 2018, p. 060704. https://doi.org/10.1063/1.5027685

[12] Perrone, D., Passot, T., Laveder, D., Valentini, F., Sulem, P. L., Zouganelis, I., Veltri, P., and Servidio, S., "Fluid Simulations of Plasma Turbulence at Ion Scales: Comparison with Vlasov-Maxwell Simulations," *Physics of Plasmas*, Vol. 25, No. 5, 2018, p. 052302. https://doi.org/10.1063/1.5026656

[13] Servidio, S., Valentini, F., Perrone, D., Greco, A., Califano, F., Matthaeus, W. H., and Veltri, P., "A Kinetic Model of Plasma Turbulence," *Journal of Plasma Physics*, Vol. 81, No. 1, 2015, p. 325810107. https://doi.org/10.1017/S0022377814000841

[14] Kravchenko, D., Lovtsov, A., and Madeev, S., "Development of 5-Cm Ion Thruster and Preliminary Numerical Simulation of Plasma in Its Discharge Chamber," 2019. https://doi.org/10.1063/1.5135485

[15] Giordano, D., "Hypersonic-Flow Governing Equations with Electromagnetic Fields," 2002. https://doi.org/10.2514/6.2002-2165

[16] Murray, M. D., Tutty, R. O., and Gabriel, B. S., "Numerical Modelling of the Gas Flow in an Ion Thruster Hollow Cathode," Vol. 398, 1997, pp. 217.

[17] Alves, E. P., and Fiuza, F., "Data-Driven Discovery of Reduced Plasma Physics Models from Fully Kinetic Simulations," *Physical Review Research*, Vol. 4, No. 3, 2022, p. 033192. https://doi.org/10.1103/PhysRevResearch.4.033192

[18] Ladický, L., Jeong, S., Solenthaler, B., Pollefeys, M., and Gross, M., "Data-Driven Fluid Simulations Using Regression Forests," *ACM Transactions on Graphics*, Vol. 34, No. 6, 2015, pp. 1–9. https://doi.org/10.1145/2816795.2818129

[19] Karyakin, A., and Salem, K., "An Analysis of Memory Power Consumption in Database Systems," 2017. https://doi.org/10.1145/3076113.3076117

[20] Vogelsang, T., "Understanding the Energy Consumption of Dynamic Random Access Memories."

[21] Princeton Research Computing, "Choosing the Number of Nodes, CPU-Cores and GPUs," 2024. Retrieved 16 May 2024. https://researchcomputing.princeton.edu/support/knowledge-base/scaling-analysis

[22] Charoy, T., "Numerical Study of Electron Transport in Hall Thrusters," 2020.

[23] Chaisiriroj, P., and Stone, R. B., "Emerging Methods for 2D and 3D Physics Simulation to Support Early Design Decisions," 2023. https://doi.org/10.1115/DETC2023-112481

[24] Idelsohn, S. R., Oñate, E., and Becker, P., "Particle Methods in Computational Fluid Dynamics," *Encyclopedia of Computational Mechanics Second Edition*, John Wiley & Sons, Ltd, Chichester, UK, 2017, pp. 1–41. https://doi.org/10.1002/9781119176817.ecm2113

[25] Hedjripour, A. H., Callaghan, D. P., and Baldock, T. E., "Generalized Transformation of the Lattice Boltzmann Method for Shallow Water Flows," *Journal of Hydraulic Research*, Vol. 54, No. 4, 2016, pp. 371–388. https://doi.org/10.1080/00221686.2016.1168881

[26] Taccogna, F., Cichocki, F., Eremin, D., Fubiani, G., and Garrigues, L., "Plasma Propulsion Modeling with Particle-Based Algorithms," *Journal of Applied Physics*, Vol. 134, No. 15, 2023. https://doi.org/10.1063/5.0153862

[27] Birdsall, C. K., "Particle-in-Cell Charged-Particle Simulations, plus Monte Carlo Collisions with Neutral Atoms, PIC-MCC," *IEEE Transactions on Plasma Science*, Vol. 19, No. 2, 1991, pp. 65–85. https://doi.org/10.1109/27.106800

[28] Liu, W., Cai, G., Zhang, J., Xia, G., and Wang, W., "Numerical Investigation of Plasma Behavior in a Micro DC Ion Thruster Using the Particle-in-Cell/Monte Carlo Collision (PIC/MCC) Method," *Journal of Physics D: Applied Physics*, Vol. 54, No. 44, 2021, p. 445202. https://doi.org/10.1088/1361-6463/ac1a0c

[29] Yu, K., and Samulyak, V., "SPACE Code for Beam-Plasma Interaction," *Proceedings of IPAC*, 2015.

[30] Mahalingam, S., "Particle Based Plasma Simulation for an Ion Engine Discharge Chamber," Wright State University, 2007.

[31] Arber, T. D., Bennett, K., Brady, C. S., Lawrence-Douglas, A., Ramsay, M. G., Sircombe, N. J., Gillies, P., Evans, R. G., Schmitz, H., Bell, A. R., and Ridgers, C. P., "Contemporary Particle-in-Cell Approach to Laser-Plasma Modelling," *Plasma Physics and Controlled Fusion*, Vol. 57, No. 11, 2015, p. 113001. https://doi.org/10.1088/0741-3335/57/11/113001

[32] Wirz, R. E., "Discharge Plasma Processes of Ring-Cusp Ion Thrusters," California Institute of Technology, 2005.

[33] Bastrakov, S., Donchenko, R., Gonoskov, A., Efimenko, E., Malyshev, A., Meyerov, I., and Surmin, I., "Particle-in-Cell Plasma Simulation on Heterogeneous Cluster Systems," *Journal of Computational Science*, Vol. 3, No. 6, 2012, pp. 474–479. https://doi.org/10.1016/j.jocs.2012.08.012

[34] Deluzet, F., Fubiani, G., Garrigues, L., Guillet, C., and Narski, J., "Sparse Grid Reconstructions for Particle-In-Cell Methods," *ESAIM: Mathematical Modelling and Numerical Analysis*, Vol. 56, No. 5, 2022, pp. 1809–1841. https://doi.org/10.1051/m2an/2022055

[35] Garrigues, L., Tezenas, B., Montcel, D., Fubiani, G., Bertomeu, F., Deluzet, F., Narski, J., Garrigues, L.,

Tezenas Du Montcel, B., and Narski, J., “Application of Sparse Grid Combination Techniques to Low Temperature Plasmas Particle-in-Cell Simulations. I. Capacitively Coupled Radio Frequency Discharges Application of Sparse Grid Combination Techniques to Low Temperature Plasmas Particle-In-Cell Simulations. Part 1: Capacitively Coupled Radio Frequency Discharges,” *Journal of Applied Physics*, Vol. 2021, No. 15, 2021, p. 153303. https://doi.org/10.1063/5.0044363ï

[36] Garrigues, L., Tezenas, B., Montcel, D., Fubiani, G., Reman, B., Garrigues, L., and Tezenas Du Montcel, B., “Application of Sparse Grid Combination Techniques to Low Temperature Plasmas Particle-In-Cell Simulations. Part 2 Electron Drift Instability in a Hall Thruster Application of Sparse Grid Combination Techniques to Low Temperature Plasmas Particle-In-Cell Simulations. Part 2: Electron Drift Instability in a Hall Thruster,” *Journal of Applied Physics*, Vol. 129, No. 15, 2021. https://doi.org/10.1063/5.0044865ï

[37] Wirz, R. E., “Miniature Ion Thrusters: A Review of Modern Technologies and Mission Capabilities,” 2015.

[38] Mahalingam, S., and Menart, J., “Physical Parametric Studies in an Ion Engine Discharge Chamber Using a PIC-MCC Simulation,” 2008. https://doi.org/10.2514/6.2008-4733

[39] Guo, S. L., Jin, X. L., Lei, L., Zhang, X. Y., Zhu, X. F., Yang, Z. H., and Li, B., “Numerical Simulation of Plasma and Electromagnetic Characteristics for an Ion Thruster Discharge Chamber,” *Physics of Plasmas*, Vol. 26, No. 7, 2019, p. 073502. https://doi.org/10.1063/1.5075535

[40] Kao, A., Krastins, I., Alexandrakis, M., Shevchenko, N., Eckert, S., and Pericleous, K., “A Parallel Cellular Automata Lattice Boltzmann Method for Convection-Driven Solidification,” *JOM*, Vol. 71, No. 1, 2019, pp. 48–58. https://doi.org/10.1007/s11837-018-3195-3

[41] Krzyzanowski, M., and Svyetlichnyy, D., “A Multiphysics Simulation Approach to Selective Laser Melting Modelling Based on Cellular Automata and Lattice Boltzmann Methods,” *Computational Particle Mechanics*, Vol. 9, No. 1, 2022, pp. 117–133. https://doi.org/10.1007/s40571-021-00397-y

[42] Vick, B., “Multi-Physics Modeling Using Cellular Automata,” *COMPLEX SYSTEMS-CHAMPAIGN-*, Vol. 17, No. 1/2, 2007, p. 65.

[43] Applied Computational Electromagnetics Society, and Institute of Electrical and Electronics Engineers, “2023 International Applied Computational Electromagnetics Society Symposium (ACES) : March 26-30, 2023, Monterey, California, USA,” 2023.

[44] Dalle, D. J., and Fidkowski, K. J., “Multifidelity Airfoil Shape Optimization Using Adaptive Meshing,” 2014. https://doi.org/10.2514/6.2014-2996

[45] Noever, D., and Hyams, S., “Physical Systems Modeled Without Physical Laws,” 2022.

[46] Le, Q. T., and Ooi, C., “Surrogate Modeling of Fluid Dynamics with a Multigrid Inspired Neural Network Architecture,” *Machine Learning with Applications*, Vol. 6, 2021, p. 100176. https://doi.org/10.1016/j.mlwa.2021.100176

[47] Müller, T., Evans, A., Schied, C., and Keller, A., “Instant Neural Graphics Primitives with a Multiresolution Hash Encoding,” *ACM Transactions on Graphics*, Vol. 41, No. 4, 2022. https://doi.org/10.1145/3528223.3530127

[48] Kobbelt, L., Campagna, S., Vorsatz, J., and Seidel, H.-P., “Interactive Multi-Resolution Modeling on Arbitrary Meshes,” 1998.

[49] COMSOL Multiphysics®, “Particle Tracing Module,” 2024. Retrieved 16 May 2024. https://www.comsol.com/particle-tracing-module

[50] Castedo, R., Reifarth, C., Santos, A. P., Losada, J., López, L. M., Chiquito, M., and Mancilla, J. M., “Application of Grid Convergence Index to Shock Wave Validated with LS-DYNA and ProsAir,” *Ingenieria e Investigacion*, Vol. 39, No. 3, 2019, pp. 20–26. https://doi.org/10.15446/ing.investig.v39n3.81380

[51] Baker, N., Kelly, G., and O’Sullivan, P. D., “A Grid Convergence Index Study of Mesh Style Effect on the Accuracy of the Numerical Results for an Indoor Airflow Profile,” *International Journal of Ventilation*, Vol. 19, No. 4, 2020, pp. 300–314. https://doi.org/10.1080/14733315.2019.1667558

[52] Liu, S., Xu, Z., and Liu, L., “Grid Independence Validation and Numerical Simulation of an Opposed-Piston Free-Piston Engine,” 2021. https://doi.org/10.4271/2021-01-5104

