Immersed Finite Element Particle-In-Cell Simulations of Ion Propulsion

Raed I. Kafafy

Dissertation submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Aerospace Engineering

Joseph Wang, Chair
Slimane Adjred
Tao Lin
Calvin Ribbens
Joseph Schetz

September 1, 2005
Blacksburg, Virginia

Keywords: Electric Propulsion, Ion Thruster, Finite Element, Particle-in-Cell
Copyright 2005, Raed I. Kafafy
A new particle-in-cell algorithm was developed for plasma simulations involving complex boundary conditions. The new algorithm is based on the three-dimensional immersed finite element method which is developed in this dissertation, and a modified legacy particle-in-cell code. The model also applies a new meshing technique that separates the field solution mesh from the particle pushing mesh in order to increase the computational efficiency of the model.

The new simulation model is used in two applications of great importance to the development of ion propulsion technology: the ion optics performance and the interaction between spacecraft and the ion thruster. The first application is ion optics simulations. Simulations are performed to investigate ion optics plasma flow for a whole subscale NEXT ion optics. The operating conditions modeled cover the entire cross-over to perveance limit range. The results of the ion optics simulations demonstrated good agreement with the available experimental data. The second application is ion thruster plume simulations. Simulations are performed to investigate ion thruster plume - spacecraft interactions for the Dawn spacecraft. Plume induced contaminations on the solar array are studied for a variety of ion thruster configurations including multiple thruster firings.
Dedication

To my Parents ...
Acknowledgments

Praise is due to Allah, the Cherisher and Sustainer of the Worlds.

First, I would like to thank my advisor, Prof. Joseph Wang from the Aerospace and Ocean Engineering Department of Virginia Tech, for all his teaching, guidance and support throughout this dissertation. His insight and comprehensive knowledge of Plasma Physics and the art of Particle Simulation provided me a great opportunity of learning.

I would like also to thank Prof. Tao Lin from the Department of Mathematics of Virginia Tech, for his invaluable advices and patience to teach me the art of Finite Element Analysis.

I am also grateful to the other members of my committee: Prof. Joseph Schetz from the Aerospace and Ocean Engineering Department, Prof. Slimane Adjerid from the Department of Mathematics, and Prof. Calvin Ribbens from the Department of Computer Science for their valuable advices and serving on my dissertation committee.

Furthermore, I would like to acknowledge my colleagues at the Computational Advanced Propulsion Laboratory (CAPLab) of Virginia Tech for their valuable assistance and discussions.

Finally, I would like to thank my wife and daughter for their patience, understanding, and inspiration throughout this dissertation.

This work was supported by assistantships from the Department of Aerospace and Ocean Engineering at Virginia Tech and by the Air Force Research Laboratory (AFRL) through a grant from ERC, Inc.
# Contents

1 Introduction .......................................................................................................................... 1  
1.1 Introduction ........................................................................................................................ 1  
1.2 Background ........................................................................................................................ 1  
1.2.1 Electric Propulsion ........................................................................................................ 2  
1.2.2 Ion Propulsion ............................................................................................................. 3  
1.2.3 Ion Thruster Lifetime and Failure Modes ..................................................................... 7  
1.2.4 Spacecraft-Ion Thruster Interaction and Integration Problems ................................. 8  
1.3 Modeling Ion Propulsion .................................................................................................... 9  
1.4 Motivation .......................................................................................................................... 10  
1.5 Research Objectives .......................................................................................................... 11  
1.6 Organization of the Dissertation ....................................................................................... 11  

2 Literature Review .................................................................................................................. 13  
2.1 Introduction ....................................................................................................................... 13  
2.2 Previous Work On Ion Optics Simulation – Review of the Last Decade ...................... 14  
2.3 Previous Work On Spacecraft–Ion Thruster Plume Interaction – Review of the Last Decade ................................................................................................................................. 17  
2.4 Review On Methods To Solve A Field Problem With Complex Boundaries ............... 21
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4.1</td>
<td>Body-Fitting-Grid Methods</td>
<td>22</td>
</tr>
<tr>
<td>2.4.2</td>
<td>Cartesian-Grid Methods Based On Finite Difference Discretization</td>
<td>23</td>
</tr>
<tr>
<td>2.4.3</td>
<td>Cartesian-Grid Methods Based On Finite Element Discretization</td>
<td>25</td>
</tr>
<tr>
<td>2.5</td>
<td>Contribution</td>
<td>27</td>
</tr>
<tr>
<td>3</td>
<td>A Three-Dimensional Immersed Finite Element Method</td>
<td>29</td>
</tr>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>The Interface Boundary Value Problem</td>
<td>29</td>
</tr>
<tr>
<td>3.3</td>
<td>Weak Formulation Of The Field Problem</td>
<td>31</td>
</tr>
<tr>
<td>3.4</td>
<td>A Three–Dimensional IFE Space</td>
<td>32</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Intersection Topology</td>
<td>34</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Special Intersection Topology</td>
<td>34</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Linear Local Nodal FE Basis Functions</td>
<td>36</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Linear Local Nodal IFE Basis Functions</td>
<td>36</td>
</tr>
<tr>
<td>3.4.5</td>
<td>Existence and Uniqueness</td>
<td>43</td>
</tr>
<tr>
<td>3.4.6</td>
<td>Partition of Unity and Consistency with Classical FEM</td>
<td>45</td>
</tr>
<tr>
<td>3.5</td>
<td>Numerical Experiments</td>
<td>47</td>
</tr>
<tr>
<td>3.5.1</td>
<td>An Interface Problem With a Spherical Interface</td>
<td>48</td>
</tr>
<tr>
<td>3.5.2</td>
<td>An Interface Problem With a Hemispherical Interface</td>
<td>50</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Numerical Error Analysis</td>
<td>53</td>
</tr>
<tr>
<td>4</td>
<td>Three-Dimensional IFE Field Solver</td>
<td>58</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>58</td>
</tr>
<tr>
<td>4.2</td>
<td>Mesh Generation</td>
<td>58</td>
</tr>
<tr>
<td>4.3</td>
<td>Mesh-Object Intersection</td>
<td>59</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Intersection Topology Classification</td>
<td>59</td>
</tr>
</tbody>
</table>
4.4 Assembly of the IFE System
4.4.1 Local Assembler
4.4.2 Global Assembler
4.4.3 Integration Rules
4.4.4 Sparse Storage of the System Matrix

4.5 Solution of the Nonlinear Field Problem
4.5.1 Gauss-Seidel Iteration
4.5.2 Newton-Raphson Iteration

4.6 Solution of the Sparse Linear/Linearized System
4.6.1 Preconditioned-Conjugate Gradient (PCCG) Solver
4.6.2 Preconditioners

4.7 Hardwiring The IFE Field Solver

5 The Hybrid-Grid IFE-PIC Model
5.1 Introduction
5.2 Theory
5.2.1 Plasma
5.2.2 Debye Shielding and Plasma Sheath
5.3 The Particle–In–Cell Method
5.3.1 Particle Push
5.3.2 Charge Deposit
5.3.3 Field Solve
5.3.4 Force Weighting
5.4 Particles Initial and Boundary Conditions
5.4.1 Particles Loading
5.4.2 Particles Injection
6.4.1 Simulation Domain ................................................. 99
6.4.2 Simulation Algorithm ........................................... 100
6.4.3 Streamline PIC Simulation ..................................... 103

6.5 NEXT Ion Optics ..................................................... 105

6.6 Simulation Domain Layout ........................................ 106

6.7 Standard HG-IFE-PIC Simulation ................................. 107
   6.7.1 Simulation Setup ............................................. 108
   6.7.2 Computational Performance ................................. 109
   6.7.3 Beamlet Behavior ........................................... 110
   6.7.4 Impingement Current ....................................... 113

6.8 Streamline HG-IFE-PIC Simulation: Two-Quarter Aperture ...... 114
   6.8.1 Simulation Setup ............................................. 114
   6.8.2 Computational Performance ................................. 114
   6.8.3 Beamlet Behavior ........................................... 115
   6.8.4 Impingement Current Limits ................................. 118
   6.8.5 Electron Backstreaming Limit .............................. 118

6.9 Streamline HG-IFE-PIC Simulation: Whole Subscale Gridlet ...... 120
   6.9.1 Simulation Setup ............................................. 120
   6.9.2 Computational Performance ................................. 122
   6.9.3 Beamlet Plasma Flow ...................................... 123
   6.9.4 Impingement Current Limits ................................. 125
   6.9.5 Electron Backstreaming Limit .............................. 129

6.10 Summary and Conclusion .......................................... 130

7 Ion Thruster Plume Simulations ...................................... 133
   7.1 Introduction .................................................... 133
7.2 Spacecraft Contamination ............................................. 134
  7.2.1 Modeling of Contamination ..................................... 134
  7.2.2 Measuring Molecular Film Thickness ............................ 136
7.3 Modelling of Ion Thruster Plume ........................................ 136
  7.3.1 Ion Beam .......................................................... 137
  7.3.2 Neutral Propellant Plume ......................................... 139
  7.3.3 Charge-Exchange Ions ............................................ 140
  7.3.4 Non-Propellant Efflux ........................................... 140
  7.3.5 Electrons .......................................................... 141
  7.3.6 Electrostatic Field ............................................... 142
7.4 Normalization .......................................................... 142
7.5 Simulation Model ....................................................... 143
  7.5.1 Simulation Domain ................................................ 143
  7.5.2 Simulation Algorithm ............................................. 143
  7.5.3 Deposit Calculation ............................................... 145
7.6 Dawn Spacecraft ....................................................... 146
  7.6.1 Spacecraft ........................................................ 146
  7.6.2 Ion Thruster ....................................................... 147
7.7 Spacecraft Model ....................................................... 148
7.8 Simulation Domain ..................................................... 149
7.9 Simulation Setup ....................................................... 150
7.10 Results: CASE 0 ......................................................... 151
  7.10.1 Plasma Diagnosis ................................................ 152
  7.10.2 Deposition Diagnosis ............................................ 153
7.11 Results: CASE 1 ......................................................... 153
  7.11.1 Plasma Diagnosis ................................................ 156
7.11.2 Deposition Diagnosis

7.12 Results: CASE 2

7.12.1 Plasma Diagnosis

7.12.2 Deposition Diagnosis

7.13 Summary and Conclusion

8 Conclusions

8.1 Introduction

8.2 Summary of Research

8.3 Contributions to Finite Element Analysis

8.4 Contributions to Plasma Simulation

8.5 Contributions to Ion Optics Modeling

8.6 Contributions to Spacecraft-Ion Thruster Interaction Modeling

8.7 Recommended Future Work
List of Figures

1.1 NSTAR ion thruster. .................................................. 4
1.2 A schematic of an ion thruster. ........................................ 6

3.1 Solution domain of the interface BVP. .......................... 30
3.2 Partitioning of a Cartesian IFE cell. ............................... 32
3.3 The five tetrahedra comprising a Cartesian cell. ............... 33
3.4 Intersection topologies of tetrahedral elements. ............... 35
3.5 An odd intersection topology. ........................................ 35
3.6 Two cases of possible three-edge cut in the reference element $\hat{T}$. ............................................................ 38
3.7 One of the three possible four-edge cut elements in the reference element $\hat{T}$. ............................................................ 41
3.8 Geometry of the spherical interface problem. .................... 49
3.9 Interpolation and solution errors of a spherical-interface problem. ................................................................. 52
3.10 Geometry of the hemispherical interface problem. ............ 53
3.11 Interpolation and solution errors of a hemispherical-interface problem. ............................................................. 56

4.1 Special situations of three-edge cut tetrahedron. .................. 60
4.2 Partitioning of typical interface tetrahedra into sub-tetrahedra. ................................................................. 64
4.3 Sub-tetrahedra in a three-edge cut element. ....................... 65
4.4 Sub-tetrahedra in a four-edge cut element. ...................... 66
6.19 Beamlet plasma normalized potential and ion density at perveance.

The ion density is normalized by the nominal upstream plasma density,
\[ n_0 = 1.0 \times 10^{17} \text{m}^{-3}, \]
and the potential is normalized by \( T_{e0} = 5 \text{ eV} \).

6.20 Impingement current curve for the whole gridlet (seven apertures)
model as compared with the two-quarter aperture (single aperture)
model.

6.21 Impingement current limits.

6.22 Potential profile along aperture centerlines.

6.23 Aperture electron backstreaming.

7.1 Geometry of beam profile.

7.2 Geometry of neutral plume profile.

7.3 Dawn spacecraft layout.

7.4 Geometry of the studied Dawn thruster configurations.

7.5 A 3D view of the IFE mesh. The mesh is cut away to illustrate relative
spacecraft position.

7.6 History of number of particles for Xe\(^+\) CEX ions simulation for CASE
0A. The simulation time is given in units of time steps.

7.7 Plasma properties for CASE 0A. Potential is normalized by 5 eV and
ion charge density is normalized by \( 7.61 \times 10^{10} \text{m}^{-3} \).

7.8 Xe\(^+\) CEX ion trajectories for CASE 0A.

7.9 Deposition flux of Mo atoms for CASE 0A.

7.10 Firing options of CASE 1.

7.11 History of number of particles for Xe\(^+\) CEX ions simulation for CASE
1 with all possible firing options. The simulation time is given in units
of time steps.

7.12 Plasma properties for CASE 1A. Potential is normalized by 5 eV and
ion charge density is normalized by \( 7.61 \times 10^{10} \text{m}^{-3} \).

7.13 Xe\(^+\) CEX ion trajectories for CASE 1A.
List of Tables

1.1 Specific impulse of current flight propulsion systems. ............................................. 3

3.1 $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10$. ................................. 50

3.2 $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10,000$. ................................. 50

3.3 $L^\infty$, $L^2$ and $H^1$ errors of the IFE solutions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10$. .................................................. 51

3.4 $L^\infty$, $L^2$ and $H^1$ errors of the IFE solutions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10,000$. ............................................ 51

3.5 $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10$. .................................................. 54

3.6 $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10,000$. ............................................ 54

3.7 $L^\infty$, $L^2$ and $H^1$ errors of the IFE solutions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10$. .................................................. 54

3.8 $L^\infty$, $L^2$ and $H^1$ errors of the IFE solutions generated with partitions of decreasing size $h$ and $\beta^+/\beta^- = 10,000$. ............................................ 55

3.9 Regression constants of the relation between interpolation error and mesh size. .......................................................... 55

3.10 Regression constants of the relation between IFE solution error and mesh size. .......................................................... 57
4.1 Rules for classification of intersection topologies ........................................ 61
4.2 Weights and quadrature points for integrations on tetrahedral elements .......... 63

6.1 Reference and normalized Variables ......................................................... 98
6.2 Nominal dimensions the ion optics ........................................................... 106
6.3 Nominal throttling condition of the ion optics ........................................... 106
6.4 PIC and IFE meshes used in the standard HG-IFE-PIC simulation ............... 109
6.5 Beamlet and impingement currents at typical plasma conditions as estimated by the standard HG-IFE-PIC model ........................................... 113
6.6 PIC and IFE meshes used in the streamline HG-IFE-PIC two-quart aperture simulation ................................................................. 114
6.7 PIC and IFE meshes used in the streamline HG-IFE-PIC whole gridlet simulation as a function of the upstream plasma density .................................. 122

7.1 Maximum and average deposition rates of Mo\(^+\) CEX ions on the solar array .......................................................... 169
Chapter 1

Introduction

1.1 Introduction

This chapter introduces the reader to the dissertation. It first provides the necessary background that is a prerequisite to go through this work. The concept of electric propulsion is introduced with ion propulsion, the focus of this research, described in more detail. The major research problems in ion propulsion development are briefly addressed. The state-of-the-art of modeling ion propulsion taking into consideration these problems is also reviewed. Our motivation for this research work and the objectives of the research are then stated. Finally, an outline of the whole dissertation is given.

1.2 Background

Despite the early introduction of the notion of electric propulsion, which can be traced back to as early as 1906, and the relatively early maturity of some electric propulsion concepts, electric propulsion is still in the early stages of application. To date, almost 200 solar-powered satellites in Earth orbits and a handful of spacecraft beyond Earth’s gravitational influence have benefited from the mass savings engendered by electric propulsion [13]. NASA’s Deep Space One (DS1), which was launched on October 24th 1998, is the first interplanetary spacecraft to utilize electric propulsion as primary propulsion. The success of the DS1 mission has paved the
road for ion propulsion technology to be applied in future NASA missions [59].
In the following, we will present a brief background on electric propulsion with emphasis on ion propulsion.

1.2.1 Electric Propulsion

Historically, the fundamental notion of electric propulsion was first introduced in 1911 by Tsiolkovsky which is described in his own words [13]:

\[
\text{It is possible that in time we may use electricity to produce a large velocity for the particles ejected from a rocket device.}
\]

However, the classical definition that is currently accepted within the electric propulsion community, was given by Prof. Robert Jahn in his classical textbook [30]:

\[
\text{the acceleration of gases for the purpose of producing propulsive thrust by electric heating, electric body forces, and/or electric and magnetic body forces.}
\]

By definition, electric propulsion relies on an external power source to obtain acceleration. This is the major distinction between electric propulsion and chemical propulsion, which primarily depends on the internal energy in the molecular bonds of the propellant to obtain acceleration. Electric propulsion is superior to chemical propulsion for many space mission applications because of its much higher specific impulse [70]. Specific impulses of over 17,000 s have been demonstrated in the laboratory. On the contrary, the dependence of chemical propulsion on the propellant internal energy limits the maximum specific impulse to typically about 450 s. A comparison of the specific impulse of current flight propulsion systems is given in table 1.1 [72, 45]. Electric propulsion systems may be categorized as [30]:

**Electrothermal Propulsion** acceleration of a propellant gas by electrical heat addition and expansion through a convergent/divergent nozzle. Examples include resistojets and arcjets.

**Electrostatic Propulsion** acceleration of an ionized propellant gas by the application of electric fields. Examples include gridded ion thrusters, colloid thrusters, and field emission electric propulsion (FEEP).
Propulsion System | Specific Impulse [s]  
---|---  
Monopropellant hydrazine | 330  
Bipropellant thruster | 450  
Resistojet | 300  
Arcjet | 500  
Hall thruster | 1600  
Ion thruster | 2800  
PPT | 1000  

Table 1.1: Specific impulse of current flight propulsion systems.

Electromagnetic propulsion acceleration of an ionized propellant gas by the application of both electric and magnetic fields. Examples include Hall thrusters, pulsed plasma thrusters (PPT), and magnetoplasma dynamic thrusters (MPDT).

### 1.2.2 Ion Propulsion

Ion propulsion has been under development since the 1950’s. The first ion thruster in the US was built by Dr. Harold Kaufman at NASA Glenn in 1959. The first flight tests of NASA’s ion thrusters were conducted in the 1960’s through a program called Space Electric Rocket Test (SERT). In 1964, a pair of NASA Glenn ion thrusters were launched on SERT 1 mission from Wallops Island, VA. One of the two thrusters onboard did not work, but the other operated for 31 minutes. SERT 2 carried two ion thrusters, one operated for more than five months and the other for nearly three months.

Mercury and cesium were commonly used as propellants in many early ion thrusters because of their large atomic weight and low ionization energy. SERT 1 carried one mercury and one cesium engine, and SERT 2 had two mercury engines. Regardless of the propellant, these early ion thrusters applied the same ionization and acceleration mechanism as the more recent NSTAR ion thruster which used xenon as a propellant. Despite the favorable features of mercury and cesium as propellants they were excluded from operation because of adverse contamination effects. At room temperature, mercury is a liquid and cesium is a solid; both must be heated to turn them into gases. After exiting the ion thruster, many mercury or cesium atoms would cool and condense on the spacecraft surfaces causing contamination. Hence, modern ion
thrusters use inert gases as propellants. The majority of them use xenon which is a chemically inert, colorless, odorless, and tasteless, noble gas.

The first xenon ion thruster ever flown was a Hughes engine launched in 1979 on the Air Force Geophysics Laboratory’s Spacecraft Charging at High Altitude (SCATHA) satellite. The first commercial use of a xenon ion thruster was on PanAmSat 5 (PAS-5), a communications satellite launched in August 1997, on a Russian Proton rocket from the Baikonur Cosmodrome in Kazakhstan. These ion thrusters were used for orbit maintenance and station keeping. Ion thrusters for such purposes are smaller than those designed to act as the primary propulsion system during interplanetary missions.

In the early 1990’s, NASA initiated the NASA Solar Electric Power Technology Application Readiness (NSTAR) project to develop xenon ion thrusters for deep space missions. The engineering space model of the NSTAR thruster successfully logged more than 22,000 hours (three years) of operation in a vacuum chamber at JPL. The NSTAR thruster, like the one shown in figure 1.1, was used as the primary propulsion system on the Deep Space One spacecraft.

On the other coast of the Atlantic, research and development of ion thrusters proceeded in the United Kingdom in two phases. The first phase, from the 1960’s to 1975, resulted in a prototype engine producing 10 mN, in which mercury was used as a propellant. It was proposed to be used for station keeping on Olympus, a large
research communications satellite. However, due to financial obstacles, the engine did not acquire sufficient laboratory life testing to be flight ready. In the mid 1980’s, work resumed on a new version, which was named UK-10. In UK-10, xenon replaced mercury as a propellant, in order to reduce risks of contamination.

UK-10 is a light weight focused ion beam engine with a 10 centimeter diameter beam. It was developed by the Defence Research Agency at Farnborough, and Matra Marconi Space, with a grant from ESA, and has now had over two thousand hours of life testing, in daily 3 hour pulses, in preparation for space qualification on board the ESA ARTEMIS satellite. UK-10 can achieve up to 70 mN, although it is most efficient at 25 mN.

The larger UK-25 ion thruster has achieved a thrust of 260 mN in tests to date. The thrust could probably be extended to 500 mN. These could be used to send fast probes beyond the outer solar system, to near interstellar space, with Delta class rocket’s, at costs within the reach of Discovery class missions now being flown to Mars and the asteroids [23].

Compared with chemical propulsion, the application of ion propulsion in orbit maintenance and station keeping on satellites can lead to a reduction in the propellant utilization by a factor of ten. Considerable gains in payload mass and/or orbital lifetime can be achieved. An example is a low mass, low cost Earth observation satellite which can be orbited at few hundred kilometers, for a lifetime of several years. Launching costs could be greatly reduced by the use of a Pegasus XL launcher followed by orbit raising instead of inserting the satellite directly into orbit. Even more dramatically, the 370 kg Ulysses solar polar satellite, launched in 1989, required an Inertial Upper Stage / PAM combination weighing 20 tons, to place it into orbit. With an ion thruster cluster, the same spacecraft would have required 2.4 tons of engine/propellant, and could thus have been launched by an Ariane 4. It has been calculated that, if the International Space Station were to use ion thrusters for orbit maintenance rather than chemical rockets, about 9 tons of propellant mass per year would be saved which is a considerable cost saving.

In a conventional ion thruster, as schematically illustrated in figure 1.2, the propellant is injected into the discharge chamber to be ionized. The conventional method of ionization is called electron bombardment, in which a high-energy electron (negative charge) collides with a propellant atom (neutral charge) to release a second electron, yielding two negative electrons and one positive ion. This ionization process, in a xenon ion thruster, is simply described as:

\[ e^- + Xe^0 \rightarrow Xe^+ + 2e^- \]
Electrons are generated by a hollow cathode, called the discharge cathode, located at the center of the engine on the upstream end. The electrons flow out of the discharge cathode and are attracted to the discharge chamber walls, which are charged to a high positive potential by the thruster’s power supply. High-strength magnets are placed along the discharge chamber walls so that as electrons approach the walls, they are redirected into the discharge chamber by the magnetic fields. By maximizing the time that electrons and propellant atoms remain in the discharge chamber, the chance of ionization is maximized and hence the efficiency of the ionization process.

An alternative method of ionization called electron cyclotron resonance (ECR) is also being researched at NASA. This method uses high-frequency radiation (usually microwaves), coupled with a high magnetic field to heat the electrons in the propellant atoms, causing them to break free of the propellant atoms, creating plasma. Ions can then be extracted from this plasma.

The ions produced in the discharge chamber are accelerated by electrostatic forces. The electric fields used for acceleration are generated by electrodes positioned at the downstream end of the thruster. Each set of electrodes, called ion optics or grids, contains thousands of coaxial apertures. Each set of apertures acts as a lens that electrically focuses ions through the optics. NASA’s ion thrusters use a two-electrode system, where the upstream electrode (called the screen grid) is charged highly posi-
itive, and the downstream electrode (called the accelerator grid, or accel grid) is charged highly negative. Since the ions are generated in a region of high positive potential and the accelerator grid’s potential is negative, the ions are attracted toward the accelerator grid and are focused out of the discharge chamber through the apertures, creating thousands of ion jets. The stream of all the ion jets together is called the ion beam, whereas the stream of an individual ion jet is called a beamlet.

The exhaust velocity of the ions in the beam is based on the voltage applied to the optics. While a chemical rocket’s top speed is limited by the thermal capability of the rocket nozzle, the ion thruster’s top speed is limited by the voltage that is applied to the ion optics.

Efficiency and thrust are determined by ionization voltage from anode to cathode, and by propellant feed rate. Typically, ionization voltages of over 40 volts lead to erosion of the thrust chamber and reduced life span, while fuel utilization rates of 6 milligrams per second produce 25 mN, within a field of 1100 volts.

Because the ion thruster expels a large amount of positive ions, an equal amount of negative charge must be expelled to keep the total charge of the exhaust beam neutral. A second hollow cathode called the neutralizer is located on the downstream perimeter of the thruster and expels the needed electrons [48].

1.2.3 Ion Thruster Lifetime and Failure Modes

The lifetime of an ion thruster is primarily limited by the erosion of thruster components, especially the ion optics [37]. Currently, the ion optics grids of nearly all ion thrusters are made of molybdenum (Mo). The use of carbon-based ion optics (CBIO) has been shown to sufficiently suppress the erosion of the screen electrode as to effectively remove sputter erosion of that electrode as a failure mechanism. However, the reduced erosion rates of the accel electrode will remain one of the principal thruster life-limiters.

Charge-exchange (CEX) ions play a profound role in the erosion of the accel grid. CEX ions result from the CEX collisions between fast beam ions and slow propellant neutrals according to the following reaction

\[
\text{Xe}_{\text{slow}} + \text{Xe}^+_{\text{fast}} \rightarrow \text{Xe}^+_{\text{slow}} + \text{Xe}_{\text{fast}}
\]

The erosion of the accel electrode by CEX ions can be related to three mechanisms [90]:
• the erosion of the downstream face of the electrode by ions with energies comparable to the accel electrode potential (a few hundred Volts) that form a pit and groove pattern,

• the erosion of the upstream side of the electrode by ions with energies comparable to the total accelerating voltage (up to 10 kV), and

• the erosion of the aperture walls by ions which have energies varying between the total voltage and the accel voltage.

Electrode failure by pit and groove erosion occurs when the grooves wear through the electrode causing structural and/or electrical failure. Electrode failure due to impingement on the upstream surface of the accelerator grid occurs at the onset of electron backstreaming. Either thinning of the electrode or aperture enlargement once the erosion pattern has worn through can lead to electron backstreaming. Failure resulting from aperture enlargement due to the third mechanism can occur either by electron backstreaming or structural failure. Structural failure due to aperture enlargement occurs when the aperture diameter reaches the groove of the pit and groove erosion.

1.2.4 Spacecraft-Ion Thruster Interaction and Integration Problems

Thruster plume is one of the major sources of spacecraft contamination. Thruster exhaust products may backflow towards spacecraft surfaces by several mechanisms. In the case of ion thruster plume, Wang et. al. [80] showed that the CEX ions backflow through an expansion process which is similar to the expansion of mesothermal plasma into vacuum. The electric field established by the CEX plasma around the spacecraft also controls the trajectories of the ionized contaminants.

Contaminants that adhere to the surface can either condense or be absorbed onto the surface. Condensation can be a very serious problem because it easily forms a thick layer on a surface. It is usually avoided on spacecraft surfaces by using materials that emit a very small fraction of volatile condensible material (VCM). After VCMs, the main source of deposition on spacecraft surfaces is adsorption of individual molecules. An adsorbate forms because of surface attraction between individual atoms of the substance and those of the contaminant. The degree of adherence of any individual particle depends on the gas species, the surface temperature, the composition of the
substrate, and the amount of surface coverage. As a monolayer is completed, the likelihood decreases that additional contaminant molecules will stick because they will not see any substrate molecules [24].

The presence of a thin contaminant film on the surface of a material will alter its solar absorptance. The contaminant layer will increase the absorptance of the surface material and consequently its equilibrium temperature [74].

In addition to the concern of contamination of thermal control surfaces, there is also the possibility for contamination buildup on optics or solar arrays. The presence of a contaminant film on a lens, mirror or focal plane will degrade the signal to noise ratio (SNR) of the detector and limit the dynamic range by absorbing light from the target of interest. If the contaminant film becomes too thick, the sensor will cease to function properly [74].

1.3 Modeling Ion Propulsion

The modeling of ion propulsion started a few decades ago even before the development of the first ion thruster prototype during the 1950’s by Harold Kaufman. Simple analytical and/or numerical methods were applied to model the performance of ion thrusters including such useful information as the thrust, current, divergence angle [25]. Also, other simplified and/or numerical studies have been performed to investigate the motion of CEX ions, as well as their effects on both ion thruster grid surfaces and spacecraft critical components as a whole. Despite their success to give a qualitative insight on ion thruster operation such models were incapable of predicting CEX ion sputtering and deposition rates for arbitrary geometries and operating conditions [56].

Starting in the 1990’s, the Particle-In-Cell (PIC) method [7] has been applied to model ion optics as well as ion thruster plume. PIC models are more appropriate for these applications because the mean free path length $\lambda_{mfp}$ for both ion optics and plume problems is very large as compared with the characteristic system dimensions. The mean free path length is given by

$$\lambda_{mfp} = \frac{1}{n\sigma}$$

where $n$ is the plasma density and $\sigma$ is the collision cross-section associated with a certain collision mechanism. For a typical ion thruster, $\sigma$ is in the order of $10^{-20}\text{m}^2$
and \( n \) ranges from \( 10^{16} \text{m}^{-3} \) to \( 10^{18} \text{m}^{-3} \) for the inside plasma and from \( 10^{9} \text{m}^{-3} \) to \( 10^{12} \text{m}^{-3} \) for the outside plasma. This results in \( \lambda_{mfp} \) ranging from \( 10^2 \text{m} \) to \( 10^4 \text{m} \) for the inside plasma as compared with the millimeter dimensions of the ion optics apertures, and from \( 10^8 \text{m} \) to \( 10^{11} \text{m} \) for the outside plasma as compared with the few meters dimensions of the spacecraft.

In a PIC model, real plasma particles are represented by much fewer simulation particles. A typical computational cycle includes four steps: 1) particle push, 2) charge deposit, 3) field solve, and 4) force weigh. The space charge, particle trajectories and electric fields are solved self-consistently. The PIC model applies linear schemes to deposit, or interpolate, the charge of moving particles to the discrete mesh nodes, and weigh, or interpolate, the electric fields from mesh nodes to particle positions. The details of the PIC model will be discussed in chapter 5.

Numerical accuracy and computational efficiency lead to contradicting requirements in the design and implementation of PIC models. The PIC models that are currently used in ion propulsion simulations are either based on finite difference methods or standard finite element method. A detailed literature review will be provided in chapter 2.

1.4 Motivation

Ion propulsion development is increasingly dependent upon inputs from physics based modeling. In order to apply the particle simulation method as a research tool, one needs to build a code that is sophisticated enough so that complex geometric and field effects can be modeled properly, and yet computationally efficient enough so that large-scale particle simulations can be performed routinely within reasonable time. Because of these conflicting requirements for an accurate field solver and a fast particle pusher, the plasma-material interface represents a major challenge in the application of PIC codes for ion optics modeling [35] as well as spacecraft-ion thruster interaction modeling [80].

In this study, we aim at developing a numerically accurate and yet computationally efficient simulation algorithm which we apply to simulate the main physical phenomena occurring during the operation of an ion thruster such as beam extraction and CEX ions and spacecraft contamination by the ion thruster plume species.
1.5 Research Objectives

The objective of this research is to develop a three-dimensional immersed finite element (IFE) method. This method is designed to be capable of retaining second order accuracy while solving an interface boundary value problem (BVP) on a regular, even Cartesian domain. Based on the three-dimensional IFE method, we develop a three-dimensional (IFE) field solver for plasma simulation applications. The most attractive advantage of the new field solver is that it can solve the interface field boundary value problem on a Cartesian-based tetrahedral mesh irrespective of the shape and position of the interface. It can also be used to investigate the effect of involving materials with different dielectric constants since it incorporates explicitly the material properties in the field solution.

Next, we integrate the IFE field solver to a Cartesian-grid PIC code which is modified from a legacy standard PIC code. It is well known that Cartesian-grid PIC codes are very fast in performing particle-mesh interpolations and particle pushing. We further adopt a new meshing technique in particle simulation in which we let the PIC and IFE mesh nodes to be dislocated. This allows us to stretch the IFE mesh according to the local potential gradients and plasma conditions while retaining the ultimate speed of a uniform Cartesian grid PIC code. The target of the new PIC model is large scale electrostatic plasma simulations which involve complex geometries such as those encountered in ion optics and spacecraft-ion thruster interaction problems.

1.6 Organization of the Dissertation

This dissertation is arranged in 8 chapters. The description of these chapters is as follows

Chapter 1 is an introduction in which we introduce the background and motivation for this research work. We also briefly describe our contribution.

Chapter 2 is a literature review of the research work which has been done prior to the current research work. Because of the nature of the current work, the literature review includes three parts: numerical methods to solve the field equation on irregular boundaries, ion optics modeling, and spacecraft-ion thruster interaction modeling.
Chapter 3 is the theoretical development of the three-dimensional immersed finite element (3D IFE) method. In this chapter, we also present a numerical error analysis of the method developed.

Chapter 4 presents the details of the IFE field solver that is used in all IFE–PIC simulations in the current work.

Chapter 5 introduces the Hybrid-Grid Immersed-Finite-Element Particle-In-Cell (HG-IFE-PIC) model. The details of the model are discussed. Numerical experiments showing the approximation capabilities of the model are also presented.

Chapter 6 introduces the physical and mathematical modeling of the NEXT ion optics. It also presents the results of the ion optics simulations performed at selected operating conditions.

Chapter 7 introduces the modeling of the spacecraft-ion thruster plume interactions. It also presents the simulation results obtained for selected spacecraft-ion thruster configurations and firing options.

Chapter 8 contains conclusions and a discussion of the results obtained in the current work. It also summarizes the scientific contributions made to the corresponding disciplines of science and engineering. This chapter also introduces the future research work that is recommended by the author.
Chapter 2

Literature Review

2.1 Introduction

The literature review, herein, is organized into three sub-reviews, to cover the different aspects of this study:

- A review of the ion optics models,
- A review of the spacecraft–ion thruster plume interactions models, and
- A review of the field solution methods for problems involving complex boundaries.

The first section reviews the research work conducted in ion optics simulations during the last decade (1994-2005). The next section reviews the main research work contributing to simulation of spacecraft-ion thruster plume interactions during the last decade as well. Starting in the 1990’s, the Particle-In-Cell (PIC) method \[ \text{[7]} \] has been widely applied to model ion optics as well as spacecraft ion thruster plume interactions. The third section reviews the research work performed to develop numerical methods that are capable of accurately handling elliptic problems with complex boundaries.
2.2 Previous Work On Ion Optics Simulation – Review of the Last Decade

During the last decade, many research work has been conducted to numerically study the ion optics problem. In the following, we introduce a review of most of the work done in a chronological order.

In 1994, Peng et al. [57] developed a particle simulation code to study ion optics and the effect of charge-exchange-induced grid erosion in electron bombardment ion thrusters. The code is based on particle-in-cell (PIC) and direct simulation Monte Carlo (DSMC) methods. Two versions of the code were tested. A two-dimensional axisymmetric code was presented to run on PCs. They also developed a three-dimensional code in which they showed the necessity to calculate the pitted erosion patterns observed in ground tests.

In 1996, Arakawa and Nakano [1] developed a three-dimensional optics code to calculate both beam divergence and ion-sputtering rate to grids due to charge-exchange. In the code, the simulation domain does not extend from the upstream ion sheath region to the downstream plasma. Instead, they assume an emitting surface for the ions. The position and shape of the emitting surface are initially guessed then they are iterated on during computations till convergence using the distance between the lower and higher potentials as obtained by the space-charge-limited current law (Child’s Law). This leads to a reduction in memory storage and computation time by avoiding the meshing of the upstream plasma region in which mesh size is limited by the local Debye length. Therefore, their code can be run within a reasonable time on PCs. Although the code is more efficient concerning the memory storage and computational time, the results obtained depend on the accuracy of the ion trajectories which is degraded by the inaccuracy in predicting the shape and position of the emitting surface.

In 1998, Tartz et al. [73] presented a two-dimensional, axisymmetric simulation code that can be used to tackle a comprehensive optimization of a grid system in a rather short computational time even on a PC. In their approach to the modeling of the broad beam formation and secondary grid-eroding impact of charge-exchange ions, they break down the complex interaction pattern into a series of largely independent processes, leading to a considerably reduced computational effort. The extraction of ions, calculation of potential, and ion trajectories are done self-consistently using IGUN [6]. They also tried to take into account the effect of neighboring holes on the simulated hole by estimating their influence in the two-dimensional simulations
in a spherical approximation assuming a ring-shaped aperture.

In 1999, Muravlev and Shagayda [46] developed a two-dimensional planar and axisymmetric code to be used in the simulation of ion thruster extraction grid performance and erosion process. They presented a simplified approach to predict the erosion pattern without fully three-dimensional modeling. In such approach, they used a standard computational domain with a cross-section of a 30 by 60 degrees triangle. The trajectories of both the charge-exchange ions and downstream plasma ions are calculated with three spatial components. The axial and radial components are determined from the two-dimensional results. The azimuthal component is assumed to be zero. They also studied both circular and slit apertures.

In 1999, Okawa and Takegahara [51] presented a two-dimensional axisymmetric particle simulation code with three-dimensional velocity components to investigate the beam extraction phenomena from a discharge plasma. They treated electrons as particles as well as ions. Though, the simulation was preliminary, they could estimate the position and shape of the plasma sheath boundary.

In 2000, Boyd and Crofton [9] performed a computational study on grid erosion through ion impact. Their model employs a combination of a hybrid fluid-PIC method for the plasma dynamics, and a DSMC method for collision dynamics involving momentum transfer, charge exchange, and Coulomb collisions. In the model, both single and double charge ions are treated as particles. Comparing results with experimental measurements of grid currents for the UK-T5 ion thruster, the model accurately predicts the current collected on the acceleration grid for a range of operating points. However, it significantly under predicts the current collected on the deceleration grid.

In 2001, Nakayama and Wilbur [47] performed a numerical study on a high-specific impulse many-grid ion thruster operated at a voltage of 10 kV. In the study, they used a three-dimensional particle simulation code that employs an energy compensation method, a simplified pre-sheath method, and high-speed coding. In the study, they modified the OPT code which is a two-dimensional cylindrical ion optics code. They also developed the igx code, which is a three-dimensional ion extraction simulation code.

In 2001, Wang et al. [88, 89] developed a fully three-dimensional particle simulation mode for ion optics. The model allows multiple apertures to be included explicitly in the simulation domain and makes no assumptions on the upstream sheath. Simulation results show both qualitative and quantitative agreement with experimentally observed erosion patterns in the NSTAR ion thruster. In the model, the
three-dimensional optics aperture geometry is handled by a method of sub-grid scale placement of boundaries which explicitly includes the location of the optics wall in relation to the grid in the finite difference formulation of the Poisson’s equation.

In 2002, Wang and Lin [86] developed a two-dimensional axisymmetric electric field solver for ion optics modeling. The solver is based upon the two-dimensional, axisymmetric immersed finite element (IFE) method. The IFE method is believed to provide a very promising tool to tackle ion optics simulation problems that involve complex geometries on Cartesian meshes. So, computational speed doesn’t have to be sacrificed in favor of accuracy.

In 2003, Farnell and Wilbur [20] presented simulations of the grid erosion processes for two proposed sets of ion thruster grids for the NEXT project. They discussed structural failure and electron backstreaming due to acceleration grid erosion as two possible failure mechanisms. The ffx was used in the simulation. It is a three-dimensional, Cartesian optics code, extended from the igx code with additional features such as the ability to model a wide range of grid geometries including cusp details and mis-aligned aperture pairs. The simulation is applied to a two, quarter-sized apertures in a hexagonal aperture layout. The field is solved using a relaxed Gauss-Seidel multigrid method. Erosion of a cusped grid geometry was simulated using checker-board like simulation geometry. More details about the ffx code can be found in [21].

In 2003, Emhoff and Boyd [17] presented a computational study of the NEXT ion thruster optics. The optics code they used provides a two-dimensional axisymmetric simulation of a single aperture on a uniform Cartesian grid. The code uses a PIC method to simulate Xe, Xe$^+$ and Xe$^{++}$ and the DSMC method for processing the particle collisions. They explored the dependence of the accelerator grid current on downstream domain length and suggested an improving method. They also studied the effect of beam voltage on accelerator grid aperture wall erosion and performed life estimates for both cusped and non-cusped optics geometries.

In 2003, Kafafy and Wang [34] developed a fully three-dimensional ion optics simulation code, the IFE–PIC, which is based upon a standard PIC code and adopts the recently developed three-dimensional IFE method. The developed three-dimensional IFE field solver demonstrated very good agreement with field solutions obtained by highly accurate finite element solvers such as FEMLab® by the MathWorks Inc. The ion optics simulations also show good agreement with simulations obtained by Wang’s ion optics simulation code [89], which was validated by experimental results.

In 2004, Emhoff and Boyd [18] presented a numerical study to characterize the error
associated with the solution of the field equation. They suggested an improved two-dimensional axisymmetric field solver in which the potential mesh was refined to reduce numerical errors. For a two-dimensional single aperture domain, the most refined simulation takes over 8 hours on a 3.06 GHz PC, which makes the technique prohibitively impractical for three-dimensional simulations.

In 2004, Farnell [19] used the ffx code to investigate the lifetime and propellant throughput capability of the High Power Electric Propulsion (HiPEP) thruster ion optics. Erosion predictions are presented as a function of beamlet current, accel grid voltage, and propellant utilization efficiency. He also performed a design parametric study in which design parameters were varied in a systematic manner to assess their effect on beamlet current limitations and electron backstreaming margins.

In 2004, Kafafy and Wang [35] introduced the hybrid-grid IFE–PIC (HG–IFE–PIC) code which is an extension to the IFE–PIC code. The HG–IFE–PIC code uses two different meshes for particle simulation and field solution. The PIC mesh is a uniform Cartesian mesh which retains the simplicity and speed of a standard PIC code, while the IFE mesh is a multi-zone stretched Cartesian-based tetrahedral mesh which has much less number of nodes and elements than a uniform IFE mesh. They used the HG–IFE–PIC code to study the ion optics of a subscale gridlet in a domain that includes three non-similar apertures for the first time ever. The details of the HG-IFE-PIC model and the new meshing technique is discussed in chapter 5, while the details of the ion optics simulations is given in chapter 6.

In 2005, Kafafy et al. [33] applied the HG-IFE-PIC code to investigate the dynamic behavior of beam ions in NEXT ion optics. The code was used to span the ion optics operation envelope from cross-over up to perevance. The cross-over limit predictions were in a reasonable agreement with experimental data.

## 2.3 Previous Work On Spacecraft–Ion Thruster Plume Interaction – Review of the Last Decade

In this section, we present a chronological review of the research work performed in the last decade to investigate the aspects of the spacecraft–ion thruster plume interactions through physical and numerical modeling.

In 1995, Wang and Brophy [82] developed a 3D particle-in-cell Monte-Carlo collision simulation model for ion beam-spacecraft plume interactions and presented
simulation results for ion beam emissions under various charging conditions. They found that when the interaction is within the low-charging regime, such as that of an ion propulsion application, interactions are primarily due to the charge-exchange ions; these typically do not alter the spacecraft’s charging environment. They also noted that ion beam emissions in the high-charging regime may be used to efficiently discharge a spacecraft.

In 1996, Samanta Roy et al. [67, 68] developed a two-dimensional axisymmetric physical and numerical model to investigate the plasma environments induced by an ion thruster and to assess plume backflow contamination. The model takes into account five components of the thruster effluents: propellant beam ions, propellant neutrals, propellant CEX ions, non-propellant efflux neutrals and CEX ions, and electrons. They applied the model to predict the propellant CEX plasma and sputtered Mo-grid metal efflux from the NASA 30-cm ion thruster (NSTAR) over a wide range of operating conditions. They also showed that the ratio of the propellant Xe ions to sputtered Mo ions is not constant throughout the backflow region. The Mo deposition predicted was less than what was previously expected. In addition, they examined the sensitivity of the plume backflow to the beam electron temperature and found they are strongly correlated. Finally, they evaluated the application of an optional plume shield which was found to be effective in reducing the backflow for the two-dimensional model they studied.

In 1996, Samanta Roy and Hastings [66] also developed a three-dimensional model to study the composite plume backflow of two 8-cm ion thrusters. The resulting potential structure was found to trap the propellant CEX ions and introduce a preferred ejection direction, whereas, it had a little effect on the sputtered grid material due to its higher initial energy. They concluded that the net backflow of dual thrusters is not a simple superposition of the backflow from two single thrusters.

In 1996, Samanta Roy et al. [69] developed a fully three-dimensional hybrid plasma particle-in-cell model for multi-computer environments to assess the spacecraft backflow contamination of an ion thruster. They presented results of plume backflow for a 13-cm Xenon ion thruster operating with a current level of 0.4 A on a model spacecraft. The computational domain was over 40 m³ in volume and contained 9.4 million grid points, and the number of simulation particles used was over 35 million representing CEX Xe ions produced in the plume. In their calculations, they used a massively parallel 256-node Cray T3D. However, the spacecraft model that was studied was oversimplified because of numerical constants such that it was lacking many of the important geometric details of a real spacecraft.
In 1997, Wang et al. [83] developed a global analysis of ion thruster plume interactions for interplanetary spacecraft based on fully three-dimensional electrostatic particle-in-cell Monte-Carlo simulations of the near-field region and 2$\frac{1}{2}$-dimensional electromagnetic hybrid PIC simulations of the far-field region. They showed that while the charge exchange ion interaction dominates in the vicinity of the spacecraft, the presence of the solar wind will induce kinetic couplings between the plume and the solar wind in the far-field via electromagnetic plasma instabilities. The instabilities can generate enhanced magnetic field fluctuations leading to wave-particle scatterings of both the beam ions and the CEX ions.

In 1999, Gardner and Katz [22] developed a parametric model of an electric propulsion plume including primary beam and line-of-sight ions and hollow cathode neutralizer ions. They also modeled plume optical irradiance spectra and line-of-sight intensities. The Environment Work Bench (EWB) was used to implement the models. They performed calculations to compare the thruster plumes for both the Deep Space–1 30-cm ion thruster and the SPT-100 stationary plasma thruster. They presented preliminary results and discussed the models, testing and analysis required to understand the impact of the mentioned thrusters on spacecraft subsystems.

In 1999, VanGilder et al. [77] presented a two-dimensional axisymmetric hybrid Monte Carlo particle-in-cell code to simulate the plumes of ion thrusters and Hall current thrusters. The code employed two different grids; an unstructured grid used by the Direct Simulation Monte Carlo (DSMC) code and a non-uniform Cartesian grid used by the PIC code. The authors compared the computed flow field results with the existing experimental measurements for the UK-10 ion thruster which consisted of ion flux, ion density, and floating potential data. Xe neutrals and ions were modelled directly, while electrons were described by Boltzmann law. They included the effect of finite back pressure. They noticed high sensitivity of simulation results to input conditions assumed at the thruster exit plane, particularly the uncertainty in specifying the effects of the curvature of the dished grids of the UK-10 thruster on the exit ion velocity profile. They also studied the sensitivity of simulations to certain parameters including; the cross-section of CEX reactions, the mechanics of CEX reactions, and the electron temperature. They found that the beam ions are only moderately dependent on variations in these parameters, whereas the CEX ions are sensitive to them.

In 2001, Wang et al. [80] developed a three-dimensional particle-in-cell model to compute the ion propulsion induced plasma environment for the Deep Space 1 (DS1) spacecraft. They compared simulations with in-flight measurements of CEX plasma from the ion propulsion diagnosis subsystem (IDS) on DS1. It was found that the
plasma from the plasma environment of DS1 is dominated by the CEX plasma from the plume. They found that the CEX plasma near the spacecraft surface, for a typical thruster operating conditions, had a density ranging from $10^6$ cm$^{-3}$ near the thruster to $10^4$ cm$^{-3}$ at the opposite end of the spacecraft surface and a current density ranging from $10^{-7}$ A/cm$^2$ to $10^{-9}$ A/cm$^2$. They also showed that, for an interplanetary spacecraft and a moderate charging potential, CEX ion backflow goes through an expansion process similar to that of mesothermal plasma into a vacuum or plasma wake. They also noted that since the spacecraft potential is shielded by the thin propellant CEX plasma sheath, it does not have any significant effect on the backflow of the ionized Mo particles sputtered from the thruster.

In 2003, Wang et al. [78] developed a three-dimensional PIC model which incorporates the immersed finite element (IFE) method for plasma particle simulations. The major advantage the IFE-PIC model for simulations involving complex geometric boundary conditions is to allow the direct usage of standard Cartesian mesh-based algorithms for particle pushing regardless of the geometry of object boundaries. The IFE formulation also allows material properties to be included explicitly and maintains the desired physics at the given material interface. Preliminary results were presented therein.

In 2004, Mandell et al. [44] presented a two-dimensional axisymmetric plume model of a 30-cm ion engine, operating at high jet power (>20 kW). The model accounted for the main ion components expected to be present in the thruster plume, namely main beam ions and ion products from ion-neutral scattering. They found main ions within $\sim 30$ deg of the beam axis of symmetry, and have energies in excess of 3000 eV and peak densities in the order of $10^{15}$ m$^{-3}$ near the thruster exit. CEX ions are expelled radially from the main beam due to their low energies (<100 eV) and were thus dominating the plume at large angles (>70 deg). Ions that are elastically scattered were found to retain most of the initial kinetic energy (100-3000 eV) and were found with peak densities in the order of $10^{12}$ m$^{-3}$ and populate the complete range of plume angles. In their simulations, they used the EPIC/Plume tool.

In 2004, Wang et al. [79] [11] introduced a prototype virtual laboratory for spacecraft plasma interactions and electric propulsion. This virtual laboratory consists of a simulation engine and a virtual testing environment. A user provides physical parameters and spacecraft configurations to the simulation engine. The virtual laboratory simulates the physical processes using a set of particle simulation codes and then displays the results in a virtual testing environment using immersed and/or collaborative visualization. They developed a new particle simulation code, DRACO, for plasma simulations involving complex boundary conditions. DRACO incorpo-
rates the IFE–PIC algorithm. This method allows one to use a Cartesian mesh to handle complex geometric or time-varying interface between plasma and object without sacrificing the accuracy in electric field solutions.

In 2004, Wang and Kafafy [84] used the IFE–PIC code to study the interaction of multiple ion thruster plumes with spacecraft. They studied three configurations of ion thrusters; a single thruster, three in-plane thrusters, and an array of four parallel thrusters. The thrusters in each configuration were given major possible firing options. They showed the dependence of the near-spacecraft plasma environment on spacecraft-thruster configurations and firing options. They also assessed the contamination of solar arrays by ion thruster operation for each configuration and thruster fining option. The details of this work will be provided in chapter [7].

2.4 Review On Methods To Solve A Field Problem With Complex Boundaries

The electrostatic field problem with complex boundaries may be, in general, described by a Poisson’s equation with discontinuous coefficients. This equation appears in many scientific and engineering applications which includes, but is not limited to, fluid mechanics, electromagnetics, heat transfer, biomechanics as well as plasma physics. It is well known that the solution of such equation not only slows down the entire simulation for many applications, but also leads to loss in accuracy. Solving interface problems efficiently and accurately is still a challenge because of many irregularities associated with them [41]. In order to overcome the obstacles in solving Poisson’s equations with discontinuous coefficients, many numerical methods have been developed. In the following, we present a brief review on the numerical methods which are closely related to the IFE method. We classify these methods similar to [41] as;

- body-fitting-grid methods discretization
- Cartesian-grid methods based on finite difference discretization
- Cartesian-grid methods based on finite element discretization
2.4.1 Body-Fitting-Grid Methods

It is well known that a second order accurate approximation to the solution of an interface problem can be generated by the Galerkin finite element method with the standard linear basis functions if the triangulation is aligned with the interface, i.e. body fitting grid. \[4, 10, 16\]. One advantage of the finite element formulation is that the resulting linear system of equation is symmetric positive definite for a self-adjoint elliptic operator which ensures the stability of the algorithm. It is also possible to produce second order accurate approximation to the solution of the interface problem by applying the appropriate second order accurate finite difference schemes in a body-fitting curvilinear grid \[85\].

Disadvantages of Body-Fitting-Grid Methods

It is difficult and time consuming to generate a body fitting grid for an interface problem in which the interface separates the solution domain into pieces or problems with complicated geometries. Such a difficulty becomes even more severe for moving interface problems because a new grid has to be generated at each time step \[11, 15\]. Other concerns are raised by particle simulation models. Particle simulation models using body-fitting grids usually put an annoying computational over-head to search for particles using cumbersome techniques \[27\]. Such complicated particle search techniques slow down the whole simulation and/or limit the size of the problems that can be handled.

Advantages of Cartesian-Grid Methods

The advantages of using a Cartesian grid or an adaptive Cartesian grid method to solve an interface problem is summarized below \[41, 15\]:

- Grid generation is trivial and almost costless. This becomes much more significant for moving boundary/interface problems.
- Many efficient and popular packages/solvers and numerical methods are available for Cartesian grids. For example, fast Poisson solvers such as fishpack, Navier-Stokes equation solvers in two and three dimensions on a rectangular square or a box, Clawpack for conservation laws, and FFT packages etc. It
is relatively easier to incorporate new methods using existing packages/solvers based on the same grid.

- The level set method, which has been successfully used to treat a number of moving interface/boundary problems, especially for problems with topological changes, and for problems in three dimensions - works best with Cartesian grids.

- Construction of super convergent approximations to important physical quantities such as fluxes is much easier using Cartesian grids.

2.4.2 Cartesian-Grid Methods Based On Finite Difference Discretization

The numerical methods based on Cartesian grids may have some difficulties that need to be overcome and may lose accuracy as well for curved interfaces/boundaries. Due to non-smoothness of the solutions, many standard finite difference methods do not apply for interface problems. A huge research effort has been made regarding these problems. In the following, we will only review most of the methods which are related to this work.

The Immersed Boundary Method (IBM)

Peskin [58] introduced the immersed boundary method in 1976. The method was designed to solve problems of biological fluid mechanics which often involve the interaction of a viscous incompressible fluid with an elastic membrane which can be considered as a boundary immersed in the fluid. The main idea of the method is to use a regular Eulerian computational grid for the fluid mechanics together with a Lagrangian representation of the immersed boundary. The immersed boundary exerts a singular force on the fluid and at the same time moves at the local fluid velocity. The interaction between the fluid and immersed boundary can be modeled by a well chosen discretized approximation to the Dirac delta function, which is called a discrete delta function. This approach has been applied successfully to problems of blood flow pattern in the heart, wave propagation in the cochlea, flow in collapsible tubes, aquatic animal locomotion, platelet aggregation during blood clotting, the flow of suspensions, flow and transport in a renal arteriole, and the cell and tissue
deformation under shear flow \cite{38}. However, the IBM is designed for problems with only a singular source term and continuous coefficients.

The Level Set Method (LSM)

Osher and Sethian \cite{54} first proposed the level set method in 1988. The method has been successfully used to treat a number of moving interface/boundary problems, especially for problems with topological changes, and for problems in three dimensions. The advantage of the level set method is that one can perform numerical computations involving curves and surfaces on a fixed Cartesian grid without having to parameterize these objects (this is called the Eulerian approach). The level set method works best with Cartesian grids.

The Smoothing Method for Discontinuous Coefficients

Sussman et al. \cite{71} introduced in 1994 a simple approach to adopt the level set method by smoothing out the coefficient. The level set expression of interfaces makes the smoothing method much easier for two and three dimensional problems. However, solutions are also smeared out by the smoothing method.

The Immersed Interface Method (IIM)

LeVeque and Li \cite{43} introduced the immersed interface method (IIM) in 1994. The motivation was to develop a second-order accurate finite difference scheme for elliptic equations which may have discontinuous coefficients, or singular sources along a general interface, externally imposed constraints in the jump of the solution or its normal derivatives across an interface. The method was successful in applications to one-, two- and three-dimensional problems; elliptic, parabolic, hyperbolic, and mixed type equations; fixed and moving interfaces, and many applications \cite{40, 15}. However, the resulting linear system of equations is not symmetric positive definite. In addition, the method is not very robust, because the stability of the algorithm may depend on the choice of one or more additional grid points in addition to the standard finite difference scheme.
The Embedded Curved Boundary Method (ECB)

Hewett [27] introduced the embedded curved boundary method (ECB) in 1997. The method was intended to incorporate piecewise-linear approximations to curved boundaries for codes that use orthogonal meshes. The ECB is designed to retain much of the fidelity of unstructured-grid methods while retaining simplicity and run time advantage of orthogonal mesh based codes. This method provides a practical approach to problems such as space charge emission. However, the resulting linear system of equations from this method is also not symmetric positive definite; a property that is very favorable for the stability of the solution.

2.4.3 Cartesian-Grid Methods Based On Finite Element Discretization

The Partition of Unity Method (PUM)

A new generalized finite element method that was presented by Babuška and Melenk [5] in 1997. This method features the ability to include in the finite element space knowledge about the partial differential equation being solved. In addition, finite element spaces of any regularity can be constructed very easily. The PUM method underlies many of the other finite element methods and can be considered as a generalization of the $h$ and $p$ version of the finite element method which underlies many of the finite element methods developed earlier. Although, the PUM does not present a specific set of finite element methods that are suitable for each situation, many finite element methods that have been developed later to address specific problems much more efficiently have followed the same approach of the PUM.

The Extended Finite Element Method (X-FEM)

Sakumar et al. [61] introduced the extended finite element method in 2000. This method was designed specifically to model cracks in three-dimensions. In this method, a discontinuous function and the two-dimensional asymptotic crack-tip displacement fields are added to the finite element approximation to account for the crack using the notion of partition of unity. The additional functions are used to model the presence of cracks, voids or inhomogeneities, and also to improve the accuracy in problems where aspects of the functional behavior of the solution field is known a
priori.

**The Immersed Finite Element Method (IFE)**

Lin et al. \[42\] introduced the immersed finite element method in 2001. The method constructs a rectangular finite element space based on rectangular elements to be used for the solution of boundary value problems with discontinuous coefficients. The basis functions in this space are constructed to satisfy the interface jump conditions either exactly or approximately. Such interface problem is involved in many applications such as the projection method for solving Navier-Stokes equations involving two-phase flow, the Hele-Shaw flow and many others. In 2003, Li et al. \[41\] extended the method to include triangular elements. They also considered both non-conforming and conforming finite element spaces. The corresponding interpolation functions of the method was proved to be second order accurate in the maximum norm. However, the IFE method proposed was only appropriate for two-dimensional and axisymmetric problems. Recently, Kafafy et al. \[32\] extended the IFE method to involve the solution of essentially three-dimensional problems by constructing the appropriate three-dimensional finite element space based on tetrahedral elements. Details of the development of the three-dimensional IFE method, properties of the finite element space, and numerical examples are given in chapter 3.

**Immersed Finite Element Method (IFEM)**

Zhang et al. \[93\] presented the finite element method in 2004. This method was proposed for the solution of complex fluid and deformable structure interaction problems encountered in many physical phenomena. In this method, a Lagrangian solid mesh moves on top of a background Eulerian fluid mesh which spans over the entire computational domain. Both fluid and solid domains are modelled with the finite element methods and the continuity between the fluid and solid sub-domains are enforced via the interpolation of the velocities and the distribution of the forces with the Reproducing Kernel Particle Method (RKPM) delta function. The higher-ordered RKPM delta functions enables the fluid domain to have nonuniform spatial meshes with arbitrary geometries and boundary conditions. This method was mentioned here because it shares the same name as the finite element methods presented in this work, although they follow different approaches and target different applications.
2.5 Contribution

Reviewing the current plasma simulation codes that are used for ion optics and ion thruster–spacecraft interactions, we found that most of them are incapable of resolving the complex effects of realistic geometries in large-scale simulations. Most of these codes use standard finite difference field solvers which are known to lose accuracy in the vicinity of geometric boundaries and interfaces, if not carefully adjusted which is mostly the case. Body-fitting grid finite element particle-in-cell codes are very accurate, but much time consuming. Therefore, they are limited to small size problems.

In this research, we develop a three-dimensional immersed finite element method, which we use to build a three-dimensional immersed finite element (IFE) field solver. The most attractive advantage of the IFE solver is that it can solve the interface field boundary value problem on a Cartesian-based tetrahedral mesh irrespective of the shape and position of the interface. It can also be used to investigate the effect of involving materials with different dielectric constants since it incorporates explicitly the material properties in the field solution.

Next, we integrate the IFE field solver to a Cartesian-grid particle-in-cell (PIC) code which is modified from a legacy standard PIC code. It is well known that Cartesian-grid PIC codes are much very fast in performing particle-mesh interpolations and particle pushing. We use the resulting IFE–PIC code in doing ion optics simulations as well as ion thruster–spacecraft interaction simulations.

We also introduce a novel meshing technique in particle simulation codes. In this technique, we generate two separate meshes; a uniform Cartesian mesh for the PIC code and a multi-zone stretched Cartesian-based tetrahedral mesh for the IFE field solver. The uniform Cartesian PIC mesh retains the maximum speed of a standard PIC code, whereas the multi-zone stretched Cartesian-based tetrahedral mesh provides the required mesh refinement for accurate field solution avoiding computational and storage burden caused by IFE mesh over-refinement.

Ion optics simulations are performed on the NEXT ion optics using the HG-IFE-PIC Ion Optics model to assess the ion extraction capabilities as well as the impingement current and electron backstreaming limits. In addition to the standard HG-IFE-PIC Ion Optics model, we introduce a Streamline HG-IFE-PIC Ion Optic model which is meant to be much faster and less memory consuming. We perform ion optics simulations in a two-quarter aperture domain as well as a whole ion optics gridlet seven-aperture domain. Since similarity among all apertures in a subscale gridlet
can not be clearly justified for such a small number of apertures, it may be necessary to incorporate all apertures in the simulation.

In the ion thruster–spacecraft interaction simulations, we use a simplified model for the Dawn spacecraft in which we kept most of the major geometric details to keep the model as close to reality as possible. We also study the effect of several thrusters arrangements and firing options on both near-spacecraft plasma environment and spacecraft contamination.
Chapter 3

A Three-Dimensional Immersed Finite Element Method

3.1 Introduction

This chapter introduces the three-dimensional immersed finite element method. First, we present the interface boundary value electric field problem, which is associated with ion optics and ion thruster-spacecraft interactions simulations. The problem is then put in its weak formulation. Thereafter, we develop a three-dimensional immersed finite element space that uses a structured Cartesian mesh to solve an essentially three-dimensional interface field problem. The mesh-interface intersection topologies are discussed. The local nodal basis functions of interface elements (IFE basis functions) are constructed. The existence and uniqueness of these functions are proved. The functions are also shown to form partition of unity and to be consistent with the classical finite element basis functions. Finally, several numerical experiments are conducted to demonstrate the convergence properties of the new method.

3.2 The Interface Boundary Value Problem

The immersed finite element (IFE) methods are a class of finite element methods that can use a structured Cartesian mesh to solve boundary value problems (BVP)
with discontinuities in the coefficients of the partial differential equations (PDE) across one or several arbitrary interfaces in the solution domain. A BVP of this type is referred to as an interface problem \[41\].

In an electrostatic (ES) particle-in-cell (PIC) simulation, it is possible to consider the associated field problem as an interface problem; in which all objects boundary surfaces are treated as interface surfaces as schematically illustrated in figure \[3.1\]. The interface field BVP can be generally formulated as

\[- \nabla \cdot (\epsilon \nabla \Phi) = f(x; \Phi), \; x \in \Omega \subset \mathbb{R}^3,\]

\[\Phi|_{\partial \Omega_D} = g_D,\]

\[\epsilon \frac{\partial \Phi}{\partial n}|_{\partial \Omega_N} = g_N.\]

where \(\Omega_D\) is the portion of the boundary surface where Dirichlet boundary conditions apply and \(\Omega_N\) is the portion of the boundary where Neumann boundary conditions apply.

![Figure 3.1: Solution domain of the interface BVP.](image)

The problem statement is completed by the set of jump conditions across the interface \(\Gamma\):

\[[\Phi]_\Gamma = 0,\]

\[\left[ \epsilon \frac{\partial \Phi}{\partial n} \right]_\Gamma = 0.\]

where the square brackets \([ \ldots ]\) declare the difference function across the interface. Here, for the simplicity of presentation, we assume that \(\Omega \subset \mathbb{R}^3\) is a convex polygonal
domain, the interface $\Gamma$ is a surface separating $\Omega$ into two sub-domains $\Omega^-$ and $\Omega^+$ such that $\Omega = \Omega^- \cup \Omega^+ \cup \Gamma$. Accordingly, the coefficient $\epsilon(x)$ is a piecewise constant function defined as

$$
\epsilon(x) = \begin{cases}
\epsilon^-, & x \in \Omega^-; \\
\epsilon^+, & x \in \Omega^+.
\end{cases}
$$

### 3.3 Weak Formulation Of The Field Problem

To put the interface problem in its weak form, we multiply both sides of equation (3.1) by a test function $w(x) \in H^1_0(\Omega)$ and integrate over the domain $\Omega^+ \cup \Omega^- = \Omega - \Gamma$ to obtain

$$
-\iiint_{\Omega} w \nabla \cdot (\epsilon \nabla \Phi) \, dV = \iiint_{\Omega} w f \, dV,
$$

where $dV = dx \, dy \, dz$.

By vector identity, we can write (3.6) as

$$
-\iiint_{\Omega} \nabla \cdot (w \epsilon \nabla \Phi) \, dV + \iiint_{\Omega} \epsilon \nabla \Phi \cdot \nabla w \, dV = \iiint_{\Omega} w f \, dV.
$$

Applying the Green’s theorem, we can write the first term in the left hand side of (3.7) as

$$
-\iiint_{\Omega} \nabla \cdot (w \epsilon \nabla \Phi) \, dV = -\int_{\partial \Omega} w \epsilon \frac{\partial \Phi}{\partial n} \, dS.
$$

Substitute (3.8) into (3.7),

$$
\iiint_{\Omega} \epsilon \nabla \Phi \cdot \nabla w \, dV - \int_{\partial \Omega} w \epsilon \frac{\partial \Phi}{\partial n} \, dS = \iiint_{\Omega} w f \, dV
$$

Without loss of generality, we can assume that $g_D = 0$ in (3.2) and $g_N = 0$ in (3.3). Then, (3.9) reduces to

$$
\iiint_{\Omega} \epsilon \nabla \Phi \cdot \nabla w \, dV = \iiint_{\Omega} w f \, dV
$$

In the following, we will refer to the integral equation (3.10) as the weak formulation of the field problem.
3.4 A Three–Dimensional IFE Space

Here, we develop a suitable finite element space that can use a structured partition to solve three-dimensional interface problems. Without loss of generality, we consider a Cartesian partition $\mathcal{T}_h$ of the solution domain $\Omega$ that is formed by first partitioning $\Omega$ into uniform cubes whose edges are parallel to the coordinate axes, and $\mathcal{T}_h$ is formed by further partitioning each cube into 5 tetrahedrons in the way illustrated in Figure 3.2 such that the vertices of each tetrahedron in $\mathcal{T}_h$ are nodes forming those cubes. Each of the resulting tetrahedrons is shaded, in turn, in Figure 3.3 for better illustration.

![Figure 3.2: Partitioning of a Cartesian IFE cell.](image)

Since the partition $\mathcal{T}_h$ is formed independent of the interface surface $\Gamma$, some of the tetrahedrons will intersect with $\Gamma$, but most of the elements will not when the partition is fined enough. We call any element intersecting with $\Gamma$ an interface element; otherwise, we call it a non-interface element.

Assume that $A_1, A_2, A_3,$ and $A_4$ are the four vertices of any tetrahedral element $T \in \mathcal{T}_h$. In a typical non-interface element, we will use the standard linear local nodal finite element basis functions to carry out the approximation, where $\bar{\psi}_i(x), i = 1, 2, 3, 4$ will denote these standard linear basis functions such that

$$
\bar{\psi}_i(A_j) = \begin{cases} 
1, & \text{if } i = j, \\
0, & \text{if } i \neq j.
\end{cases}
$$

The local FE space in $T$ is defined by

$$
\mathcal{S}_h(T) = \text{span}\{\bar{\psi}_i(x), i = 1, 2, 3, 4\}.
$$
On the other hand, we will use $\psi_i(x), i = 1, 2, 3, 4$ to denote the linear local nodal basis functions defined on each interface element $T \in \mathcal{T}_h$, and the local IFE space in $T$ will be defined as

$$S_h(T) = \text{span}\{\psi_i(x), i = 1, 2, 3, 4\}.$$ 

Our main effort in this section is to develop the local finite element space $S_h(T)$ for each interface element $T \in \mathcal{T}_h$ such that each function in this local space will satisfy the interface jump conditions in a certain approximation sense. Afterwards, for each node $x_i$ in the partition $\mathcal{T}_h$, we introduce a global basis function $\Psi_i(x)$ such that

$$\Psi_i|_T \in \begin{cases} S_h(T), & \text{if } T \text{ is an interface element;} \\ \mathcal{S}_h(T), & \text{if } T \text{ is a non-interface element;} \end{cases}$$ 

and for any node pair $(x_i, x_j)$ of $\mathcal{T}_h$, we have

$$\Psi_i(x_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$
where $x_i$ and $x_j$ are two nodes in the mesh $T_h$. Finally, the IFE space on the whole domain $\Omega$ is defined as follows:

$$S_h(\Omega) = \text{span}\{\Psi_i(x) \mid x_i \text{ is a node of } T_h\}.$$ 

The related FE method based on this IFE space can be formulated as usual, and is called an IFE method.

### 3.4.1 Intersection Topology

Investigating the possible intersection topologies between a typical interface tetrahedral element $T \in T_h$ and the interface surface $\Gamma$, we find that it is necessary to consider only two types of interface element, assuming that the partition size is small enough compared to the interface surface curvature, the rest of the interface elements can be effectively considered as non-interface elements. These two typical intersection topologies are (see Figure 3.4)

1. **Three-edge cut**: The interface surface $\Gamma$ intersects with the edges of $T$ at three distinct points which are not on the same face of $T$.

2. **Four-edge cut**: The interface surface $\Gamma$ intersects with the edges of $T$ at four distinct points on four different edges, and each three of these intersection points are not on a line.

### 3.4.2 Special Intersection Topology

Other intersection topologies are certainly possible. One unusual case is that the interface element $T$ is separated by the interface $\Gamma$ into three or more subregions, each of which has a volume comparable to the others. This generally indicates that the partition is not fine enough in the neighborhood of $T$, and a finer partition is necessary. Another possibility is that the six edges of the tetrahedron are intersected by a smooth enough surface, like the surface of a sphere with a large enough radius compared to the mesh size. Some edges might be even intersected more than once, see Figure 3.5. In such a case, we notice that since the surface is smooth, each cut divides the tetrahedron into two subregions: a very small subregion whose volume is nearly zero, and a larger subregion which nearly occupies the whole tetrahedron. For
such case, we can neglect the effect from the subregions which have much smaller volume and assume the tetrahedron is fully occupied by the larger subregion. In other words, we can just consider the tetrahedron as a non-interface tetrahedron, sitting on one side of the interface. Other situations may be handled similarly.

Figure 3.4: Intersection topologies of tetrahedral elements.

Figure 3.5: An odd intersection topology.
3.4.3 Linear Local Nodal FE Basis Functions

For a typical non-interface tetrahedron $T$, with vertices $A_i, (i = 1, 2, 3, 4)$, we can introduce the following four linear local nodal basis functions

$$\psi_i(x) = b_1 x + b_2 y + b_3 z + b_4, \quad x \in T\quad (3.11)$$

$$i = 1, 2, 3, 4$$

that satisfy the following nodal values constraints:

$$\psi_i(A_j) = \begin{cases} 1, & i = j, \\
0, & i \neq j. \end{cases}$$

$$i, j = 1, 2, 3, 4$$

3.4.4 Linear Local Nodal IFE Basis Functions

For a typical interface tetrahedron $T$, with vertices $A_i, (i = 1, 2, 3, 4)$, the interface $\Gamma$ divides $T$ into two sub-elements: $T^+ = T \cap \Omega^+$ and $T^- = T \cap \Omega^-$. With this natural partition of $T$, we can introduce four piecewise linear local nodal basis functions

$$\psi_i(x) = \begin{cases} \psi_i^+(x) = b_1 x + b_2 y + b_3 z + b_4, & x \in T^+ \\
\psi_i^-(x) = b_5 x + b_6 y + b_7 z + b_8, & x \in T^- \end{cases},$$

$$i = 1, 2, 3, 4$$

that satisfy the following constraints:

- Nodal values specification:

$$\psi_i(A_j) = \begin{cases} 1, & i = j, \\
0, & i \neq j. \end{cases}$$

$$i, j = 1, 2, 3, 4$$

- The continuity across the interface inside the tetrahedron $\Gamma_T = T \cap \Gamma$:

$$\psi_i^+(P_j) = \psi_i^-(P_j),$$

$$i = 1, 2, 3, 4$$ and $$j = 1, 2, 3.$$
Here $P_j, j = 1, 2, 3$ are three points on $\Gamma_T$ whose choice will be made clear in
the next two subsections according to the involved topology of $\Gamma_T$.

- The flux continuity across $\tilde{\Gamma}_T$:

$$\int_{\tilde{\Gamma}_T} \left( \beta \frac{\partial \psi^+}{\partial n} - \beta \frac{\partial \psi^-}{\partial n} \right) ds = 0,$$

$$i = 1, 2, 3, 4.$$

Here $\tilde{\Gamma}_T$ is the plane determined by the points $P_j, j = 1, 2, 3$ above, and $n$ is
the normal of $\Gamma_T$.

These constraints provide a system of eight simultaneous linear algebraic equations
about the coefficients of each local nodal basis function. To show that these condi-
tions are enough to precisely determine a local nodal basis function, we will carry
out the discussion in the usual way by transferring the interface element $T$ to the
reference tetrahedron $\hat{T}$ with the following affine transformation:

$$\mathbf{x} = A_4 + \mathbf{B} \cdot \mathbf{r} \quad (3.13)$$

where the vertices of the reference tetrahedron $\hat{T}$ are $\hat{A}_1 = (1, 0, 0)^T, \hat{A}_2 = (0, 1, 0)^T, \hat{A}_3 = (0, 0, 1)^T$ and $\hat{A}_4 = (0, 0, 0)^T$, $\mathbf{x} = [x, y, z]^T$ is the coordinate of a point in the interface element $T$, and $\mathbf{r} = [r, s, t]^T$ is the coordinate of its corresponding point in the reference tetrahedron, and the transformation matrix $\mathbf{B}$ is given by

$$\mathbf{B} = \begin{pmatrix}
x_1 - x_4 & x_2 - x_4 & x_3 - x_4 \\
y_1 - y_4 & y_2 - y_4 & y_3 - y_4 \\
z_1 - z_4 & z_2 - z_4 & z_3 - z_4
\end{pmatrix}.$$

In reference coordinates, the local nodal basis functions have the following expressions:

$$\tilde{\psi}_i(\mathbf{x}) = \hat{\psi}_i(\mathbf{r}) = \begin{cases}
\hat{\psi}_i^+(\mathbf{r}) = a_{i,1}r + a_{i,2}s + a_{i,3}t + a_{i,4}, & \mathbf{r} \in \hat{T}^+ \\
\hat{\psi}_i^- (\mathbf{r}) = a_{i,5}r + a_{i,6}s + a_{i,7}t + a_{i,8}, & \mathbf{r} \in \hat{T}^- 
\end{cases} \quad (3.14)$$

$$i = 1, 2, 3, 4.$$

Then, we plan to show that the nodal value specifications and the interface jump
conditions will uniquely determine $\psi_i(\mathbf{r}), i = 1, 2, 3, 4$ and the same conclusion follows
for the local nodal basis functions $\psi_i(\mathbf{x}), i = 1, 2, 3, 4$. To be specific, we need to
discuss the two intersection topologies separately.
Three-edge cut element

We first consider the three-edge cut case in which we assume that the interface $\Gamma$ intersects the edges of an interface element $T$ at three points $P_1, P_2$ and $P_3$, see Figure 3.6. Depending on which edges the interface $\Gamma$ passes through, there are four possible three-edge cut, two of which are plotted in Figure 3.6. We will discuss these two cases, and the arguments for the second case apply to the rest.

![Figure 3.6: Two cases of possible three-edge cut in the reference element $\hat{T}$.

Assume that the images of $P_1, P_2, P_3$ in the reference element $\hat{T}$ under the affine mapping are $\hat{P}_1, \hat{P}_2$ and $\hat{P}_3$, with the following coordinates:

Three-edge cut (case 1):

$$\hat{P}_1 = [\xi, 0, 0]^T, \quad \hat{P}_2 = [0, \eta, 0]^T, \quad \hat{P}_3 = [0, 0, \zeta]^T,$$  \hspace{1cm} (3.15)

Three-edge cut (case 2):

$$\hat{P}_1 = [\xi, 0, 0]^T, \quad \hat{P}_2 = [1 - \eta, \eta, 0]^T, \quad \hat{P}_3 = [1 - \zeta, 0, \zeta]^T,$$  \hspace{1cm} (3.16)

with

$$0 \leq \xi \leq 1, \quad 0 \leq \eta \leq 1, \quad 0 \leq \zeta \leq 1.$$  \hspace{1cm} (3.17)
For three-edge cut (case 1), the nodal value specifications lead to the following equations:

\[ a_{i,4} = \hat{\psi}_i(\hat{A}_4) = \psi_i(A_4), \]
\[ a_{i,5} + a_{i,8} = \hat{\psi}_i(\hat{A}_1) = \psi_i(A_1), \]
\[ a_{i,6} + a_{i,8} = \hat{\psi}_i(\hat{A}_2) = \psi_i(A_2), \]
\[ a_{i,7} + a_{i,8} = \hat{\psi}_i(\hat{A}_3) = \psi_i(A_3). \]

The continuity jump condition across the interface leads to:

\[ \xi a_{i,1} + a_{i,4} = \xi a_{i,5} + a_{i,8}, \]
\[ \eta a_{i,2} + a_{i,4} = \eta a_{i,6} + a_{i,8}, \]
\[ \zeta a_{i,3} + a_{i,4} = \zeta a_{i,7} + a_{i,8}. \]

As for the flux continuity condition, we first note that the normal \( \mathbf{n} = [n_1, n_2, n_3]^T \) is mapped to the vector \( \hat{\mathbf{n}} = B^{-1}\mathbf{n} = [\hat{n}_1, \hat{n}_2, \hat{n}_3]^T \) in the reference coordinate system. By direct calculation, we can see that the flux continuity condition reduces to:

\[ \beta^+ \frac{\partial \hat{\psi}_i}{\partial \hat{\mathbf{n}}} - \beta^- \frac{\partial \hat{\psi}_i}{\partial \mathbf{n}} = 0, \]

and this leads to:

\[ (\hat{n}_1 a_{i,1} + \hat{n}_2 a_{i,2} + \hat{n}_3 a_{i,3}) \beta^+ = (\hat{n}_1 a_{i,5} + \hat{n}_2 a_{i,6} + \hat{n}_3 a_{i,7}) \beta^- . \]

Solving these 8 linear equations above, we obtain the following formulas about the coefficients:

\[ a_{i,1} = \frac{\xi \psi_i(A_1) - \psi_i(A_4) + (1 - \xi)a_{i,8}}{\xi} \]
\[ a_{i,2} = \frac{\eta \psi_i(A_2) - \psi_i(A_4) + (1 - \eta)a_{i,8}}{\eta} \]
\[ a_{i,3} = \frac{\zeta \psi_i(A_3) - \psi_i(A_4) + (1 - \zeta)a_{i,8}}{\zeta} \]
\[ a_{i,4} = \psi_i(A_4) \]
\[ a_{i,5} = \psi_i(A_1) - a_{i,8} \]
\[ a_{i,6} = \psi_i(A_2) - a_{i,8} \]
\[ a_{i,7} = \psi_i(A_3) - a_{i,8} \]
\[ a_{i,8} = \frac{\lambda_{i,1} \beta^- + \lambda_{i,2} \beta^+}{\lambda_3 \beta^- + \lambda_4 \beta^+} . \]
where
\[
\begin{align*}
\lambda_{i,1} &= \xi \eta \zeta (\hat{n}_1 \psi_i(A_1) + \hat{n}_2 \psi_i(A_2) + \hat{n}_3 \psi_i(A_3)), \\
\lambda_{i,2} &= \psi_i(A_4) (\hat{n}_1 \eta \zeta + \hat{n}_2 \xi \zeta + \hat{n}_3 \xi \eta) - \lambda_{i,1}, \\
\lambda_3 &= \xi \eta \zeta (\hat{n}_1 + \hat{n}_2 + \hat{n}_3), \\
\lambda_4 &= \hat{n}_1 \eta \zeta + \hat{n}_2 \xi \zeta + \hat{n}_3 \xi \eta - \lambda_3.
\end{align*}
\]

For the three-edge cut (case 2), the nodal value specifications lead to the following equations
\[
\begin{align*}
a_{i,4} &= \hat{\psi}_i(\hat{A}_4) = \psi_i(A_4), \\
a_{i,5} + a_{i,8} &= \hat{\psi}_i(\hat{A}_1) = \psi_i(A_1), \\
a_{i,2} + a_{i,4} &= \hat{\psi}_i(\hat{A}_2) = \psi_i(A_2), \\
a_{i,3} + a_{i,4} &= \hat{\psi}_i(\hat{A}_3) = \psi_i(A_3).
\end{align*}
\]

The continuity jump condition across the interface leads to
\[
\begin{align*}
\xi a_{i,1} + a_{i,4} &= \xi a_{i,5} + a_{i,8}, \\
(1 - \eta) a_{i,1} + \eta a_{i,2} + a_{i,4} &= (1 - \eta) a_{i,5} + \eta a_{i,6} + a_{i,8}, \\
(1 - \zeta) a_{i,1} + \zeta a_{i,3} + a_{i,4} &= (1 - \zeta) a_{i,5} + \zeta a_{i,7} + a_{i,8}.
\end{align*}
\]

The flux continuity leads to
\[
(\hat{n}_1 a_{i,1} + \hat{n}_2 a_{i,2} + \hat{n}_3 a_{i,3}) \beta^+ = (\hat{n}_1 a_{i,5} + \hat{n}_2 a_{i,6} + \hat{n}_3 a_{i,7}) \beta^-.
\]

Solving these 8 linear equations above, we obtain the following formulas about the coefficients:
\[
\begin{align*}
a_{i,1} &= \frac{\xi \psi_i(A_1) - \psi_i(A_4) + (1 - \xi) a_{i,8}}{\xi}, \\
a_{i,2} &= \psi_i(A_2) - \psi_i(A_4), \\
a_{i,3} &= \psi_i(A_3) - \psi_i(A_4), \\
a_{i,4} &= \psi_i(A_4), \\
a_{i,5} &= \psi_i(A_1) - a_{i,8}, \\
a_{i,6} &= \frac{\psi_i(A_2) \eta \zeta + \psi_i(A_4) [1 + \eta + \xi (1 - \eta)] + a_{i,8} (1 - \eta - \xi)}{\eta \xi}, \\
a_{i,7} &= \frac{\psi_i(A_3) \zeta \xi + \psi_i(A_4) [1 + \zeta + \xi (1 - \zeta)] + a_{i,8} (1 - \zeta - \xi)}{\zeta \xi}, \\
a_{i,8} &= \frac{\lambda_{i,1} \beta^- + \lambda_{i,2} \beta^+}{\lambda_3 \beta^- + \lambda_4 \beta^+}.
\end{align*}
\]
where

\[ \lambda_{i,1} = \xi \eta \zeta \left[ \hat{n}_1 \psi_i(A_1) + \hat{n}_2 \psi_i(A_2) + \hat{n}_3 \psi_i(A_3) \right] - \psi_i(A_4)(1 - \xi) \left[ \hat{n}_2 \zeta (1 - \eta) + \hat{n}_3 \eta (1 - \zeta) \right], \]

\[ \lambda_{i,2} = \eta \zeta \psi_i(A_4) \left[ \hat{n}_1 + \xi (\hat{n}_2 + \hat{n}_3) \right] - \xi \eta \zeta \left[ \hat{n}_1 \psi_i(A_1) + \hat{n}_2 \psi_i(A_2) + \hat{n}_3 \psi_i(A_3) \right], \]

\[ \lambda_3 = \hat{n}_1 \xi \eta \zeta + \hat{n}_2 \zeta (-1 + \eta + \xi) + \hat{n}_3 \eta (-1 + \zeta + \xi), \]

\[ \lambda_4 = \hat{n}_1 \eta \zeta (1 - \xi). \]

Formulas for the coefficients of the \( \hat{\psi} \) in the other two cases are similar to those of case 2.

**Four-edge cut element**

We now turn to the cases for four-edge cut elements. There are three possible four-edge cut elements, one of them is plotted in Figure 3.7. We will discuss only this type of four-edge cut elements since the construction procedure for its IFE local nodal basis function and the discussion on their properties can be readily applied to any of the other types four-edge cut elements.

![Figure 3.7: One of the three possible four-edge cut elements in the reference element \( \hat{T} \)](image)

In a four-edge cut interface element \( T \), the four intersection points \( P_i, i = 1, 2, 3, 4 \) may not, in general, belong to the same plane. So, in order to apply the solution flux continuity condition, we replace the physical interface surface \( \Gamma \cap T \) inside the interface tetrahedron by a plane that is a good approximation of \( \Gamma \cap T \). Many
procedures may be used to choose such an approximating plane. For example, one can neglect the intersection point that has the minimum normal distance from the plane made by the remaining three intersection points, and let $\hat{\Gamma}_T$ be the plane spanned by these remaining points. As usual, $\hat{\Gamma}_T$ is an $O(h^2)$ or better approximation to $\Gamma \cap T$ provided that $\Gamma \cap T$ is smooth enough. To be specific, we assume that $P_4$ is the point to be eliminated, and $\hat{\Gamma}_T$ is spanned by $P_1$, $P_2$ and $P_3$, see illustration in Figure 3.7. In this configuration, $P_4$ is not on $\hat{\Gamma}_T$. As before, we use the transformation (3.13) to map $T$ to the reference element $\hat{T}$ such that the images of $P_1$, $P_2$ and $P_3$ are

$$\hat{P}_1 = [\xi, 0, 0]^T, \quad \hat{P}_2 = [0, \eta, 0]^T, \quad \hat{P}_3 = [0, 1 - \zeta, \zeta]^T. \tag{3.18}$$

We need only to construct the basis functions in the reference element whose formula is given by (3.14). Then the local nodal IFE basis functions on $T$ is constructed by applying the nodal value specifications and the interface jump conditions as follows.

The nodal value specifications at the vertices:

$$a_{i,5} + a_{i,8} = \psi_i(A_1),$$
$$a_{i,6} + a_{i,8} = \psi_i(A_2),$$
$$a_{i,3} + a_{i,4} = \psi_i(A_3),$$
$$a_{i,4} = \psi_i(A_4).$$

The continuity at $P_1$, $P_2$ and $P_3$:

$$\xi a_{i,1} + a_{i,4} = \xi a_{i,5} + a_{i,8},$$
$$\eta a_{i,2} + a_{i,4} = \eta a_{i,3} + a_{i,8},$$
$$(1 - \zeta)a_{i,2} + \zeta a_{i,3} + a_{i,4} = (1 - \zeta)a_{i,6} + \zeta a_{i,7} + a_{i,8}.$$

The continuity of the flux across the approximate interface $\hat{\Gamma}_T$:

$$(a_{i,5} \hat{n}_1 + a_{i,6} \hat{n}_2 + a_{i,7} \hat{n}_3) \beta^- = (a_{i,1} \hat{n}_1 + a_{i,2} \hat{n}_2 + a_{i,3} \hat{n}_3) \beta^+. $$


These equations lead to the following formulas for coefficients of $\psi_i(r)$:

\[
\begin{align*}
    a_{i,1} &= \frac{\xi \psi_i(A_1) - \psi_i(A_4) + (1 - \xi)a_8}{\xi} \\
    a_{i,2} &= \frac{\eta \psi_i(A_2) - \psi_i(A_4) + (1 - \eta)a_8}{\eta} \\
    a_{i,3} &= \psi_i(A_3) - \psi_i(A_4) \\
    a_{i,4} &= \psi_i(A_4) \\
    a_{i,5} &= \psi_i(A_1) - a_{i,8} \\
    a_{i,6} &= \psi_i(A_2) - a_{i,8} \\
    a_{i,7} &= \frac{\eta \zeta \psi_i(A_3) - (1 - \eta)(1 - \zeta)\psi_i(A_4) + (1 - \eta - \zeta)a_{i,8}}{\eta \zeta} \\
    a_{i,8} &= \frac{\lambda_{i,1}\beta^- + \lambda_{i,2}\beta^+}{\lambda_3\beta^- + \lambda_4\beta^+}.
\end{align*}
\]

with

\[
\begin{align*}
    \lambda_{i,1} &= \xi \eta \zeta [\hat{n}_1 \psi_i(A_1) + \hat{n}_2 \psi_i(A_2) + \hat{n}_3 \psi_i(A_3)] - \hat{n}_3 \xi (1 - \eta)(1 - \zeta)\psi_i(A_4), \\
    \lambda_{i,2} &= \zeta [\hat{n}_1 \eta + (\hat{n}_2 + \hat{n}_3 \eta)\xi] \psi_i(A_4) - \xi \eta \zeta [\hat{n}_1 \psi_i(A_1) + \hat{n}_2 \psi_i(A_2) + \hat{n}_3 \psi_i(A_3)], \\
    \lambda_3 &= (\hat{n}_1 + \hat{n}_2)\eta \zeta + \hat{n}_3 \xi (-1 + \eta + \zeta), \\
    \lambda_4 &= \hat{n}_1 \eta \zeta (1 - \xi) + \hat{n}_2 \xi \zeta (1 - \eta).
\end{align*}
\]

### 3.4.5 Existence and Uniqueness

Then we can easily prove the following theorem about the existence and uniqueness of the local nodal basis functions.

**Theorem 3.1** Assume that $T$ is an interface element such that

\[
\lambda_3\beta^- + \lambda_4\beta^+ \neq 0, \quad (3.19)
\]

then the local nodal basis functions defined above in this interface element are uniquely determined by the nodal value specifications and the interface jump conditions.

The assumption in this theorem is true in many situations. For example, this is always true when vertex $A_4$ of $T$ is one corner of the cube to which $T$ belongs.
fact, in this situation, \( \hat{n} \) is the same as the normal of the plane passing the points \( \hat{P}_1, \hat{P}_2 \) and \( \hat{P}_3 \). For the first case of a three-edge cut interface element, we know that \( \hat{n} = [\hat{n}_1, \hat{n}_2, \hat{n}_3]^T \) is parallel to \( [\zeta, \eta \zeta, \eta \xi]^T \) and further we have

\[
\begin{align*}
\lambda_3 &= \eta (\xi \zeta)^2 + \xi (\eta \zeta)^2 + \zeta (\eta \xi)^2, \\
\lambda_4 &= (1 - \eta)(\xi \zeta)^2 + (1 - \xi)(\eta \zeta)^2 + (1 - \zeta)(\eta \xi)^2.
\end{align*}
\] (3.20) (3.21)

If (3.19) is not true, then we must have

\[
\begin{align*}
(\xi \zeta)^2[\eta \beta^- + (1 - \eta) \beta^+] &= 0, \\
(\eta \zeta)^2[\xi \beta^- + (1 - \xi) \beta^+] &= 0, \\
(\eta \xi)^2[\zeta \beta^- + (1 - \zeta) \beta^+] &= 0.
\end{align*}
\]

All of these together imply that at least two of the three variables \( \eta, \xi \) and \( \zeta \) must be zero which is not allowed by the three-edge cut definition.

For the second case of a three-edge cut interface element, we note that \( \hat{n} = [\hat{n}_1, \hat{n}_2, \hat{n}_3]T = [\eta \zeta, \zeta (-1 + \eta) + \xi (-1 + \zeta + \xi)]^T \) and further we have

\[
\begin{align*}
\lambda_3 &= \xi (\eta \zeta)^2 + \zeta^2 (-1 + \eta + \xi)^2 + \eta^2 (-1 + \zeta + \xi)^2, \\
\lambda_4 &= (\eta \zeta)^2(1 - \xi).
\end{align*}
\] (3.22)

If (3.19) is not true, then we must have

\[
\begin{align*}
(\eta \zeta)^2[\xi \beta^- + (1 - \xi) \beta^+] &= 0, \\
\zeta^2(-1 + \eta + \xi)^2 \beta^- &= 0, \\
\eta^2(-1 + \zeta + \xi)^2 \beta^- &= 0.
\end{align*}
\]

All of these together will force \( \hat{P}_1, \hat{P}_2 \) and \( \hat{P}_3 \) to be on the same face of \( \hat{T} \) which implies that \( P_1, P_2 \) and \( P_3 \) must be on the same face of \( T \) contradicting the definition.

Finally, for the four-edge cut configuration considered above, we know that \( \hat{n} \) is parallel to \( [\eta \zeta, \zeta \xi, \xi (-1 + \eta + \zeta)]^T \) and

\[
\begin{align*}
\lambda_3 &= (\eta + \xi) \zeta^2 \eta + \xi^2 (-1 + \eta + \zeta)^2, \\
\lambda_4 &= (\eta \zeta)^2(1 - \xi) + (\zeta \xi)^2(1 - \eta).
\end{align*}
\]

If (3.19) is not true, then we must have

\[
\begin{align*}
(\eta + \xi) \zeta^2 \eta + \xi^2 (-1 + \eta + \zeta)^2 &= 0, \\
(\eta \zeta)^2(1 - \xi) + (\zeta \xi)^2(1 - \eta) &= 0,
\end{align*}
\]
and we can easily see that these conditions lead to situations in which either two of $P_1, P_2, P_3$ are the same or these three points are on a line. Again, this contradicts with the definition that $T$ is a four-edge cut interface element.

To be specific, we let $\psi_i(x), i = 1, 2, 3, 4$ be such that

$$\psi_i(A_j) = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{if } i \neq j. \end{cases}$$

### 3.4.6 Partition of Unity and Consistency with Classical FEM

The local nodal IFE basis functions are closely related to their counterparts for the standard FE space. In particular, we note the two properties stated in the following theorem.

**Theorem 3.2** Assume that $T$ is an interface element such that (3.19) holds, then the local nodal basis functions in this interface element have the following properties:

- **Partition of unity:**
  $$\sum_{i=1}^{4} \psi_i(x) = 1, \quad \forall x \in T. \quad (3.23)$$

- **These IFE local nodal basis functions are consistent with the standard FE local nodal basis functions in the following sense: if the coefficient $\beta$ has no discontinuity, i.e., $\beta^- = \beta^+ = \beta$, then**
  $$\psi_i(x) = \overline{\psi}_i(x), \quad x \in T. \quad (3.24)$$

**Proof.** Using the definition of the linear IFE local nodal basis functions, we have

$$\sum_{i=1}^{4} \psi_i(x) = \begin{cases} \sum_{i=1}^{4} \psi^+_i(x) = \sum_{i=1}^{4} (a_{i,4} + a_{i,1} r + a_{i,2} s + a_{i,3} t), & x \in T^+ \\ \sum_{i=1}^{4} \psi^-_i(x) = \sum_{i=1}^{4} (a_{i,8} + a_{i,5} r + a_{i,6} s + a_{i,7} t), & x \in T^- \end{cases} \quad (3.25)$$
First, we have
\[ \sum_{i=1}^{n} a_{i,4} = \psi_1(A_4) + \psi_2(A_4) + \psi_3(A_4) + \psi_4(A_4) = 1. \]

For the three-edge cut case 1: We note that
\[ \sum_{i=1}^{4} \lambda_{i,1} = \lambda_3, \]
\[ \sum_{i=1}^{4} \lambda_{i,2} = \lambda_4. \]

Then
\[ \sum_{i=1}^{4} a_{i,8} = \frac{\sum_{i=1}^{4} \lambda_{i,1} \beta^- + \sum_{i=1}^{4} \lambda_{i,2} \beta^+}{\lambda_3 \beta^- + \lambda_4 \beta^+} = 1. \]

Further, we have
\[ \sum_{i=1}^{4} a_{i,1} = \frac{\xi \sum_{i=1}^{4} \psi_i(A_1) - \sum_{i=1}^{4} \psi_i(A_4) - (1 - \xi) \sum_{i=1}^{4} a_{i,8}}{\xi} = \frac{\xi - 1 + (1 - \xi)}{\xi} = 0, \]
\[ \sum_{i=1}^{4} a_{i,2} = \frac{\eta \sum_{i=1}^{4} \psi_i(A_2) - \sum_{i=1}^{4} \psi_i(A_4) + (1 - \eta) \sum_{i=1}^{4} a_{i,8}}{\eta} = \frac{\eta - 1 + (1 - \eta)}{\eta} = 0, \]
\[ \sum_{i=1}^{4} a_{i,3} = \frac{\zeta \sum_{i=1}^{4} \psi_i(A_3) - \sum_{i=1}^{4} \psi_i(A_4) + (1 - \zeta) \sum_{i=1}^{4} a_{i,8}}{\zeta} = \frac{\zeta - 1 + (1 - \zeta)}{\zeta} = 0, \]
\[ \sum_{i=1}^{4} a_{i,5} = \sum_{i=1}^{4} \psi_i(A_1) - \sum_{i=1}^{4} a_{i,8} = 1 - 1 = 0, \]
\[ \sum_{i=1}^{4} a_{i,6} = \sum_{i=1}^{4} \psi_i(A_2) - \sum_{i=1}^{4} a_{i,8} = 1 - 1 = 0, \]
\[ \sum_{i=1}^{4} a_{i,7} = \sum_{i=1}^{4} \psi_i(A_3) - \sum_{i=1}^{4} a_{i,8} = 1 - 1 = 0. \]

Then (3.23) follows by putting above results in (3.25). Similar arguments can be applied to show that (3.23) holds in other cases.
As for the consistence, without loss of generality, we consider the three-edge cut case 1, other cases can be proved similarly. First note that when $\beta^- = \beta^+ = \beta$, we have

$$a_{i,8} = \frac{\lambda_{i,1} + \lambda_{i,2}}{\lambda_3 + \lambda_4}$$

$$= \frac{\psi_i(A_1)(\hat{n}_1 \eta \zeta + \hat{n}_2 \xi \zeta + \hat{n}_3 \xi \eta)}{\hat{n}_1 \eta \zeta + \hat{n}_2 \xi \zeta + \hat{n}_3 \xi \eta} = \psi_i(A_4) = a_{i,4},$$

$$a_{i,1} = \frac{\xi \psi_i(A_1) - \psi_i(A_4) + (1 - \xi) a_{i,8}}{\xi}$$

$$= \psi_i(A_1) - \psi_i(A_4),$$

$$a_{i,2} = \psi_i(A_2) - \psi_i(A_4),$$

$$a_{i,3} = \psi_i(A_3) - \psi_i(A_4),$$

$$a_{i,4} = \psi_i(A_4),$$

$$a_{i,5} = \psi_i(A_1) - \psi_i(A_4),$$

$$a_{i,6} = \psi_i(A_2) - \psi_i(A_4),$$

$$a_{i,7} = \psi_i(A_3) - \psi_i(A_4).$$

Then we have

$$\hat{\psi}_i^+(r) = a_{i,4} + a_{i,1}r + a_{i,2}s + a_{i,3}t$$

$$= a_{i,8} + a_{i,5}r + a_{i,6}s + a_{i,7}t = \hat{\psi}_i^-(r),$$

which directly implies that

$$\psi_i^+(x) = \psi_i^-(x).$$

This together with the fact that

$$\psi_i(A_j) = \begin{cases} 1, & \text{when } i = j, \\ 0, & \text{when } i \neq j, \end{cases}$$

we can see that $\psi_i(x) = \overline{\psi}_i(x)$ which further implies (3.24)

### 3.5 Numerical Experiments

In this section, we report some numerical experiments using the IFE method developed in this chapter for solving full 3-Dimensional interface problems of the Poisson’s
equation. The interface problems in these experiments are chosen such that the interfaces are not trivial and the exact solutions are known, so we can assess and rate of convergence of the IFE method. Uniform Cartesian meshes $\mathcal{T}_h$ with mesh size $h$ are used in all the numerical experiments. Our numerical experiments indicate that the IFE method developed in this chapter has a rate of convergence comparable to the standard linear finite element method based on body-fit meshes.

### 3.5.1 An Interface Problem With a Spherical Interface

In the following experiment, the interface problem has the following solution domain in the 3-dimensional Cartesian space.

$$
\Omega = (-1, 1) \times (-1, 1) \times (-1, 1) \subset \mathbb{R}^3,
$$

which is separated into two sub-domains by a sphere $\Gamma$ of radius $r_0 = \frac{11}{7\pi}$ and centered at $(x, y, z) = (0, 0, 0)$ as shown in Figure 3.8.

The part of $\Omega$ inside the sphere is $\Omega^+$ and the rest of $\Omega$ is $\Omega^-$. The coefficient $\beta$ is a piece-wise constant function which is given by

$$
\beta(x) = \begin{cases} 
\beta^+, & r \leq r_0, \\
\beta^-, & r > r_0,
\end{cases}
$$

where

$$
r = \sqrt{x^2 + y^2 + z^2}
$$

The boundary conditions used are

$$
\Phi(-1, y, z) = g_{D1}(x, y, z), \quad \Phi(1, y, z) = g_{D2}(x, y, z),
$$

$$
\Phi(x, -1, z) = g_{D3}(x, y, z), \quad \Phi(x, 1, z) = g_{D4}(x, y, z),
$$

$$
\Phi(x, y, -1) = g_{D5}(x, y, z), \quad \Phi(x, y, 1) = g_{D6}(x, y, z).
$$

The functions $f, g_{D1}, g_{D2}, g_{D3}, g_{D4}, g_{D5},$ and $g_{D6}$ are chosen such that the BVP (3.1)–(3.3) has the following exact solution:

$$
\Phi(x) = \begin{cases} 
\Phi^-(x) = \frac{1}{\beta} r^\alpha, & r \leq r_0, \\
\Phi^+(x) = \frac{1}{\beta^2} r^\alpha + \left(\frac{1}{\beta^2} - \frac{1}{\beta^2}\right) r_0^\alpha, & r > r_0,
\end{cases}
$$

where

$$
r = \sqrt{x^2 + y^2 + z^2}, \alpha = 3
$$
We carried out several numerical simulations using the IFE solver with a sequence of Cartesian tetrahedral partitions of decreasing sizes.

First, the numerical interpolating accuracy of the IFE basis functions is investigated through evaluating the $L^2$ and $H^1$ interpolation errors defined as

\[ I_0 = \| \Phi - \tilde{\Phi} \|_0, \]
\[ I_1 = \| \Phi - \tilde{\Phi} \|_1, \]

where $\Phi$ is the exact solution and $\tilde{\Phi}$ is the interpolated solution. We perform the numerical experiment for two values of $\beta^+ / \beta^-$, namely 10 and 10,000. The results are provided in Table 3.1 and Table 3.2. In the following, $h$ is a measure of the partition size, which is selected herein to be max($\triangle x_{\text{max}}, \triangle y_{\text{max}}, \triangle z_{\text{max}}$).

Second, we investigate the accuracy of the 3 dimensional IFE method to solve the current interface problem through evaluating the $L^\infty$, $L^2$ and $H^1$ errors. The nu-
Table 3.1: $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+ / \beta^- = 10$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$I_0(h)$</th>
<th>$I_1(h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>0.02863074709784</td>
<td>0.60886262114381</td>
</tr>
<tr>
<td>1/25</td>
<td>0.01832005577936</td>
<td>0.48689457895889</td>
</tr>
<tr>
<td>1/15</td>
<td>0.01272535823591</td>
<td>0.40587074291109</td>
</tr>
<tr>
<td>1/20</td>
<td>0.00715981392140</td>
<td>0.30451088877053</td>
</tr>
<tr>
<td>1/25</td>
<td>0.00458276030469</td>
<td>0.24361267026969</td>
</tr>
<tr>
<td>1/30</td>
<td>0.00318292840028</td>
<td>0.20305190742799</td>
</tr>
<tr>
<td>1/40</td>
<td>0.00179054795537</td>
<td>0.15229665761824</td>
</tr>
</tbody>
</table>

Table 3.2: $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+ / \beta^- = 10,000$.

<table>
<thead>
<tr>
<th>$h$</th>
<th>$I_0(h)$</th>
<th>$I_1(h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>0.02862398910049</td>
<td>0.60894404495919</td>
</tr>
<tr>
<td>1/25</td>
<td>0.01831579080587</td>
<td>0.48689863165034</td>
</tr>
<tr>
<td>1/15</td>
<td>0.01272274827558</td>
<td>0.40587259699362</td>
</tr>
<tr>
<td>1/20</td>
<td>0.00715869552022</td>
<td>0.30451418926116</td>
</tr>
<tr>
<td>1/25</td>
<td>0.00458213262751</td>
<td>0.24360460779908</td>
</tr>
<tr>
<td>1/30</td>
<td>0.00318254744428</td>
<td>0.20305086088117</td>
</tr>
<tr>
<td>1/40</td>
<td>0.00179036875335</td>
<td>0.15229351567142</td>
</tr>
</tbody>
</table>

Numerical experiment is performed using the same values of $\beta^+ / \beta^-$, i.e. 10 and 10,000, and the results are provided in Table 3.3 and Table 3.4. Plots of the interpolation and solution errors are also shown in figure 3.3.

### 3.5.2 An Interface Problem With a Hemispherical Interface

Now, we examine a similar interface problem, but with the following solution domain in the 3-dimensional Cartesian space.

$$\Omega = (-1, 1) \times (-1, 1) \times (0, 1) \subset \mathbb{R}^3,$$
which is separated into two sub-domains by a hemisphere $\Gamma$ of radius $r_0 = \frac{11}{7\pi}$ and centered at $(x, y, z) = (0, 0, 0)$ as shown in Figure 3.10.

The part of $\Omega$ inside the hemisphere is $\Omega^+$ and the rest of $\Omega$ is $\Omega^-$. The coefficient $\beta$ is defined the same as in the above experiment. The boundary conditions used are

$$
\Phi(-1, y, z) = g_{D_1}(x, y, z), \quad \Phi(1, y, z) = g_{D_2}(x, y, z), \\
\Phi(x, -1, z) = g_{D_2}(x, y, z), \quad \Phi(x, 1, z) = g_{D_4}(x, y, z), \\
\Phi(x, y, 0) = g_{D_5}(x, y, z), \quad \Phi(x, y, 1) = g_{D_6}(x, y, z).
$$
The functions $f, g_{D1}, g_{D2}, g_{D3}, g_{D4}, g_{D5},$ and $g_{D6}$ are chosen such that the BVP (3.1)–(3.3) has the following exact solution:

$$\Phi(x) = \begin{cases} 
\Phi^-(x) = \frac{1}{\beta^2} r^\alpha, & r \leq r_0, \\
\Phi^+(x) = \frac{1}{\beta^2} r^\alpha + \left(\frac{1}{\beta^2} - \frac{1}{r_0^2}\right) r_0^\alpha, & r > r_0, 
\end{cases}$$

$$r = \sqrt{x^2 + y^2 + z^2}, \alpha = 3$$

Figure 3.9: Interpolation and solution errors of a spherical-interface problem.
The numerical simulations are carried out in this experiment in a similar fashion to that of the spherical interface experiment using two values for $\beta^+/\beta^-$, namely 10 and 10,000. The numerical interpolation and solution errors are also evaluated in the same way as above.

The interpolation errors are provided in Table 3.5 and Table 3.6 and the solution errors are provided in Table 3.7 and Table 3.8. Plots of the interpolation and solution errors are also shown in figure 3.11.

### 3.5.3 Numerical Error Analysis

From the results of the numerical analysis conducted in the 3 dimensional interface experiments and summarized in tables 3.1 through 3.8, we notice that the relation between the $L^\infty$, $L^2$ and $H^1$ errors for both interfaces, and the partition size $h$ can
Table 3.5: $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+ / \beta^- = 10$. 

<table>
<thead>
<tr>
<th>$h$</th>
<th>$I_0(h)$</th>
<th>$I_1(h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>0.01526655291243</td>
<td>0.38077157301392</td>
</tr>
<tr>
<td>2/25</td>
<td>0.00976136954738</td>
<td>0.30406200274363</td>
</tr>
<tr>
<td>1/15</td>
<td>0.00678042653309</td>
<td>0.2534689687812</td>
</tr>
<tr>
<td>1/20</td>
<td>0.00381496195444</td>
<td>0.19015432274360</td>
</tr>
<tr>
<td>1/25</td>
<td>0.00244181941472</td>
<td>0.15213094692520</td>
</tr>
<tr>
<td>1/30</td>
<td>0.00169589484799</td>
<td>0.12680351542069</td>
</tr>
<tr>
<td>1/40</td>
<td>0.00095400761812</td>
<td>0.09509529612487</td>
</tr>
</tbody>
</table>

Table 3.6: $L^2$ and $H^1$ interpolation errors of IFE functions generated with partitions of decreasing size $h$ and $\beta^+ / \beta^- = 10,000$. 

<table>
<thead>
<tr>
<th>$h$</th>
<th>$E_\infty(h)$</th>
<th>$E_0(h)$</th>
<th>$E_1(h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>0.00653992629705</td>
<td>0.01345544182078</td>
<td>0.33596149997161</td>
</tr>
<tr>
<td>2/25</td>
<td>0.00426605332010</td>
<td>0.00858970098315</td>
<td>0.26665430426189</td>
</tr>
<tr>
<td>1/15</td>
<td>0.00299254262135</td>
<td>0.00596785718525</td>
<td>0.22142076897879</td>
</tr>
<tr>
<td>1/20</td>
<td>0.00170715887587</td>
<td>0.00335098172302</td>
<td>0.16528868949120</td>
</tr>
<tr>
<td>1/25</td>
<td>0.00144258947210</td>
<td>0.00214061727808</td>
<td>0.1318303427550</td>
</tr>
<tr>
<td>1/30</td>
<td>0.00129672652667</td>
<td>0.001491095902063</td>
<td>0.10968845519083</td>
</tr>
<tr>
<td>1/40</td>
<td>0.00103930066509</td>
<td>0.00084168504315</td>
<td>0.08210823418007</td>
</tr>
</tbody>
</table>

Table 3.7: $L^\infty$, $L^2$ and $H^1$ errors of the IFE solutions generated with partitions of decreasing size $h$ and $\beta^+ / \beta^- = 10$. 

<table>
<thead>
<tr>
<th>$h$</th>
<th>$E_\infty(h)$</th>
<th>$E_0(h)$</th>
<th>$E_1(h)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/10</td>
<td>0.00653992629705</td>
<td>0.01345544182078</td>
<td>0.33596149997161</td>
</tr>
<tr>
<td>2/25</td>
<td>0.00426605332010</td>
<td>0.00858970098315</td>
<td>0.26665430426189</td>
</tr>
<tr>
<td>1/15</td>
<td>0.00299254262135</td>
<td>0.00596785718525</td>
<td>0.22142076897879</td>
</tr>
<tr>
<td>1/20</td>
<td>0.00170715887587</td>
<td>0.00335098172302</td>
<td>0.16528868949120</td>
</tr>
<tr>
<td>1/25</td>
<td>0.00144258947210</td>
<td>0.00214061727808</td>
<td>0.1318303427550</td>
</tr>
<tr>
<td>1/30</td>
<td>0.00129672652667</td>
<td>0.001491095902063</td>
<td>0.10968845519083</td>
</tr>
<tr>
<td>1/40</td>
<td>0.00103930066509</td>
<td>0.00084168504315</td>
<td>0.08210823418007</td>
</tr>
</tbody>
</table>
be put in the general regression form

$$ Error = a \ h^b \quad (3.26) $$

where $a$ and $b$ are two constants to be determined from the actual values of the interpolation and solution errors for each case. We can arrange the numerical analysis of the results of the previous examples in Table 3.9 and Table 3.10.

Table 3.9: Regression constants of the relation between interpolation error and mesh size.

The regression relations presented above provoke a second order convergence of the $L^2$ error and a first order convergence of the $H^1$ error. This is exactly true for the interpolation error of the IFE basis functions and approximately true for the numerical solution error of the IFE method.
Figure 3.11: Interpolation and solution errors of a hemispherical-interface problem.
<table>
<thead>
<tr>
<th>Model</th>
<th>β⁺/β⁻</th>
<th>$L^\infty$ error</th>
<th>$L^2$ error</th>
<th>$H^1$ error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$a$</td>
<td>$b$</td>
<td>$a$</td>
</tr>
<tr>
<td>Sphere</td>
<td>10</td>
<td>1.01</td>
<td>1.91</td>
<td>2.49</td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>3.8</td>
<td>2.23</td>
<td>5.77</td>
</tr>
<tr>
<td>Hemisphere</td>
<td>10</td>
<td>0.12</td>
<td>1.34</td>
<td>1.34</td>
</tr>
<tr>
<td></td>
<td>10,000</td>
<td>2.78</td>
<td>1.87</td>
<td>1.22</td>
</tr>
</tbody>
</table>

Table 3.10: Regression constants of the relation between IFE solution error and mesh size.
Chapter 4

Three-Dimensional IFE Field Solver

4.1 Introduction

In this chapter, we discuss the computational and programming aspects of the 3D IFE field solver which is used in all ES-PIC simulations in this work. First, we present the 3D Cartesian-based tetrahedral mesh used in the solver, the intersection of the mesh with the immersed objects and the implications of the mesh-object intersections. Then, we mention the details of constructing the IFE system. We also present the aspects of solving the resulting system of equations and the numerical methods applied to obtain the approximate field solution.

4.2 Mesh Generation

The IFE solver uses a Cartesian-based tetrahedral mesh. Tetrahedral elements are selected over other elements such as rectangular brick elements for the simplicity in the classification of possible intersection topologies between a mesh element and an object boundary. In fact, for a tetrahedral element, we could reduce all possible topologies into only two significant topologies.

The Cartesian-based tetrahedral mesh is constructed by first generating the base Cartesian mesh. Then, each cell in this Cartesian mesh is further partitioned into five
tetrahedra, as illustrated in the previous chapter. This makes the mesh generation trivial and quite fast.

4.3 Mesh-Object Intersection

To define the boundaries of immersed objects, it is possible to either provide this information through an input ascii file that contains a triangular surface mesh for each object which can be generated by Computer-Aided-Design (CAD) software, or by defining the algebraic equations that describes the object surfaces. The first choice makes the IFE solver more versatile since nearly any object shape can be generated. The second choice makes the IFE independent of any other software package, but limits our selection of objects to what have been predefined in the code. The IFE solver has been coded such that it can accept both ways of object definition. However, in this work we only consider the definition of objects by algebraic equations.

We have established a data base for most frequently used objects such as rectangular boxes, circular cylinders, spheres, thin plates, and so on. Each object surface is only checked against all the tetrahedra that lie in its vicinity. The intersection point of the object surface with each of the line segments, or edges, in each tetrahedra is determined, if existing.

4.3.1 Intersection Topology Classification

Typically an object surface will intersect with an interface tetrahedron in either three or four distinct points giving the type for the interface element as three-edge cut or four-edge cut. These intersection points normally do not lie on the same face or edge of the tetrahedron. Although, we may encounter situations in which this is not the case. Hence, the interface tetrahedron should be classified accordingly. To simplify the classification procedure, we use an index to classify the intersection topology.

Let $e_{ind}$ be an edge intersection index, which is defined as

$$e_{ind} = \begin{cases} 
1, & \text{if the intersection point lies between the edge ends}, \\
-1, & \text{if the intersection point lies at one of the ends}. 
\end{cases}$$

(4.1)

Note that the conditions in the above definition are satisfied within a preset geometric tolerance. Also, note that we only consider one intersection point per edge. Then,
we define a tetrahedron intersection index as

\[ t\text{ind} = \sum_{j=1}^{6} e\text{ind}_j \] (4.2)

Using the tetrahedron intersection index \( t\text{ind} \), we can classify the special intersection situations illustrated in figure 4.1 according to the rules in the look-up table 4.1.

Figure 4.1: Special situations of three-edge cut tetrahedron.
<table>
<thead>
<tr>
<th>tind</th>
<th>Classification of intersection topology</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>four-edge cut</td>
</tr>
<tr>
<td>3</td>
<td>three-edge cut</td>
</tr>
<tr>
<td>-1</td>
<td>three-edge cut</td>
</tr>
<tr>
<td>-4</td>
<td>three-edge cut</td>
</tr>
<tr>
<td>-6</td>
<td>non-interface</td>
</tr>
<tr>
<td>-5</td>
<td>non-interface</td>
</tr>
<tr>
<td>-3</td>
<td>non-interface</td>
</tr>
</tbody>
</table>

Table 4.1: Rules for classification of intersection topologies.

4.4 Assembly of the IFE System

Inside the IFE solver, the stiffness matrix, $K$, the right hand side (RHS) vector, $F$ and the mass matrix, $M$, (if the Newton-Raphson iteration is used to solve the nonlinear system) are all assembled. The stiffness matrix needs to be assembled only once, while the RHS vector and the mass matrix need to be assembled at each iteration inside the field solution loop.

4.4.1 Local Assembler

The finite element stiffness matrix, the mass matrix, and the RHS vector are all first assembled on each finite element. The local FE stiffness matrix is given by

$$K_{ij}^e = \int_{\Omega_e} \epsilon \nabla \psi_i \cdot \nabla \psi_j \, d\Omega ,$$

(4.3)

and the local RHS vector is given by

$$F_i^e = \int_{\Omega_e} \psi_i f(u) \, d\Omega ,$$

(4.4)

and the local mass matrix is given by

$$M_{ij}^e = \int_{\Omega_e} \psi_i \psi_j f'(u) d\Omega$$

(4.5)

where $f'(u) = \frac{\partial f}{\partial u}$.
4.4.2 Global Assembler

The global stiffness matrix, mass matrix and RHS vector are obtained by assembling the corresponding local matrices and vectors for all elements. The global stiffness matrix is then given by

\[ K_{ij} = \sum_{e=1}^{NOE} K_{ij}^e, \]  

(4.6)

where \( NOE \) is the number of finite elements. The global RHS vector is given by

\[ F_i = \sum_{e=1}^{NOE} F_i^e, \]  

(4.7)

and the global mass matrix is given by

\[ M_{ij} = \sum_{e=1}^{NOE} M_{ij}^e. \]  

(4.8)

4.4.3 Integration Rules

The construction of the local element stiffness matrix, RHS vector and mass matrix involves the evaluation of elemental volume integrals. Integration rules are usually referred to as quadrature rules. A quadrature rule has the form

\[ \int_{\Omega_e} g(\mathbf{x}) d\Omega \approx \sum_{k=1}^{NQP} W_k g(x_k) \]  

(4.9)

where \( g(x) \) is the integrand, \( NQP \) is the number of quadrature points, \( W_k \) is the weight corresponding to quadrature point \( k \), and \( x_k \) is the coordinates of the quadrature point. A quadrature rule is said to be exact to order \( q \), if it is exact when the function \( g(x_k) \) is a polynomial of degree \( q \) or less.

Gaussian Quadratures

Among quadrature rules, Gaussian quadratures are preferred for finite element applications because they have fewer function evaluations for a given order. With
Gaussian quadrature, the weights and evaluation points are determined so that the integration rule is exact to as high an order as possible. In the IFE solver, four-point Gauss quadrature rules are applied to calculate the numerical integrations on tetrahedral elements which have the form

\[ \int_{\Omega_e} g(x) d\Omega \approx V \sum_{k=1}^{4} w_k g(r_k) \]  (4.10)

where \( g(x) \) is the integrand, \( V \) is the element volume, \( w_k \) is the weight corresponding to quadrature point \( k \), \( x_k \) is the physical coordinates of the quadrature point, and \( r_k = [r_k, s_k, t_k]^T \) is the tetrahedral coordinates of the quadrature point. Note that in the above integration rule, we assume that \( \sum_{k=1}^{4} w_k = 1 \). The error in these rules is of order \( O(h^3) \), where \( h \) is the size of the tetrahedral element. The weights and coordinates of the applied four-point Gaussian quadrature rules are provided in table 4.2 (see for example [31]).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( w_k )</th>
<th>( r_k )</th>
<th>( s_k )</th>
<th>( t_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.25</td>
<td>0.58541019662496854</td>
<td>0.13819660112501052</td>
<td>0.13819660112501052</td>
</tr>
<tr>
<td>2</td>
<td>0.25</td>
<td>0.13819660112501052</td>
<td>0.58541019662496854</td>
<td>0.13819660112501052</td>
</tr>
<tr>
<td>3</td>
<td>0.25</td>
<td>0.13819660112501052</td>
<td>0.13819660112501052</td>
<td>0.58541019662496854</td>
</tr>
<tr>
<td>4</td>
<td>0.25</td>
<td>0.13819660112501052</td>
<td>0.13819660112501052</td>
<td>0.13819660112501052</td>
</tr>
</tbody>
</table>

Table 4.2: Weights and quadrature points for integrations on tetrahedral elements.

Since all physical quantities are only available at IFE mesh nodes, we need to use the elements basis functions to interpolate these quantities onto quadrature points. Thus, we can write

\[ u(x) = \sum_{i=1}^{NEN} u_i \psi_i(x) \]  (4.11)

**Integration on Interface Elements**

Each Interface element is divided by the interface surface into two sub-elements, each of which has its own material property. Hence, linear local basis functions
defined on a non-interface element are piece-wise linear. Thus, numerical integration over a non-interface element, should be done piece-wisely. To use Gauss quadrature rule for tetrahedral elements, each sub-element is further partitioned into a number of tetrahedra depending on the topology of the sub-element [see figures 4.2]. The detailed partitioning of the sub-elements is shown in figure 4.3 for a three-edge cut element and figure 4.4 for a four-edge cut cut element.

\[ (a) \text{ 3-edge-cut interface tetrahedron} \quad (b) \text{ 4-edge-cut interface tetrahedron} \]

Figure 4.2: Partitioning of typical interface tetrahedra into sub-tetrahedra.

The integration of the local mass matrix, for example can be done as

\[ M_{ij} = \sum_{p=1}^{NSP} \sum_{k=1}^{NQP} \psi_i(x_k^p)\psi_j(x_k^p)f'(u_k^p) \]

(4.12)

where \( NSP \) is the number of the sub-tetrahedral partitions, \( V \) is the volume of the sub-tetrahedral partition \( p \), \( x_k^p \) is the coordinate of the quadrature \( k \) in the sub-tetrahedral partition \( p \), and \( u_k^p \) is the value of the function \( u \) at \( x_k^p \). The assembly of the stiffness matrix and the RHS vector can be done similarly.

4.4.4 Sparse Storage of the System Matrix

In the IFE field solver, we assemble the global stiffness matrix. Although not necessary, the assembly of the global stiffness matrix introduces a valuable reduction in
the overall computational time of the field solution. The global stiffness matrix for a moderate size 3-dimensional domain, for example \((50 \times 50 \times 50)\), has a number of entries of about \(125,000 \times 125,000\). This requires a storage space of about 116 GB, if we use 8-Byte double precision declaration! Of course, this is an extremely huge storage space which is beyond the storage capacity of any available single PC or workstation, especially if we notice that this space is required only for the storage of the stiffness matrix. Fortunately, we do not have to assign this astronomically huge space for the stiffness matrix, since the stiffness matrix is quite sparse and nearly full of zero entries.

The correct storage format is one which stores only the non-zero entries in the stiffness matrix which are much less than the total number of entries (less than 0.02% for the above example). The storage format must also be good for matrix operations [63]. The sparse storage scheme used in the IFE field solver is a compressed row storage
(CRS) format which is a quite common one [60]. The system matrix is stored as a data structure which contains three one-dimensional arrays, $K$, $JCOL$ and $SROW$. $K$ is an 8-Byte double precision array which includes the values of the nonzero entries in the system matrix arranged row by row. $JCOL$ is a 4-Byte integer array which includes the column indices of the nonzero entries. Both arrays are of length $NZ$, where $NZ$ is the number of the non-zero entries in the system matrix. $SROW$ is a 4-Byte integer array of length $N + 1$ which includes pointers to the starting locations of each row in $K$, where $N$ is the order of the system matrix. The last entry of $SROW$ is $SROW(1) + NZ$. Of course the symmetry of the system matrix, if considered, can introduce a further storage reduction. However, this is not used here in order to speed up the matrix-vector multiplications associated with the preconditioned conjugate gradient solver. This will be described later in this chapter.
4.5 Solution of the Nonlinear Field Problem

Using the definition of the local nodal linear basis functions for both interface and non-interface elements, the potential function $\Phi(x)$ can be approximated by

$$\Phi(x) \approx \sum_{j=1}^{N} \Phi_j \Psi_j(x)$$  \hspace{1cm} (4.13)

where $N$ is the number of mesh nodes and $\Phi_j$ is the value of the potential at node $j$. After substituting into the weak formulation of the field interface BVP, we get the following system of nonlinear algebraic equations.

$$K_{ij} \Phi_j = F_i(\Phi_l)$$  \hspace{1cm} (4.14)

where

$$K_{ij} = \int_{\Omega} \epsilon \nabla \Psi_i \cdot \nabla \Psi_j \, d\Omega,$$  \hspace{1cm} (4.15)

$$F_i(\Phi_l) = \int_{\Omega} \Psi_i f(\Phi_l) \, d\Omega,$$  \hspace{1cm} (4.16)

$$i, j, l = 1, 2, ..., N$$  \hspace{1cm} (4.17)

There are many iterative methods to solve such a nonlinear algebraic system. We studied two candidate methods:

1. the Gauss-Seidel iterative method, and

2. the Newton-Raphson iterative method.

4.5.1 Gauss-Seidel Iteration

The first candidate is the block Gauss-Seidel iterative method. In which, we rearrange the mesh nodes such that the unknown vector $\Phi$ can be split into two vectors $\Phi^+$ and $\Phi^-$ associated with the sub-domains $\Omega^+$ and $\Omega^-$ respectively. Then the nonlinear system of equations can be permuted as

$$K \cdot \Phi = F(\Phi) = \left\{ \begin{array}{l} F^+(\Phi) \\ F^- (\Phi) \end{array} \right\},$$

$$K = \begin{bmatrix} K^{++} & K^{+-} \\ K^{-+} & K^{--} \end{bmatrix}$$  \hspace{1cm} (4.18)
Now, the block Gauss-Seidel iteration is carried out as follows. Assuming the \( n \)-th iteration \( \Phi^n \) has been found, then we can first solve for \( [\Phi^+]^{n+1} \) from
\[
K^{++} \cdot [\Phi^+]^{n+1} = F^+(\Phi^n) - K^{+-} \cdot [\Phi^-]^n
\]
and then solve for \( [\Phi^-]^{n+1} \) from
\[
K^{--} \cdot [\Phi^-]^{n+1} = F^-(\Phi^n) - K^{-+} \cdot [\Phi^+]^n
\]
This way, the iteration \( n+1 \) is obtained from iteration \( n \) by solving two smaller linear systems which can be solved efficiently using the preconditioned conjugate gradient method for instance. As we notice, no matrix assembly is required at each iteration yet the right hand side vector \( F \) needs to be assembled and updated.

The block Gauss-Seidel method is superior in the aspects of storage and computational efficiency. However, in some numerical situations, the method becomes marginally stable, or even unstable. This behavior of the block Gauss-Seidel method has been noticed in both ion optics and thruster–spacecraft interaction simulations.

### 4.5.2 Newton-Raphson Iteration

The second candidate method is the Newton-Raphson iterative method. We apply a single-step Newton-Raphson linearization to the system of simultaneous nonlinear algebraic equations (4.14). Hence, we obtain the following system of iterative linear equations
\[
\begin{bmatrix}
K_{ij} - \frac{\partial F_i(\Phi^n_l)}{\partial \Phi_j}
\end{bmatrix}
(\Phi^{n+1}_j - \Phi^n_j) = F_i(\Phi^n_l) - K_{ij} \Phi^n_j
\]
where \( i, j, l = 1, 2, \ldots, N \) and \( n \) is the number of the previous iterations. It is worth noting, here, that although a Newton’s method requires the assembly of the system tangent stiffness matrix in equation (4.22) at each iteration as compared to e.g. a Gauss-Seidel iteration, the Newton’s method proved to be stable in handling almost all problem situations encountered [34]. To overcome the computational overload accompanying the assembly of the tangent stiffness matrix, all element-level computations are hardwired to increase the computational speed of the matrix assembly.
4.6 Solution of the Sparse Linear/Linearized System

The system that results from the FE discretization of the interface boundary value field problem (IBVFP) is a large sparse system which is either linear in nature or linearized by numerical schemes. Thus, we need to apply numerical methods which are able to efficiently handle such large sparse linear systems. Direct matrix methods are not considered here because the system matrix is very large and sparse, but unstructured.

4.6.1 Preconditioned-Conjugate Gradient (PCCG) Solver

The Conjugate Gradient (CG) method is one of the best known iterative techniques for solving sparse symmetric positive definite linear systems [63]. It is the oldest and best known of the non-stationary methods. The method proceeds by generating vector sequences of iterates (i.e., successive approximations to the solution), residuals corresponding to the iterates, and search directions used in updating the iterates and residuals. Although the length of these sequences can become large, only a small number of vectors needs to be kept in memory. In every iteration of the method, two inner products are performed in order to compute update scalars that are defined to make the sequences satisfy certain orthogonality conditions. On a symmetric positive definite linear system these conditions imply that the distance to the true solution is minimized in some norm. The iterates are updated in each iteration by a multiple of the search direction vector.

4.6.2 Preconditioners

The CG method is most often used in combination with a suitable approximation $K$ for the system matrix $A$. $K$ is called the preconditioner and is chosen to be positive definite as well as $A$. Therefore, the CG is to be applied to the system $K^{-1}Ax = K^{-1}b$. However, this may not be always possible, because $K$ is most likely not symmetric, and hence $K^{-1}A$. The way out of this, as suggested by [76], is to introduce a non-standard inner product, with respect to whom $K^{-1}A$ will be symmetric positive definite. The algorithm for the Preconditioned Conjugate Gradient Method to solve a sparse linear system $Ax = b$ using a preconditioner $K$
is given in figure 4.5.

\[ x_0 \text{ is an initial guess, } r_0 = b - Ax_0 \]
\[ \text{for } i = 1, 2, ..., \]
\[ \text{Solve } Kw_{i-1} = r_{i-1} \]
\[ \rho_{i-1} = r_{i-1}^H w_{i-1} \]
\[ \text{if } i = 1 \]
\[ p_i = w_{i-1} \]
\[ \text{else} \]
\[ \beta_{i-1} = \rho_{i-1}/\rho_{i-2} \]
\[ p_i = w_{i-1} + \beta_{i-1} p_{i-1} \]
\[ \text{end if} \]
\[ q_i = Ap_i \]
\[ \alpha_i = \rho_{i-1}/p_i^H q_i \]
\[ x_i = x_{i-1} + \alpha_i p_i \]
\[ r_i = r_{i-1} - \alpha_i q_i \]
\[ \text{if } x_i \text{ is accurate enough then quit} \]
\[ \text{end.} \]

Figure 4.5: Preconditioned conjugate gradient algorithm.

Two preconditioners have been tried: 1) an incomplete Cholesky decomposition preconditioner, and a Jacobi diagonal preconditioner.

### Incomplete Cholesky Decomposition

The incomplete Cholesky (IC) decomposition can be used as a preconditioner for the system matrix. Here, we use an incomplete Cholesky decomposition with no fill-in (IC(0)). The algorithm for the IC(0) is shown in figure 4.6. The IC(0) preconditioner is very expensive to construct. However, it usually requires fewer number of iterations as compared to a diagonal preconditioner.

### Jacobi Diagonal Preconditioner

We can simply use the diagonal of the system matrix \( \text{diag}(A) \) as an approximation for \( A \). The resulting preconditioned CG method is called Jacobi-CG method. It is always recommended to try the diagonal preconditioner which, despite its simplicity, can be very effective in approximating the solution in many practical problems.

In fact, in our situation, we found that using IC(0) leads to a much longer CPU time than the diagonal preconditioner despite the number of iterations is reduced
for $k = 1, 2, \ldots, n - 1$
\[ d = 1/a_{k,k} \]
for $i = k + 1, k + 2, \ldots, n$
\[ e = d a_{i,k}; a_{i,k} = e \]
for $j = k + 1, k + 2, \ldots, n$
\[ a_{i,j} = a_{i,j} - e a_{k,j} \]
end if
end j
end if
end i
end k.

Figure 4.6: Incomplete Cholesky decomposition preconditioner.

by nearly one order of magnitude. Therefore, we decided to use a simple diagonal preconditioner in the field solver instead of the IC(0) preconditioner.

4.7 Hardwiring The IFE Field Solver

In the hardwired IFE solver, all the computations that can be done at the finite element level and are independent of the element specific size, orientation and skewness properties - are done only once and stored in an appropriate data structures for later usage in the IFE global assembly. This drastically reduces the computational overhead required by the IFE solver.

Because of the programming complexities associated with the local assembly over interface elements, integrations will not be hardwired. However, this will not affect the computational efficiency of the solver, because the interface elements are much fewer than non-interface elements.

Hardwired Local Assembler

The local stiffness matrix can be written as
\[
K_{ij}^e = \mathcal{V} \epsilon \sum_{k=1}^{NQP} \sum_{d=1}^{ND} w_k \frac{\partial \psi_i(x_k)}{\partial x_d} \frac{\partial \psi_j(x_k)}{\partial x_d} \\
= \sum_{d=1}^{ND} \frac{\epsilon \mathcal{V}}{h_d^2} \left[ \sum_{k=1}^{NQP} w_k \frac{\partial \psi_i(x_k)}{\partial \hat{x}_d} \frac{\partial \psi_j(x_k)}{\partial \hat{x}_d} \right] 
\]

(4.23)

where \(d\) is the coordinate index, \(ND\) is the number of space dimensions, and \(\hat{x}_d = \frac{x_d}{h_d}\). We note that the quantity between square brackets in equation (4.23) is independent of the element size \(\mathcal{V}\), stretching parameters \((h_d, d = 1, 2, ..., ND)\), and material property \(\epsilon\), hence it will be calculated for the elements in a unit-cell, and stored for later usage in the assembly of the non-interface element local stiffness matrices.

The local RHS vector can be written as

\[
F^e_i = \mathcal{V} \sum_{k=1}^{NQP} [w_k \psi_i(x_k)] f(u_k) 
\]

(4.24)

where

\[
u_k = \sum_{i=1}^{NEN} u_i \left[ \psi_i(\vec{x}_k) \right]
\]

Finally, the local mass matrix can be written as

\[
M_{ij}^e = \mathcal{V} \sum_{k=1}^{NQP} [w_k \psi_i(x_k) \psi_j(x_k)] f'(u_k) 
\]

(4.25)

where \(f'(u_k) = \frac{\partial f(u_k)}{\partial u}\). Here, also, the quantities between square brackets are element-independent.
Chapter 5

The Hybrid-Grid IFE-PIC Model

5.1 Introduction

In this chapter, we first introduce the PIC method based on which the HG-IFE-PIC is developed. We describe how particles and fields are initialized, and how boundary conditions are setup. The diagnoses of particle and field quantities are also addressed. Although the PIC algorithm itself is applicable to both electrostatic (ES) and electromagnetic (EM) problems, we will limit our discussion to ES problems. This is because all the simulation problems we are concerned with in this work have a negligible time rate of change of the accompanied magnetic field. The details of the physical models are left to other chapters.

Second, we present the IFE-PIC model which incorporates the three-dimensional IFE field solver developed in chapter 4 and a modification of a legacy PIC model. The most attractive feature of the new model is that it retains both the second order accuracy of IFE field solver and the fast particle pushing of a Cartesian PIC code.

Finally, we introduce a new meshing technique which is applied to enhance the meshing capabilities of the IFE-PIC code. We allow the IFE and the PIC mesh nodes to be displaced from each other instead of being collocated. The PIC Cartesian mesh is kept uniform to retain the efficiency in determining the particles locations. The IFE mesh, which is used to solve the field, is stretched according to the local potential gradients and plasma conditions. This allows us to use a much less number of elements and mesh nodes for the same problem which implies less memory storage and computational time.
5.2 Theory

5.2.1 Plasma

A plasma, as defined in [28], is a hot, fully ionized gas which may be regarded as a collection of positive ions and negative electrons interacting through their mutual electromagnetic fields \((\mathbf{E} \text{ and } \mathbf{B})\). The fields are related to the charge and current density \((\rho \text{ and } \mathbf{j})\) by the Maxwell’s equations which can be written in vacuum as

\[
\nabla \cdot \mathbf{B} = 0, \\
\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}, \\
\nabla \cdot \mathbf{E} = \rho / \epsilon_0, \\
\nabla \times \mathbf{E} = \frac{\partial \mathbf{B}}{\partial t},
\]

where the speed of light in free space \(c = 1/\sqrt{\epsilon_0 \mu_0}\). In many applications, the induced magnetic fields are negligibly small and the electrostatic approximation is sufficient. Under such approximation, the Maxwell’s equations reduce to [28]

\[
\nabla \cdot \mathbf{B} = 0, \\
\nabla \times \mathbf{B} = 0, \\
\nabla \cdot \mathbf{E} = \rho / \epsilon_0, \\
\nabla \times \mathbf{E} = 0.
\]

Since \(\nabla \times \mathbf{E} = 0\), the electrostatic potential \(\Phi\) can be introduced such that

\[
\mathbf{E} = -\nabla \Phi,
\]

and hence

\[
\nabla \cdot \mathbf{E} = -\nabla \cdot \nabla \Phi = -\nabla^2 \Phi = \rho / \epsilon_0.
\]

Thus, the Maxwell’s equations reduce to the familiar Poisson’s equation, which can be also written for a general medium with permittivity \(\epsilon\) as

\[
\nabla \cdot (\epsilon \mathbf{E}) = -\nabla \cdot \epsilon \nabla \Phi = \rho.
\]

The force on a plasma particle carrying a charge \(q\) and moving through electromagnetic fields with a velocity \(\mathbf{v}\) is given by the Lorentz’ force law

\[
\mathbf{F} = q (\mathbf{E} + \mathbf{v} \times \mathbf{B}),
\]
and the motion of the charged particle is determined by Newton’s second law

$$m \frac{dv}{dt} = F.$$  \hspace{1cm} (5.13)

The charge density and current density in a small enough volume $V$, occupied by $N$ charged particles with charges $q_1, q_2, \ldots, q_N$, may be calculated from

$$\rho = \frac{1}{V} \sum_{i=1}^{N} q_i,$$  \hspace{1cm} (5.14)

$$j = \frac{1}{V} \sum_{i=1}^{N} q_i v_i.$$  \hspace{1cm} (5.15)

### 5.2.2 Debye Shielding and Plasma Sheath

One of the most important properties of a plasma is that it can shield out electric fields over a certain length scale so that the plasma remains quasi-neutral. This length scale is called the Debye length and denoted by $\lambda_D$. When a charged object is introduced into a plasma, the $E$-field of the object is shielded out over a region called the sheath. The sheath thickness is typically several Debye lengths. For length scales smaller than the Debye length, the charged particles behave as individual particles subject to electrical effects. On longer scales, collective coupling of particles can occur. According to this fundamental length scale, a plasma-surface interaction is considered electrically-coupled if $\lambda_D >> L$ and electrically-uncoupled if $\lambda_D << L$, where $L$ is a characteristic length of the problem [24].

### 5.3 The Particle–In–Cell Method

Since the space scale of interest is of the order of the Debye length and the orbits of individual ions and electros are important, the simulation of ES plasma is well rendered by the Particle-in-Cell (PIC) method [7]. In a PIC model, real plasma particles are represented by much fewer number of simulation particles. Simulation particles are usually referred to as super, or macro particles. Each simulation particle is assigned the total charge and mass of many real particles such that the overall charge and mass of the real and simulation systems are equal.
The PIC method samples the space charge density on discrete mesh nodes given the continuous positions of particle charges. Knowing the space charge density, the Poisson’s equation can be solved for the electric field. Obtaining the fields, the electric force at each particle position can be calculated from the electric fields evaluated at mesh nodes. The equations of motion of simulation particles could, then, be integrated to update the velocity and position of each particle. A typical computational cycle of a PIC code consists of four main steps:

1. **Particle Push** or integration of the equations of motion.
2. **Charge Deposit** or weighting the particle charges from particle positions to mesh nodes.
3. **Field Solve** or solving the electrostatic field given the imposed boundary conditions and the space charge density.
4. **Force Weighting** or weighting the forces from mesh nodes to particle positions.

Each of these steps will be elaborated in the following.

### 5.3.1 Particle Push

A typical PIC code may call the particle pusher subroutine thousands of times for each one of typically millions of particles. Therefore, the equations of motion must be integrated using the fastest possible scheme which should also retain acceptable accuracy. In addition, the scheme applied should require the least possible storage per integration per particle. The most commonly used integration scheme is the *leap-frog* scheme. The motion of each particle can be described by the following equations

\[
\frac{mv}{\Delta t} = F \quad (5.16)
\]
\[
\frac{dx}{dt} = v \quad (5.17)
\]

These equations may be approximated by the following two time-centered finite difference (FD) formulations which are shifted by a half time step.

\[
m\frac{v^{n+1/2} - v^{n-1/2}}{\Delta t} = F^n \quad (5.18)
\]
\[
x^{n+1} - x^n \quad \frac{\Delta t}{\Delta t} = v^{n+1/2} \quad (5.19)
\]
These two FD equations together form the leap frog scheme which is illustrated in figure [5.1]. The leap frog method is second order accurate and has a numerical error which vanishes as $\Delta t \to 0$ [7].

**5.3.2 Charge Deposit**

It is necessary to calculate the charge density in the discrete field mesh points from the continuous particle positions in order to provide the field solver with the space charge density. The sampling of the particle charges on the discrete mesh is called weighting. Several weighting techniques may be applied. In the PIC method, a first-order weighting scheme is applied which smoothes the density and field fluctuations. In a three dimensional space, each particle has a rectangular parallelepiped shape with the same dimensions and orientation as a PIC mesh cell. As the particle moves inside a PIC cell, its total charge is assigned to the eight surrounding vertices of that cell. The portion of the total particle charge which is assigned to a certain cell vertex is proportional to the volume of the rectangular parallelepiped included between the opposite cell vertex and the particle position. This is illustrated for a two-dimensional situation in figure [5.2]. In a three-dimensional domain, the contribution of the particle $p$, which is located at position $x_p$, for instance, to the charge density at node $(i, j, k)$ is calculated from

$$q_{i,j,k} = q_p \frac{V(x_{i+1,j+1,k+1}, x_p)}{V(x_{i,j,k}, x_{i+1,j+1,k+1})} \quad (5.20)$$
5.3.3 Field Solve

When the charge density is known at the mesh nodes, we can solve the Poisson’s equation \((7.11)\) for the electric potential which may be numerically differentiated to obtain the electric field. The Poisson’s equation is an \textit{elliptic} PDE in space. It requires boundary condition information about the potential or potential gradient (electric field) over all domain boundaries. Many numerical techniques have been used long time ago to tackle the Poisson’s equation either through finite difference or finite element methods. Iterative and direct solvers are available, but for all significant applications, the size of the problem makes the application of direct solvers unfeasible. Fast iterative solvers, like Fast Fourier Transforms (FFT), can be applied for problems with periodic boundary conditions which are not considered here because of the physical nature of our problems.

An efficient method for the solution of the Poisson’s equation, is a necessary requirement for the practical implementation of the PIC algorithm. A rule of thumb for a PIC model with a well-balanced calculations is to equally divide computational time between pushing particles and solving the field \([28]\). It is well-known that the methods available for the solution of the partial differential equations (PDE) prescribing the fields depend very much on the details of the equations and there is no single method that is universally the best.

It is shown elsewhere in this dissertation, that the total space charge density \(\rho\) is
usually a function of the local potential function, based on the assumed Boltzman’s
electron distribution model. Accordingly, the Poisson’s equation is \textit{nonlinear}. Also, the geometrical and physical nature of the simulation problem enforces a \textit{three-dimensional} analysis of the problem.

We have chosen the immersed finite element (IFE) method to build our field solver because of the high accuracy it provides at an affordable additional computational cost, when arbitrary-shaped objects are immersed in a Cartesian mesh. The motivation for using the IFE is elaborated in chapter 3.

\subsection{Force Weighting}

In a PIC model, the electrostatic forces are weighted from the the calculated field values at mesh nodes to the arbitrary particles positions. It is strongly recommended to use the same weighting method as in the charge deposit step which greatly reduces the numerical noise associated [7].

\subsection{Particles Initial and Boundary Conditions}

The initial position and velocity distribution of simulation particles and the treatment of the particles as they hit object surfaces or the boundaries of the simulation domain are very crucial issues in particle simulation that have to be carefully considered.

\subsubsection{Particles Loading}

The initial loading of particles in the simulation domain is not crucial if only the steady-state solution is what we seek from the particle simulation. However, if the particles are wisely loaded, this may reduce the computational time till steady-state effectively. For example, in ion optics simulations particles may be loaded from the upstream boundary and all the way down to the upstream face of the screen grid.
5.4.2 Particles Injection

As the simulation goes and particles emigrate through the simulation domain, new particles have to be injected from the appropriate sides to fill in the space left empty. If one of the simulation domain sides is immersed in equilibrium plasma, then the particles should be injected from this side with a drifted Maxwellian velocity distribution as given by equation.

A Maxwellian velocity distribution can be generated numerically in the PIC code using the method described in [7]. In such method, a Maxwellian (or Gaussian) distribution can be generated from a set of uniform random numbers, $R_1, R_2, \ldots, R_M$ with $R_i \in ]0, 1[$, $i = 1, 2, \ldots, M$ according to

$$v_M = v_\text{th} \left( \sum_{i=1}^{M} R_i - \frac{M}{2} \right) \left( \frac{M}{12} \right)^{-1/2}. \quad (5.21)$$

For this distribution, the calculated maximum velocity is $v_{\text{max}} = v_\text{th} \sqrt{3M}$. In the PIC model developed in this work, we use $M = 3$.

5.4.3 Particles Boundary Conditions

When a particle hits the surface of an object or one of the sides of the simulation domain, it is normally absorbed. If the hit surface is a plane of symmetry, then the particles is reflected back to the domain with reflection of the velocity on the surface. If the problem is periodic, so if a particle hits one of the domain sides, it is injected back from the opposite side with the same velocity. So, the boundary conditions of the particles are: absorption, reflection, or periodic.

Absorption

If a particle hits an absorbing surface it is simply taken away from the simulation domain and its associated data is removed from the particle array. Absorption BC is applied when particles are allowed to freely migrate out of the simulation domain in the lack of any physical condition to send them back. It is also applied when the a charged particle is neutralized due to collision with an electrically conducting solid surface since neutral particles do not contribute to the space charge of the domain.
Reflection

Two types of particle reflections are considered here; specular reflection and diffuse reflection.

Specular Reflection is perfectly elastic with the particle velocity component normal to the surface being reversed, while that parallel to the surface is maintained. This is applied on symmetric boundary surfaces.

Diffuse Reflection is assumed when the particles hitting a solid surface are reflected back to the simulation domain. The velocity of each particle after reflection is independent of its incident velocity. However, the velocities of the reflected particles as a whole are distributed according to the half-range equilibrium Maxwellian distribution for the particles that are directed away from the surface.

The two types of reflection boundary conditions are illustrated in figure 5.3. In the simulations conducted in this work, we always assume specular particle reflections on planes of symmetry. Particles hitting solid surfaces are absorbed and taken away from the simulation domain.

Figure 5.3: Reflection particle boundary conditions.
Periodic boundary conditions are applied when the simulations domain represents a repeating unit in the physical problem. Particles emerging out of a periodic boundary surface are sent back to the simulation domain but from the opposite periodic boundary surface. Particle velocities and in-plane locations are maintained.

5.5 The IFE–PIC Model

The use of a uniform Cartesian mesh in a PIC code makes the process of weighting the field and depositing the particle charge quite easy and fast, because the PIC cell where each particle is located in can be trivially determined via indexing. This is why many of the PIC codes developers prefer to use a uniform Cartesian mesh [9, 57, 20].

Typically, the mesh nodes of the PIC code and the IFE solver are collocated such that the physical quantities are deposited from particle locations onto PIC mesh nodes, then they are automatically copied onto IFE mesh nodes (see figure 5.4 for illustration). However, this technique has the drawback of over-refining the mesh in applications involving very non-uniform plasma. In the following, we suggest a new meshing concept that was applied to enhance the meshing capabilities of the IFE-PIC code.

![Interpolation procedure for a collocated IFE-PIC mesh.](image-url)
5.6 The Concept of Hybrid–Grid

The essence of the IFE–PIC code is nearly preserved in the new version of the code. However, in the new version, we introduce a new meshing technique in which we allow the IFE and the PIC mesh nodes to be displaced from each other instead of being collocated according to the physical and computational needs of the problem of concern. The uniformity of the PIC Cartesian mesh is kept to attain the easiness and fastness for determining the particles locations. The IFE mesh, which is used to solve the field, is stretched according to the potential gradients as imposed by physics. This allows us to use a much less number of elements and mesh nodes for the same problem which implies less memory storage and computational time.

5.6.1 IFE Mesh Stretching

In the following, we assume that the IFE mesh is a Cartesian-based tetrahedral stretched mesh, while the PIC mesh is still a Cartesian uniform mesh. Assume each IFE mesh cell is a (generalized cube), which has the dimensions $h_1$, $h_2$, ..., and $h_{ND}$ in the $x_1$, $x_2$, ..., and $x_{ND}$ directions respectively, where $ND$ is the number of space dimensions. In a two-dimensional space ($ND = 2$), each IFE-cell is further partitioned into two triangles, while in a three-dimensional space ($ND = 3$), each IFE-cell is further partitioned into five tetrahedra.

The basis functions of any mesh element can be obtained by scaling the basis functions of an element which vertices lie on the corners of a unit-size IFE–cell and both mesh elements are positioned at the same orientation [see illustration in figure 5.5].

The linear basis functions of a general tetrahedral element can be expressed as

![Figure 5.5: IFE–cell stretching.](image)
\[ \psi_i(\vec{x}) = a_1(x_1/h_1) + a_2(x_2/h_2) + a_3(x_3/h_3) + a_4 . \quad (5.22) \]

The gradient of the basis functions will be
\[ \nabla \psi_i(\vec{x}) = \left( \frac{a_1}{h_1}, \frac{a_2}{h_2}, \frac{a_3}{h_3} \right)^T . \quad (5.23) \]

### 5.6.2 IFE Mesh Stretching Rule

The IFE mesh stretching follows the potential gradients and local plasma conditions implied by the physics of the problem. The mesh is stretched in each coordinate independently. For example, the IFE mesh is stretched in the \( x \) axis according to the following stretching rule \[14\]
\[ \frac{x(\xi) - x_0}{L} = \frac{\beta + 1 - (\beta - 1) \left( \frac{\beta+1}{\beta-1} \right)^{1-\xi}}{\left( \frac{\beta+1}{\beta-1} \right)^{1-\xi} + 1} , \quad (5.24) \]

where \( x \) is the physical coordinate, \( \xi \) is a logical coordinate such that \( 0 \leq \xi \leq 1 \), and \( \beta \) is a stretching parameter such that \( \beta > 1 \). Note that
\[ x(0) = x_0, \]
\[ x(1) = x_0 + L . \]

Figure 5.6 shows the variation of the mesh stretching rule with stretching parameter. As the figure illustrates, the smaller the stretching parameter, the more grid points will be stacked towards \( x_0 \) and the less points will be stacked towards \( x_L \). The number of logical grid points is determined from the minimum physical grid size, \( \Delta x_{\text{min}} \), and the stretching parameter, \( \beta \), by solving equation (5.24) for \( \Delta \xi \) which can be expressed as
\[ \Delta \xi = 1 - \frac{\log a + b}{\log a} \quad (5.25) \]

where \( a = \frac{\beta+1}{\beta-1} \), \( b = \frac{\Delta x_{\text{min}}}{(\beta-1)L} \). Hence, the number of logical grid points, \( N_\xi = 1 + \text{floor}(\frac{1}{\Delta \xi}) \). Obviously, the number of physical grid points, \( N_x = N_\xi \). The stretching rule (5.24) can also be applied in a reverse direction to obtain a reverse stretching if needed.
5.7 HG–IFE–PIC Interpolation Procedure

Determining the location of each particle with respect to the PIC mesh, we can interpolate, or deposit, the particle quantities to the PIC mesh nodes. These quantities can, then, be interpolated to the IFE mesh through the IFE local basis functions. This requires us to have an IFE-PIC mesh connectivity array which determines the IFE mesh element in which each PIC mesh node occurs. After solving the field equation on the IFE mesh, the field quantities are interpolated back to the particle locations through the IFE local basis functions using a reversed interpolation procedure (see figure 5.7).

5.7.1 Particle-PIC Deposition

The physical quantities of simulation particles are interpolated onto the PIC mesh using a typical linear weighting functions of a standard PIC code.
Figure 5.7: Interpolation procedure for a hybrid-grid IFE-PIC. The PIC mesh is shown in light grey and the IFE mesh in dark grey.

5.7.2 PIC-IFE Mesh Interpolation

The physical quantities defined on the PIC mesh nodes are interpolated into the IFE mesh through linear interpolation. Note that the IFE mesh size should be no smaller than the PIC mesh size, otherwise the distribution of the deposited particles quantities, like charge density, will be lost among IFE-PIC interpolations and noisy field solution will be encountered.
5.7.3 IFE-PIC Mesh Interpolation

The physical quantities obtained on the IFE mesh are interpolated into the PIC mesh through the finite element basis functions constructed on the IFE mesh

\[ u_{i,j} = \sum_{k=1}^{\text{NEN}} u_k \psi_k(\vec{x}_{i,j}) \]  

(5.26)

where \( u_{i,j} \) is the field quantity \( u \) interpolated at the PIC mesh node \((i, j)\), \( \text{NEN} \) is the number of element nodes ( = 4 in tetrahedral elements), \( u_k \) is the field quantity evaluated at the IFE mesh element node \( k \) and \( \psi_k(\vec{x}_{i,j}) \) is the IFE local basis function \( k \) evaluated at the PIC mesh node location \( \vec{x}_{i,j} \).

5.8 Numerical Experiments

To assess the accuracy of the new HG-IFE-PIC model, we perform the following set of numerical experiments.

5.8.1 Single Particle Motion

In this numerical experiment, we use the HG-IFE-PIC model to simulate the orbital motion of a single charged particle in the presence of crossed electric and magnetic fields. The simulation domain is shown in figure 5.8.

The equation of motion of the charged particle is given by

\[ m \frac{dV}{dt} = q(E + V \times B) \]  

(5.27)

Here, we assume that \( E = (-E_0, 0, 0) \) and \( B = (0, 0, B_0) \) such that \( E_0 > 0 \) and \( B_0 > 0 \). The exact solution of the particle trajectory may be expressed as [12]

\[ x(t) = \frac{V_+}{\omega_c} \left[ \sin (\omega_c t + \delta) - \sin \delta \right] + x(0), \]  

(5.28)

\[ y(t) = \frac{V_+}{\omega_c} \left[ \cos (\omega_c t + \delta) - \cos \delta \right] + \frac{E_0}{B_0} t + y(0), \]  

(5.29)

\[ z(t) = w(0) t + z(0). \]  

(5.30)
where $V = (u, v, w)^T$, $\omega_c = \frac{qB_0}{m}$, $\delta = \cos^{-1}\left(\frac{u(0)}{V_\perp}\right)$, and $V_\perp = \sqrt{u^2 + \left(v - \frac{E_0}{B_0}\right)^2}$.

The HG-IFE-PIC is used to solve the particle motion using several values of the stretching parameter. Figure 5.9 shows the variation of the maximum trajectory error as induced by the numerical solution with the stretching parameter. The maximum trajectory error is defined here as the maximum absolute difference between the numerical and exact particle trajectories. It is obvious from the figure, that the stretching parameter has approximately no effect on the particle trajectory.

### 5.8.2 Plasma Flow Through Ion Optics

In this numerical experiment, we investigate the effect of the stretching parameter on the solution of plasma flow through ion optics. The simulation setup is as shown in figure 5.10. A streamline HG-IFE-PIC model is used here. The details of the model is left to a subsequent chapter. In brief, ion particles are injected from the upstream boundary surface with Bohm velocity. Ion trajectories, electrostatic field and space charge are all solved self-consistently. The upstream boundary is immersed in the
Figure 5.9: Effect of stretching parameter on the trajectory of a single charged particle.

discharge plasma and hence have a fixed potential equal to the plasma potential. All other boundaries have Neumann boundary conditions. Each optics grid is set to its corresponding potential. There is no exact solution for this problem to compare our numerical solution with. Therefore, we will compare the solution of the HG-IFE-PIC obtained on a stretched IFE mesh with a base solution in which the IFE mesh is uniform. The stretching is done here only in the downstream zone along the $z$ direction. The solution of each stretched mesh is compared to the base solution through

$$||\Delta \Phi||_2 = ||\Phi|_{\beta} - \Phi|_{\beta \to \infty}||_2$$

The quantity $||\Delta \Phi||_2$ is plot against $\beta$ in figure 5.11. From the plot, we notice that the error introduced by the mesh stretching vanishes as $\beta \to \infty$. Specifically, for $\beta \geq 1.2$, the error is nearly negligible.
Figure 5.10: Simulation domain of the ion optics plasma flow experiment.

Figure 5.11: Effect of stretching parameter on ion optics potential solution.
Chapter 6

Ion Optics Simulations

6.1 Introduction

In this chapter, we introduce the physical and mathematical modeling of ion thruster optics, and present the results of the simulations performed on the NEXT ion optics. All physical processes that are known to have profound effects on the operational performance and lifetime of ion thrusters are discussed. Normalization of the mathematical model is then performed. A simulation algorithm for the simulation of ion optics plasma flow and processes is developed. Two simulation models are introduced: a standard HG-IFE-PIC ion optics model and a streamline HG-IFE-PIC ion optics model. We also discuss in this chapter all the numerical aspects associated with our simulation models.

Both standard and streamline HG-IFE-PIC ion optics models are used to perform ion optics simulation on the NEXT ion optics. We perform ion beamlet simulations in a two-quarter aperture domain as well as a whole gridlet domain. The discharge plasma conditions are changed to span the whole operation envelope of the grid system from cross-over through perveance limits. Impingement current limits as well as electron backstreaming limits are assessed through numerical simulation.

All simulation runs in this chapter were performed on Idesk, a Dell® Workstation. Idesk has dual Intel® Xeon™ processors each of 3.0 GHz and has 2 GB of total installed memory.
6.2 Physical and Mathematical Modeling of Ion Optics

6.2.1 Upstream Discharge Chamber Plasma

In a typical ion thruster, a propellant (usually a heavy-atom inert gas, such as xenon) is injected into a discharge chamber. The propellant is ionized by electron impact. Propellant neutrals and ions leave the discharge chamber with a temperature close to the discharge chamber wall temperature, typically about 300 K. The electron temperature is much higher than ion and neutral temperature. A value of few electron volts is usually assumed for the upstream plasma electron temperature. In this study, we will assume a fixed value of 5 eV for the upstream plasma electron temperature.

The upstream plasma density can be related to the beamlet current and upstream plasma electron temperature through the following relation.

\[ I_b = e n_{i,sh} v_{Bohm} A_{sh} \]

\[ n_{i,sh} = n_0 \exp \left( -\frac{e \Delta V}{kT_{e0}} \right) \]

where \( I_b \) is the beamlet current, \( n_{i,sh} \) is the ion density in the plasma sheath, \( A_{sh} \) is the sheath area, \( \Delta V \) is the potential drop in the presheath. Standard Langmuir probe theory assume a potential drop of \( eT_{e0}/2k \) in the presheath and the Boltzmann relation to obtain \( 0.6 = \exp(-1/2) \) multiplication constant in the above expression for the ion saturation current. It is now clear that this multiplication constant may be a variable depending on the plasma conditions [52].

Since electrons are much more mobile than ions, they adjust their locations much faster than ions due to field changes. Therefore, electron density may be considered to follow a Boltzmann’s distribution in the regions where electrons are most likely to exist, i.e. the upstream and downstream regions of the ion optics. Specifically, in the upstream region, the electron density is given by

\[ n_e = n_0 \exp \left( \frac{e(\Phi - \Phi_0)}{kT_{e0}} \right) \],

where \( \Phi_0, n_0, \) and \( T_{e0} \) are, respectively, the potential, density and electron temperature of the discharge plasma.
6.2.2 Ion Optics Beam Extraction

Propellant ions are extracted from the discharge plasma by the electrostatic field of the ion optics to form a very high velocity ion beam. The ion optics consists of two, or more, electrically conducting grids with thousands of matching apertures. The apertures are typically arranged in hexagonal layout. The grids are electrically and physically isolated from each other by a dielectric spacing ring. Electric potentials are applied to the grids such that the upstream grid, which is called the screen grid, is set at a potential, $\Phi_s$, slightly above the discharge plasma potential, $\Phi_0$, to screen out plasma electrons. On the other hand, the downstream-next grid, which is called the accel grid, is set at a much lower potential, $\Phi_a$, to provide the accelerating field.

6.2.3 Downstream Neutralization Plasma

At far downstream of the ion optics, electrons are emitted through a neutralizer cathode into the ion beam at the same rate as ions ejection to prevent the ion thruster system from charging. Electron temperature in the downstream neutralization plasma is in the same order of magnitude as the electron temperature in the upstream discharge plasma, but the exact value is not a well-determined quantity. Recent wear tests for the NEXT ion engine reported electron temperature in the downstream plasma of about 1.2 eV [37]. It was also shown that reducing the neutralizer flow rate increases the electron temperature up to 4.0 eV. In this study, we will assume a fixed downstream electron temperature of 1.5 eV for all simulations.

Similar to the upstream plasma, the downstream electron density is given by

$$n_e = n_\infty \exp \left( \frac{e(\Phi - \Phi_\infty)}{kT_{e\infty}} \right), \quad (6.4)$$

where $\Phi_\infty$, $n_\infty$, and $T_{e\infty}$ are, respectively, the potential, density and electron temperature of the downstream neutralized plasma.

6.2.4 Dynamics of Beam Ions

In ion optics, as the mean free path of particle-particle collisions is very larger than the dimensions of the ion optics aperture, one may consider the flow of plasma through the ion optics as collision-less. Neglecting collisional effects, the ions dynamics can be completely described by knowing their kinematics (velocities and
positions) and the local electrostatic force acting on them. The trajectory of each ion is determined from the integration of its equation of motion given by Newton’s second law as

\[
\frac{d}{dt}(mv) = F = qE, \quad (6.5)
\]

\[
v = \frac{dx}{dt}. \quad (6.6)
\]

The knowledge of ion trajectories is, in turn, required to compute the space charge density which drives the solution of the electrostatic field.

### 6.2.5 Electrostatic Field

The electrostatic field is determined from the solution of the Poisson’s equation

\[
\nabla \cdot D = \nabla \cdot (\epsilon_0 E) = \rho, \quad (6.7)
\]

where

\[
E = -\nabla \Phi. \quad (6.8)
\]

The overall space charge density equals the ions charge density minus the electrons charge density, i.e.

\[
\rho = \rho_i - \rho_e,
\]

where \( \rho_i = qn_i \) and \( \rho_e = en_e \).

### 6.2.6 Beam Current Extraction

The ability of an ion optics grid system to extract the maximum ion current per hole for the minimum total accelerating voltage is a measure of its performance. The nomenclature and dimensions of a typical grid system are illustrated in figure 6.1. This maximum extracted ion beam current is referred to as perveance and is defined by 2

\[
\frac{I_{k,\text{max}}}{V_T^{3/2}} = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q_i (d_s)}{m_i (T_e)}}, \quad (6.9)
\]
where $I_b$ is the beamlet ion current assumed to be emitted from a plasma sheath whose area is constant, and approximately equal to that of the screen aperture, $l_e$ is the effective acceleration length which is either given by

$$l_e = \sqrt{l_g^2 + \frac{d_s^2}{4}}, \quad (6.10)$$

or by [29]

$$l_e = \sqrt{(l_g + t_s)^2 + \frac{d_s^2}{4}}. \quad (6.11)$$

Beamlet current variations may be expressed in terms of the maximum normalized perveance per hole as [3]

$$\frac{I_{b,max}}{V_{T}^{3/2}} \left( \frac{l_e}{d_s} \right)^2 = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q_i}{m_i}}. \quad (6.12)$$
The actual normalized perveance per hole may be expressed as \[ NP_H = \frac{I_b}{V_T^{3/2}} \left( \frac{l_e}{d_s} \right)^2, \] \hspace{1cm} (6.13)

The $NP_H$ in the above definition has the dimensions of $A/V^{3/2}$. The normalized perveance per unit grid area may be also used for the evaluation of the beamlet current variations, which is given by \[ P_T = \frac{I_b}{A_{TG} V_T^{3/2} \epsilon_0} \frac{l_e^2}{9 \sqrt{2 q_i m_i}}, \] \hspace{1cm} (6.14)

where $A_{TG}$ is the aperture and web area, and $l_e$ is given by (6.11).

In this study, we will follow the standard Langmuir probe theory to estimate the upstream plasma ion density from beamlet current to setup our simulation using equation (6.1). We assume a hemispherical screen aperture sheath surface in order to calculate the sheath area. Of course, the sheath surface is not always hemispherical and depends on the upstream plasma condition. The correct and self-consistent relation between the beamlet current and upstream plasma is obtained from the simulation as a function of the upstream plasma condition.

### 6.2.7 Impingement Current Limits

Typical accel grid impingement current data display a U shape when plotted as a ratio of impingement-to-beamlet current versus beamlet current \[39\]. At low beamlet currents, the relative impingement current rises due to cross-over ion impingement on the downstream edge of the accel hole barrels. The reason is that the sheath that separates the chamber discharge plasma from the ion acceleration region becomes too dished upstream to the point where ions are over-focused and their trajectories cross. At moderate beamlet currents, the relative impingement current is flat and at a value dependent upon the background neutral density and the propellant utilization efficiency of the ion source. As the beamlet current is increased to higher values, the relative impingement current will again rise quickly indicating that direct ion interception is occurring on the upstream edge of the accel hole barrels due to perveance (or space-charge) limitations. This is due to that the upstream plasma sheath becomes much less dished, and the beamlet becomes under-focused. The different operating modes of an ion beamlet is shown in figure 6.2.
6.2.8 Electron Backstreaming

The ability of an ion optics system to impart a negative potential throughout the beamlet volume near the axial location of the accel grid determines its capacity to stop beam plasma electrons from backstreaming into the discharge chamber. The geometry of a typical ion optics aperture set applies boundary conditions that result in an electrostatic potential saddle point being formed near the axial location of the accel grid on the beamlet centerline. The saddle point presents the lowest resistance path to electrons on trajectories that could carry them from the beam plasma toward
the discharge plasma [92].

The backstreaming limit is the magnitude of negative voltage that must be applied to the accel grid to prevent beam plasma electrons from backstreaming. Ideally the accel grid voltage should be held negative of but as close to this limit as possible. This will ensure that damage due to the small current of charge exchange ions that sputter erode and limit the lifetime of this grid will be minimized. Unfortunately, the backstreaming limit can change as the accel grid wears over time, and compromises on selecting the magnitude of the accel voltage must be made. Many factors can affect the backstreaming voltage including aperture geometry and beamlet current. The plasma flow field environment in the ion beam is also an important factor in determining the backstreaming limit [92]. The onset of backstreaming can also be strongly affected by the operational conditions associated with the neutralizer and conductive plasma-bridge that forms between the neutralizer plasma and the beam plasma [21].

### 6.3 Normalization

The governing equations of a plasma simulation are typically normalized to avoid dealing with too large or too small numbers which are the main source of many undesirable round-off errors. To normalize the equations of the ion optics model, we will assume the set of reference variables listed in Table 6.1.

<table>
<thead>
<tr>
<th>Variable name</th>
<th>Reference</th>
<th>Normalized</th>
</tr>
</thead>
<tbody>
<tr>
<td>mass</td>
<td>$m_{\text{ref}}$</td>
<td>$m'$</td>
</tr>
<tr>
<td>charge</td>
<td>$q_{\text{ref}}$</td>
<td>$q'$</td>
</tr>
<tr>
<td>length</td>
<td>$L_{\text{ref}}$</td>
<td>$L'$</td>
</tr>
<tr>
<td>time</td>
<td>$t_{\text{ref}}$</td>
<td>$t'$</td>
</tr>
<tr>
<td>velocity</td>
<td>$v_{\text{ref}}$</td>
<td>$v'$</td>
</tr>
<tr>
<td>potential</td>
<td>$\Phi_{\text{ref}}$</td>
<td>$\Phi'$</td>
</tr>
<tr>
<td>number density</td>
<td>$n_{\text{ref}}$</td>
<td>$n'$</td>
</tr>
<tr>
<td>electric permittivity</td>
<td>$\epsilon_{\text{ref}}$</td>
<td>$\epsilon'$</td>
</tr>
<tr>
<td>temperature</td>
<td>$T_{\text{ref}}$</td>
<td>$T'$</td>
</tr>
</tbody>
</table>

Table 6.1: Reference and normalized Variables.
We choose
\[ L_{ref} = \sqrt{\frac{\epsilon_{ref}\Phi_{ref}}{n_{ref}q_{ref}}} }, \]
such that the normalized Poisson’s equation takes the form
\[ -\nabla' \cdot (\epsilon'\nabla'\Phi') = \sum_j n'_j q'_j. \]

Let
\[ t_{ref} = \frac{L_{ref}}{v_{ref}} \quad \text{and} \quad v_{ref} = \sqrt{\frac{q_{ref}\Phi_{ref}}{m_{ref}}}. \]

Then, the normalized Newton’s second law becomes
\[ m'\frac{dv'}{dt'} = -q'\nabla'\Phi'. \]

We also choose
\[ \Phi_{ref} = \frac{kT_{ref}}{q_{ref}}, \]
such that the normalized Boltzmann’s distribution becomes
\[ n'_e = n'_\infty \exp \left( \frac{e'(\Phi' - \Phi'_\infty)}{T_{e\infty}} \right). \]

In ion optics simulation, the reference variables are commonly chosen as
\[ q_{ref} = e, \quad m_{ref} = m_i, \quad T_{ref} = T_{e0}, \quad \epsilon_{ref} = \epsilon_0, \quad \text{and} \quad n_{ref} = n_0. \]

For which the reference length \( L_{ref} \) becomes the Debye length of the upstream discharge plasma \( \lambda_{D_0} \). For convenience, the ‘primes’ will be dropped from all normalized variables.

### 6.4 Simulation Model

#### 6.4.1 Simulation Domain

Our simulation domain extends, as in [89], from the upstream discharge plasma to the downstream neutralization plasma. We make no assumption about the upstream
plasma sheath surface. The ion extraction is determined self-consistently from the acceleration voltage drop across the optics and the upstream plasma boundary conditions. This puts a constraint on the mesh size in the upstream region that it should not exceed the Debye length of the upstream plasma, $\lambda_D$. 

The downstream boundary represents the neutralized propellant plasma, which has a plasma potential $\Phi_\infty$, density $n_\infty$ and electron temperature $T_{e\infty}$. We define a quasi-neutral zone near the downstream boundary within which we assume that the propellant ions are neutralized. We take the downstream plasma density to be the average ion density within the quasi-neutral zone. Hence, the downstream plasma density is updated each time step.

### 6.4.2 Simulation Algorithm

The trajectories of beamlet ions are very slightly affected by other ion species, such as CEX propellant ions and CEX non-propellant ions. Momentum transfer collisional effects are even much more insignificant because of the very large mean-free-path length of the beamlet plasma as compared with the ion optics dimensions. Thus, we can perform the simulation of ion optics plasma flow three separate phases. In the first phase, we simulate the flow of beamlet ions and electrons as a collision-less plasma. Ions are treated as particles and electrons are assumed to follow a Boltzman’s distribution. In the second phase, we simulate the flow of the propellant neutral atoms as a collision-less gaseous flow.

In the simulation of beamlet ions, we will use the PIC algorithm described in full details in a previous chapter. The details of the PIC algorithm will not be repeated here. The beamlet ions are represented by macro-particles. The amount of the ion current extracted by the ion optics is determined by the upstream plasma conditions.

### Boundary Conditions

Since the upstream boundary surface is set to be immersed in the upstream plasma, we apply Dirichlet potential boundary condition on the upstream surface, i.e.

$$\Phi_{upstream} = \Phi_0.$$  

All side surfaces are assumed either to have a symmetric potential boundary condition or an open potential boundary condition according to the simulation setup. Hence,
Neumann potential BC is assumed on all side surfaces, i.e.
\[
\left( \frac{\partial \Phi}{\partial n} \right)_{\text{side}} = 0.
\]

The downstream boundary surface is set to be immersed in the downstream plasma. However, we do not know a priori the necessary extension of the downstream region, so we assume a Neumann potential BC on the downstream boundary surface such that the potential on the downstream surface is determined self-consistently, i.e.
\[
\left( \frac{\partial \Phi}{\partial n} \right)_{\text{downstream}} = 0.
\]

We expect then that the potential in the downstream region will asymptote to the downstream plasma potential.

The setup of the simulation domain also affects the particle BC. In the ion beamlet simulation, we allow absorption boundary condition on the upstream and downstream boundary surfaces. Absorption boundary conditions are also assumed on open boundary surfaces as well as solid grid surfaces. On all symmetric surfaces, we assume reflection boundary condition. The same particle boundary conditions are applied to the simulation of charge-exchange ions.

In the neutral efflux simulation, we use the same setting of boundary conditions except on solid surfaces where neutral particles are assumed to reflect diffusively with a half-range equilibrium Maxwellian distribution for the particles that are directed away from the surface. Complete thermal accommodation is assumed, which means that the temperature of the reflected particles is equal to the temperature of the reflecting wall.

**Domain and Mesh Size**

To perform the ion beamlet simulations properly, the upstream boundary needs to be set far enough from the screen grid and the PIC mesh size should be no larger than the Debye length based on the upstream plasma conditions, \( \lambda_{D0} \). Since \( \lambda_{D0} \propto 1/\sqrt{n_0} \), the PIC mesh needs to be reduced as the discharge plasma density increases. Instead of using too many PIC meshes, we have selected a set of PIC meshes. Each of them will be suited for a range of upstream plasma densities.

Using the HG-IFE-PIC mode, we can separate the PIC mesh used for sampling particle charge from the IFE mesh used for sampling and solving the electric field.
Therefore, the size of the IFE mesh cells can be varied according to the potential gradients and local plasma conditions. In our simulation, we keep the IFE mesh cell size equal to the PIC mesh cell size in the upstream zone where upstream plasma conditions are prevailing and potential gradients are very large in the plasma sheath region. In the downstream zone, where downstream plasma is prevailing and potential gradients are small, we stretch the IFE mesh in the downstream direction keeping the maximum mesh cell size smaller than the Debye length of the downstream plasma, $\lambda_{D\infty}$.

**Particle Injection**

In order to have a stable upstream plasma sheath, beam ions are injected into the domain from the upstream boundary at the Bohm velocity, which is given by

$$v_{\text{Bohm}} = \sqrt{\frac{kT_{e0}}{m_i}}.$$  \hspace{1cm} (6.15)

where $m_i$ is the ion mass $\approx 131.3$ AMU for Xe$^+$. A random velocity component is added to the velocity of the injected ions to account for the Maxwellian velocity distribution they have as they leave the discharge chamber. The thermal velocity of the ions is calculated based on the wall temperature of the discharge chamber, $T_w \approx 300$ K.

**Time Step**

The simulation time step is calculated based on a Courant condition to assure stable calculations and minimize trajectory errors as described the chapter of the PIC model. Since the velocity of the beamlet ions increase as they move along the electrostatic field lines, the constant simulation time step should be calculated based on the maximum velocity of beamlet ions. The maximum velocity of beamlet ions can be simply estimated from the conservation of energy of ions as

$$v_{\text{max}} = \sqrt{2(\Phi_0 - \Phi_a) + v_{\text{Bohm}}^2}.$$  

For a typical ion thruster condition, $v_{\text{max}} \approx 28v_{\text{Bohm}}$ which means that the ions initially injected at the Bohm speed will be pushed at a much smaller time step than what is imposed by the Courant condition. This results in a very long wasted computational time to fill the upstream zone by beamlet ions.
One solution is to initialize our simulation with ions injected all over the upstream zone. Of course, these ions will not have the correct velocity but they will eventually adjust their velocities according to the local electric field.

Alternatively, we can apply an adaptive time stepping. In which, we can push our simulation particles at the maximum allowed time step based on Courant condition to accelerate the simulation. As the ions approach the ion optics region, the time step will be automatically reduced to account for the increase in ion velocity. The number of injected particles and the length of the injection strip length should be adapted to account for the change in the time step.

In our simulation, we applied both techniques to achieve the best computational performance of our PIC code.

**Current Sampling**

The current passing through a virtual surface or collected by a solid surface is sampled in the PIC simulation by summing up all currents passing through the corresponding surface, which is calculated from

\[
I_k = \frac{q_p N_p}{\Delta t}
\]  

(6.16)

where \( I_k \) is the current collected by the surface \( k \), \( q_p \) is the particle charge, \( N_p \) is the number of collected simulation particles within time step \( \Delta t \). Currents passing through virtual surfaces are calculated similarly.

**6.4.3 Streamline PIC Simulation**

Since the bulk velocity of beamlet ions, \( v_{i_b} \), is much larger than the thermal velocity, \( v_{i_{th}} \), typically, \( v_{i_{th}} \approx 0.003 - 0.08v_{i_b} \) in ion thrusters, we may neglect the ion thermal velocity with respect to the ion bulk velocity. This is physically equivalent to assuming a cold beamlet plasma. This assumption allows us to use a streamline PIC model which dramatically reduces the computational time and memory requirements.

In a streamline PIC model, we assume that the particles injected from the upstream domain at any time step will exactly follow the pathes of the particles injected in previous time steps. Of course, this is only valid for a steady-state flow of particles with no random velocity involved which is the case here. This means that there is
no need to continuously inject particles at each time steps. Instead, we can initially inject a set of particles and keep accumulating the charge they deposit to the mesh nodes as they migrate in the simulation domain. At any subsequent time step, the accumulated charge density in the domain will be exactly equivalent to the charge density of trains of particles resulting from a continuous particle injection.

The advantage of using the streamline PIC instead of a standard PIC is that we can attain the same degree of charge density smoothness with a much fewer number of traced simulation particles. For example, to trace 10,000 particles in a 100–cell long domain with the streamline PIC is equivalent to tracing 100,000,000 particles injected every 0.01 time step with a standard PIC, if we assume all particles are all moving at a speed of unity.

Also, since the number of simulated particles is very small compared to a standard PIC, the particle pushing step becomes extremely fast. So, we can push the particles through a fixed electric field and accumulate the domain charge density and do not update the field until all the particles leave the domain. The electric field is then updated and we start a new simulation loop by flushing, or erasing the mesh charge density and injecting a new set of particles. This process is repeated until steady-state is achieved. Convergence can be claimed by satisfying the following criteria

\[
||\Delta \Phi^n||_\infty = \max_{i=1}^N |\Phi^n_i - \Phi^{n-1}_i| < \epsilon ||\Delta \Phi^1||_\infty,
\]

\[
||\Delta \rho^n||_\infty = \max_{i=1}^N |\rho^n_i - \rho^{n-1}_i| < \epsilon ||\Delta \rho^1||_\infty,
\]

\[
||\Delta \Phi^n||_2 = \frac{1}{N} \sqrt{\sum_{i=1}^N (\Phi^n_i - \Phi^{n-1}_i)^2} < \epsilon ||\Delta \Phi^1||_2,
\]

\[
||\Delta \rho^n||_2 = \frac{1}{N} \sqrt{\sum_{i=1}^N (\rho^n_i - \rho^{n-1}_i)^2} < \epsilon ||\Delta \rho^1||_2.
\]

Typically, \(\epsilon\) is chosen as \(10^{-3}\). The streamline PIC algorithm is summarized below

**Streamline PIC Algorithm**

1. Solve field equation in the simulation domain.

2. Apply field boundary conditions.
3. Calculate E field.
4. Flush mesh charge density.
5. Inject a *sheet* of simulation particles.
6. Deposit particles charge to mesh nodes.
7. Update particle velocities.
8. Move particles.
9. Apply particle boundary and surface conditions.
10. If particles exist in simulation domain, go to 6.
11. Apply charge boundary conditions.
12. If steady-state achieved then exit, unless go to 1.

## 6.5 NEXT Ion Optics

The ion optics of the NASA’s Evolutionary Xenon Thruster (NEXT) system [55] is studied using the ion optics models developed above in this chapter. The NEXT has a 40 cm beam extraction diameter giving more than twice the area of the 30 cm NSTAR thruster. The NEXT thruster also has an accelerator grid 50% thicker than the NSTAR thruster to afford a longer service life. Since NEXT is still under development, we select in this study one of the proposed geometric setups for NEXT. The nominal dimensions of the NEXT ion optics under investigation is listed in Table 6.2. The nominal throttling condition of the NEXT ion optics is listed in Table 6.3. The net accelerating voltage, $V_n$, is defined here as the difference between the upstream plasma potential and the downstream plasma potential, i.e.

$$V_n = \Phi_0 - \Phi_\infty$$
screen hole diameter, $d_s$ 0.0595 in
screen grid thickness, $t_s$ 0.015 in
acceleration hole diameter, $d_a$ 0.043 in
acceleration grid thickness, $t_a$ 0.03 in
screen to acceleration grid gap, $l_g$ 0.026 in
center-to-center hole spacing, $l_{cc}$ 0.0874 in

Table 6.2: Nominal dimensions the ion optics.

<table>
<thead>
<tr>
<th></th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>net accelerating voltage, $V_n$</td>
<td>1800 V</td>
</tr>
<tr>
<td>screen grid voltage, $V_s$</td>
<td>1780 V</td>
</tr>
<tr>
<td>acceleration grid voltage, $V_a$</td>
<td>$-210$ V</td>
</tr>
</tbody>
</table>

Table 6.3: Nominal throttling condition of the ion optics.

### 6.6 Simulation Domain Layout

In the following simulations, we will consider two simulation domain layouts. The first layout includes two-quarter apertures, as in \[89\]. This layout implicitly assumes full geometric similarity among all ion optics apertures such that the simulation of the two-quarter aperture domain fully accounts for the hexagonal layout of the ion optics apertures as shown in figure 6.3. Due to the similarity assumption, symmetric boundary conditions can be applied on all side surfaces. The PIC models which use this layout are successful to predict erosion patterns with a reasonable accuracy \[89\]. We will refer to this layout as the two-quarter apertures layout and to the model incorporating this layout as the two-quarter aperture model or the single aperture model since all grid apertures in such model are assumed identical.

The two-quarter apertures layout is based upon a well-justified assumption of aperture similarity when simulating a full scale ion optics with thousands of surrounded apertures and a very small fraction of un-surrounded edge apertures. However, this assumption may not be justified as well if we are interested in the simulation of a sub-scale ion optics with much fewer apertures where the contribution of edge apertures is more profound. Experimental studies have shown that the performance of an ion optics depends on the fraction of surrounded apertures to total number of apertures \[39\]. The similarity assumption also lack justification if we are focusing on the behavior of edge apertures in a full scale ion optics.

Therefore, we introduce a second simulation layout which includes one quarter of a
sub-scale ion optics. In this layout, the geometric differences between surrounded and edge apertures are fully considered as shown in figure 6.4. Only overall geometric symmetry of the whole sub-scale gridlet is assumed which is clearly evident.

6.7 Standard HG-IFE-PIC Simulation

In this section, use the the standard PIC model to analyze the NEXT ion optics at the given throttling condition. The results of this model will be used to validate the
streamline PIC model, also described in the same chapter.

### 6.7.1 Simulation Setup

Here, we consider an upstream discharge plasma electron temperature $T_{e0} = 5 \text{ eV}$, a *nominal* upstream plasma density of $1.0 \times 10^{17} \text{ m}^{-3}$, a downstream plasma potential $\Phi_\infty = 0$ and a downstream plasma electron temperature $T_{e\infty} = 1.5 \text{ eV}$. For the given upstream electron temperature and nominal plasma density, the Debye length $\lambda_{D0} \approx 5.26 \times 10^{-5} \text{ m}$. The ions temperature in the discharge plasma is assumed to be equal to the discharge chamber wall temperature, so $T_i = T_w \approx 300 \text{ K}$. We will also consider two extreme operating conditions one at $0.05 \times 10^{17} \text{ m}^{-3}$ which represents cross-over condition and the other at $14.0 \times 10^{17} \text{ m}^{-3}$ which represents a perveance.
condition.

Our simulation domain includes two-quarter apertures. This domain retains the hexagonal layout nature of the ion optics grid and allows the application of symmetric boundary conditions on all side faces.

Since \( \lambda_{D0} \propto 1/\sqrt{n_0} \), a suitable PIC mesh needs to be used for each discharge plasma density. We setup each uniform Cartesian PIC mesh such that \( \Delta z = \lambda_{D0} \). In the lateral directions, we use \( \Delta x = \Delta z \) and \( \Delta y = \Delta z \sqrt{3} \). The Cartesian-based tetrahedral stretched IFE mesh is setup such that \( (h_z)_{\text{min}} = \lambda_{D0} \) and \( (h_z)_{\text{max}} < \lambda_{D\infty} \), where \( h_z \) is the length of the IFE base Cartesian cells in the \( z \)-direction. The stretching parameter used in all cases is 1.2. In the lateral directions, we use a uniform mesh size. Table 6.4 summarizes the mesh parameters for all runs. The PIC and IFE meshes used for the simulation run of \( n_0 = 1.0 \times 10^{17} \text{m}^{-3} \) is shown in figure 6.5.

<table>
<thead>
<tr>
<th>Plasma density ( n_0[\text{m}^{-3}] )</th>
<th>PIC Mesh</th>
<th>IFE Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \Delta z [\text{m}] )</td>
<td>( n_x \times n_y \times n_z )</td>
</tr>
<tr>
<td>( 0.05 \times 10^{17} )</td>
<td>( 5.2566 \times 10^{-5} )</td>
<td>( 22 \times 22 \times 271 )</td>
</tr>
<tr>
<td>( 1.0 \times 10^{17} )</td>
<td>( 5.2566 \times 10^{-5} )</td>
<td>( 22 \times 22 \times 251 )</td>
</tr>
<tr>
<td>( 14.0 \times 10^{17} )</td>
<td>( 1.31415 \times 10^{-5} )</td>
<td>( 43 \times 43 \times 401 )</td>
</tr>
</tbody>
</table>

Table 6.4: PIC and IFE meshes used in the standard HG-IFE-PIC simulation.

### 6.7.2 Computational Performance

The computational time required to reach steady-state beam ion trajectories using the standard PIC model depends on the upstream plasma condition. It varies from less than 3 hours, for densities much less than the cross-over limit, to more than 60 hours for densities much larger than the perveance limit. The number of simulated particles was around 5 million particles in all runs. It is worth noting here the advantage of using the HG–IFE–PIC to speed up the three dimensional PIC ion optics simulation such that the whole operation envelope can be spanned in about two weeks of cheap computational time using available PCs or workstations.

The results of the ion optics simulation of NEXT ion optics using the standard HG-IFE-PIC model are presented and discussed in the following sections.
6.7.3 Beamlet Behavior

The beamlet ion density contours and selected potential contour lines on the \( y = 0 \) plane, which represents the horizontal plane cutting through the middle of a typical aperture, are shown in figure 6.6 for three selected operating conditions; namely \( n_0 = 0.05 \times 10^{17} \text{m}^{-3}, 1.0 \times 10^{17} \text{m}^{-3}, \) and \( 1.4 \times 10^{17} \text{m}^{-3} \). These conditions represent cross-over, nominal, and perveance conditions respectively. The simulation data was mirrored about the \( y-z \) plane to illustrate the whole beamlet flow through a single aperture. In the plots, the potential contour lines are shown for values from \(-210 \text{ V}\) to \( 1790 \text{ V} \) with a step of \( 200 \text{ V} \). The potential contour lines at \( \Phi = 0, 1780, \) and \( 1795 \text{ V} \) are also shown. The beamlet ion density contours in the plots are normalized by the nominal discharge plasma density.

The beamlet potential contours on the \( y = 0 \) plane are shown in figure 6.7 for the same operating conditions. These conditions represent cross-over, nominal, perveance conditions respectively. Potentials are normalized by the upstream plasma electron temperature, \( T_{e0} = 5 \text{ eV} \).

From the plots, we notice that the ion density decreases as the ions approach the screen aperture due to electrostatic ion expansion. The electrostatic field then focuses the ion beamlet as it passes through the screen aperture to a density larger than the upstream density. The focus point of the beamlet may be defined as the point along
(a) Cross-over, \( n_0 = 0.05 \times 10^{17} \text{m}^{-3} \)

(b) Normal, \( n_0 = 1.0 \times 10^{17} \text{m}^{-3} \)

(c) Perveance, \( n_0 = 14.0 \times 10^{17} \text{m}^{-3} \)

Figure 6.6: Beamlet plasma potential and ion density. Ion density is normalized by \( n_0 = 1.0 \times 10^{17} \text{m}^{-3} \). Potential contour lines are shown for the values from \(-210 \text{ V}\) to \(1790 \text{ V}\) with a step of \(200 \text{ V}\). The \(1780 \text{ V}, 1795 \text{ V}\) and zero potential contour lines are also shown.

beamlet axis at which the beamlet has minimum cross-section. We notice that the beamlet focus point moves in the upstream direction, i.e. the beamlet becomes more focused, as the upstream plasma density is decreased and vice versa.

Also from these plots, we can easily distinguish the upstream plasma sheath, which may be approximated by the iso-potential surface with \( \Phi = \Phi_s \). We notice that the shape and size of the sheath changes drastically with the variation in the upstream density. The accurate estimation of the size and shape of the upstream plasma sheath is known to be of crucial importance to ion optics studies, since it determines the ion extraction performance of an ion optics grid set. For example, at the nominal
Raed I. Kafafy  Chapter 6. Ion Optics Simulations

(a) Cross-over, $n_0 = 0.05 \times 10^{17}\text{m}^{-3}$

(b) Normal, $n_0 = 1.0 \times 10^{17}\text{m}^{-3}$

(c) Perveance, $n_0 = 14.0 \times 10^{17}\text{m}^{-3}$

Figure 6.7: Beamlet plasma potential. Potential contour lines are shown for the values from $-210\text{ V}$ to $1790\text{ V}$ with a step of $200\text{ V}$. The $1780\text{ V}$, $1795\text{ V}$ and zero potential contour lines are also shown. Potential is normalized by $T_e^0 = 5\text{ eV}$.

upstream density, the upstream plasma sheath is attached to the screen grid, and moderately dished with a convex and nearly spherical upstream side. At the lowest upstream density, the upstream sheath detaches from the screen grid, on the other limit, at the highest upstream density, the upstream sheath is being pushed by the dense upstream plasma towards the screen aperture to the extent it penetrates through the aperture opening.
6.7.4 Impingement Current

By varying the upstream plasma density from cross-over through perveance conditions, we can scan the operating envelope of an ion optics grid set. For each density, the accel grid impingement current, $I_i$, is calculated as well as the beamlet current, $I_b$, itself.

The *instantaneous* current sampled in a standard PIC simulation suffers from a high-frequency noise. This is mainly due to the short sampling time step which is imposed by the Courant condition. Conclusions made based on this noisy readings are misleading. Here, we apply *low-pass* noise reduction filter to reduce the noise accompanying the sampled current signal. We choose to apply a simple *integrator* filter which is both simple and efficient. Such filter is applied through

\[
    \bar{I}_j = \frac{1}{\tau} \int_{t_j - \tau/2}^{t_j + \tau/2} I \, dt \\
    \approx \frac{1}{\tau} \sum_{i = j - N_I/2}^{j + N_I/2} I_i \Delta t_i
\]

(6.17)

where $\bar{I}_j$ is the filtered current at sample time $t = t_j$, $\tau$ is the time band-width of the filter, $N_I$ is the number of sampling within the filter band-width. It is obvious that the wider the width of the filter, the higher the noise reduction is, but also the more sampling information lost. In this work, we assume a band-width of 50 $\Delta t$.

The beamlet and impingement currents predicted by the standard HG-IFE-PIC simulation for the studied plasma conditions, i.e. $n_0 = 0.05 \times 10^{17}$ m$^{-3}$, $1.0 \times 10^{17}$ m$^{-3}$, and $14.0 \times 10^{17}$ m$^{-3}$ are listed in Table 6.5.

<table>
<thead>
<tr>
<th>Plasma density [m$^{-3}$]</th>
<th>$I_b$ [mA]</th>
<th>$I_i/I_b$ [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.05 \times 10^{17}$</td>
<td>0.0045</td>
<td>4.63</td>
</tr>
<tr>
<td>$1.0 \times 10^{17}$</td>
<td>0.0678</td>
<td>0.0</td>
</tr>
<tr>
<td>$14.0 \times 10^{17}$</td>
<td>0.5017</td>
<td>20.75</td>
</tr>
</tbody>
</table>

Table 6.5: Beamlet and impingement currents at typical plasma conditions as estimated by the standard HG-IFE-PIC model.
6.8 Streamline HG-IFE-PIC Simulation: Two-Quarter Aperture

6.8.1 Simulation Setup

Here, we consider the same problem setup as in the standard HG-IFE-PIC simulation. To explore the whole operation envelope of the NEXT ion optics, we vary the upstream plasma density from $0.05 \times 10^{17} \text{m}^{-3}$ to $14.0 \times 10^{17} \text{m}^{-3}$. This covers the whole operation envelope from cross-over to perveance.

Our simulation domain includes two-quarter apertures. This domain retains the hexagonal layout nature of the ion optics grid and allows the application of symmetric boundary conditions on all side faces. The uniform Cartesian PIC mesh and the Cartesian-based tetrahedral stretched IFE mesh are setup in the same manner as we did in the standard HG-IFE-PIC simulation. Instead of using too many PIC meshes, we have selected a set of PIC meshes. Each of them can be used to resolve the discharge plasma sheath of a range of discharge plasma densities. Table 6.6 summarizes the mesh parameters for all runs.

<table>
<thead>
<tr>
<th>Plasma density $\times 10^{17} \text{m}^{-3}$</th>
<th>PIC Mesh</th>
<th>IFE Mesh</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.05 \leq n_0 \leq 0.2$</td>
<td>$5.2566 \times 10^{-5}$</td>
<td>$22 \times 22 \times 271$</td>
</tr>
<tr>
<td>$0.5 \leq n_0 \leq 1.0$</td>
<td>$5.2566 \times 10^{-5}$</td>
<td>$22 \times 22 \times 251$</td>
</tr>
<tr>
<td>$2.0 \leq n_0 \leq 4.0$</td>
<td>$2.6283 \times 10^{-5}$</td>
<td>$43 \times 43 \times 341$</td>
</tr>
<tr>
<td>$6.0 \leq n_0 \leq 10.0$</td>
<td>$1.7522 \times 10^{-5}$</td>
<td>$43 \times 43 \times 331$</td>
</tr>
<tr>
<td>$12.0 \leq n_0 \leq 14.0$</td>
<td>$1.3142 \times 10^{-5}$</td>
<td>$43 \times 43 \times 401$</td>
</tr>
</tbody>
</table>

Table 6.6: PIC and IFE meshes used in the streamline HG-IFE-PIC two-quarter aperture simulation.

6.8.2 Computational Performance

The computational time required to reach steady-state ion trajectories using the standard PIC model depends on the upstream plasma condition. It varies from less than 40 minutes, for densities much less than the cross-over limit, to more than 3 hours for densities much larger than the perveance limit. The number of simulated particles injected at the beginning of each simulation loop was 44,100 particles. It
is worth noting here the advantage of using the streamline HG–IFE–PIC to speed up the standard HG-IFE-PIC ion optics simulation such that the time to span the whole operation envelope can be reduced to less than a single day of cheap CPU time.

The steady-state beamlet profile is claimed when the $L^\infty$ and $L^2$ norms of the difference in the potential and ion charge density of two successive simulation loops are reduced by three orders of magnitude from the initial values. The convergence history of the streamline HG-IFE-PIC code at $n_0 = 0.05 \times 10^{17} \text{m}^{-3}$, for example, is shown in figure 6.8. From the figure, we notice that the simulation reaches steady state after about 20 simulation loops.

![Figure 6.8: Convergence history of the streamline HG-IFE-PIC simulation at $n_0 = 0.05 \times 10^{17} \text{m}^{-3}$.](image)

6.8.3 Beamlet Behavior

The simulation data obtained here are mirrored as above. Figure 6.9 shows the ion density and potential for typical plasma densities. In the plots, the potential contour lines are shown for values from $-210 \text{ V}$ to $1790 \text{ V}$ with a step of $200 \text{ V}$. The potential contour line of $\Phi = 0$, $1780 \text{ V}$, and $1795 \text{ V}$ are also shown. The beamlet ion density contours in the plots are normalized by the nominal discharge plasma density.
Figure 6.9: Beamlet plasma potential and ion density. Ion density is normalized by \( n_0 = 1.0 \times 10^{17} \text{m}^{-3} \). Potential contour lines are shown for the values from \(-210 \text{ V}\) to \(1790 \text{ V}\) with a step of \(200 \text{ V}\). The \(1780 \text{ V}, 1795 \text{ V}\) and zero potential contour lines are also shown.

The beamlet potential contours on the \(y = 0\) plane are shown in figure 6.10 for typical operating conditions. These conditions represent cross-over, nominal, perveance conditions respectively. Potentials are normalized by the upstream plasma electron temperature, \(T_{e0} = 5 \text{ eV}\).

Comparing the plots in figure 6.9 with those in figure 6.6, we notice that the potential lines from both models are quite close to each other for every plasma condition especially in the upstream and ion optics regions. The density contours are also reasonably close. The smoother density contours obtained by the streamline PIC model is due to the fact that we neglect the random motion of beamlet ions with

(a) Cross-over, \(n_0 = 0.05 \times 10^{17} \text{m}^{-3}\)

(b) Normal, \(n_0 = 1.0 \times 10^{17} \text{m}^{-3}\)

(c) Perveance, \(n_0 = 14.0 \times 10^{17} \text{m}^{-3}\)
Figure 6.10: Beamlet plasma potential. Potential contour lines are shown for the values from $-210$ V to $1790$ V with a step of 200 V. The $1780$ V, $1795$ V and zero potential contour lines are also shown. Potential is normalized by $T_{e0} = 5$ eV.

respect to the bulk motion. However, the beamlet focusing is nearly the same as calculated by both models which is of much importance to determine the amount of extracted beamlet current and access grid impingement current. Therefore, it is acceptable to use the streamline HG-IFE-PIC model for ion optics simulations when beamlet behavior is our concern which gives us the opportunely to tackle problems out of reach of the standard HG-IFE-PIC code. In the following, all beamlet results were obtained using the streamline HG-IFE-PIC model.
6.8.4 Impingement Current Limits

By varying the upstream plasma density from cross-over through perveance conditions, we can scan the operating envelope of an ion optics grid set. For each density, the accel grid impingement current, \( I_i \), is calculated as well as the beamlet current, \( I_b \), itself. The ratio of the impingement current of the accel grid to the beamlet current, \( I_i/I_b \), is plotted against the upstream density, beamlet current and the normalized perveance per unit grid area, in figure 6.11. From the plots, we notice that the impingement current is zero over a wide range of plasma densities, or beamlet currents. This range defines the feasible operation range of the ion optics grid set. The lower limit of this range is the cross-over limit which occurs approximately at \( I_b = 0.017 \) mA. The upper limit which is the perveance limit occurs approximately at \( I_b = 0.464 \) mA. As the density, or beamlet current, goes beyond these limits, the impingement current increases. The zero impingement current in between the cross-over limit and the perveance limit shown is because charge-exchange ions are not included in the simulation. The inclusion of charge-exchange ions will result in small non-zero impingement current in between the limits and shift the entire curve upwards. However, the impingement current limits are nearly unaltered.

The experimental work performed at Colorado State University (CSU) on NEXT subscale gridlets provides measurements for the cross-over limit for different NEXT grid sets operated at several throttling values. These measurements are summarized in figure 6.12 [91]. From which, we estimate a cross-over limit for the grid set and throttling condition studied here of \( \approx 0.035 \) mA which is about twice the cross-over limit predicted by our simulation. It is worth noting here that current ion optics codes, which are widely used, such as CEX2D, CEX3D and igx [47] are generally incapable of predicting the cross-over limit. The only known ion optics code which is capable of predicting the cross-over limit is the ffx code [21]. However, its predictions is about a factor of 2 to 4 lower than the experimentally measured limits [91]. In order to improve the cross-over limit prediction of the streamline HG-IFE-PIC code, we will involve the whole gridlet in the simulation instead of including only two-quarter apertures. This will be discussed in the next section.

6.8.5 Electron Backstreaming Limit

The onset of electron backstreaming in a certain gridlet aperture is defined here as the accel grid voltage at which the potential profile along the aperture centerline becomes monotonic. At which condition, the potential well downstream of the accel grid
which hinders the neutralization electrons from heading towards the accel grid will diminish. The voltage drop along the centerline of either of the apertures considered in the simulation is plotted against the accel grid voltage in figure 6.14. From the figure, we notice that the electron backstreaming limit occurs at $V_a \approx -69.8$ V. The potential profile along the aperture centerline is shown in figure 6.13. Due to their negligible effect on beamlet potential profile along the aperture centerline where the
beamlet ion density is very large, CEX ions are not considered in the estimation of the electron backstreaming limits.

**6.9 Streamline HG-IFE-PIC Simulation: Whole Subscale Gridlet**

**6.9.1 Simulation Setup**

The simulation domain is illustrated in figure 6.4. We consider the same upstream and downstream plasma conditions as in the two-quarter apertures simulation.
Figure 6.13: Potential profile along aperture centerline.

Figure 6.14: Aperture electron backstreaming.

The uniform Cartesian PIC mesh and the Cartesian-based tetrahedral stretched IFE
mesh are setup in the same fashion as we did in the two-quarter simulation. We also select, here, a set of PIC meshes. Each of these meshes can be used to resolve the discharge plasma sheath of a range of discharge plasma densities. Table 6.7 summarizes the mesh parameters for all runs. The PIC and IFE meshes used for the simulation run of $n_0 = 1.0 \times 10^{17} \text{m}^{-3}$ are shown in figure 6.15.

<table>
<thead>
<tr>
<th>Plasma density $\times 10^{17}[\text{m}^{-3}]$</th>
<th>PIC Mesh $\Delta z [\text{m}]$</th>
<th>$n_x \times n_y \times n_z$</th>
<th>IFE Mesh</th>
<th>nodes</th>
<th>elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05 $\leq n_0 \leq 0.2$</td>
<td>$5.2566 \times 10^{-5}$</td>
<td>$88 \times 51 \times 271$</td>
<td></td>
<td>857,208</td>
<td>4,132,500</td>
</tr>
<tr>
<td>0.5 $\leq n_0 \leq 1.0$</td>
<td>$5.2566 \times 10^{-5}$</td>
<td>$88 \times 51 \times 251$</td>
<td></td>
<td>740,520</td>
<td>3,567,000</td>
</tr>
<tr>
<td>2.0 $\leq n_0 \leq 4.0$</td>
<td>$2.6283 \times 10^{-5}$</td>
<td>$175 \times 101 \times 341$</td>
<td></td>
<td>2,244,725</td>
<td>10,962,000</td>
</tr>
<tr>
<td>6.0 $\leq n_0 \leq 10.0$</td>
<td>$1.7522 \times 10^{-5}$</td>
<td>$175 \times 101 \times 331$</td>
<td></td>
<td>1,767,500</td>
<td>8,613,000</td>
</tr>
<tr>
<td>12.0 $\leq n_0 \leq 14.0$</td>
<td>$1.3142 \times 10^{-5}$</td>
<td>$175 \times 101 \times 401$</td>
<td></td>
<td>2,050,300</td>
<td>10,005,000</td>
</tr>
</tbody>
</table>

Table 6.7: PIC and IFE meshes used in the streamline HG-IFE-PIC whole gridlet simulation as a function of the upstream plasma density.

### 6.9.2 Computational Performance

The computational time required to reach steady-state ion trajectories using the standard HG–IFE–PIC model depends on the upstream plasma condition. It varies from less than 3 hours, for densities much less than the cross-over limit, to more than 20 hours for densities much larger than the perveance limit. The number of simulated particles injected at the beginning of each simulation loop was 156,600 particles. It is worth noting here the advantage of using the streamline HG–IFE–PIC to speed up the standard HG-IFE-PIC ion optics simulation such that the time to span the whole operation envelope can be reduced to less than a week of cheap computation time. A comparison among the CPU time of the standard HG-IFE-PIC two-quarter aperture simulation, the streamline HG-IFE-PIC two-quarter aperture simulation and the streamline HG-IFE-PIC whole gridlet simulation at selected upstream plasma densities is shown in figure 6.16. From the plot, we notice the huge reduction in computational time achieved by the streamline ion optics model which makes even the whole gridlet simulations feasible in a reasonable time.
6.9.3 Beamlet Plasma Flow

The potential and beamlet ion density on the $y = 0$ plane which represents the horizontal plane cutting through the middle of the gridlet are shown in figure 6.17 for the nominal operating condition at $n_0 = 1.0 \times 10^{17} \text{m}^{-3}$. The potential and density from a single aperture (two-quarter apertures) model is also shown for comparison. As
noticed from this figure, very slight differences in the density and potential contours can be noticed between the seven and single aperture models. The most noticeable difference is in the potential contours in the neutralization zone which is primarily due to the boundary effects.

Figure 6.18 illustrates the cross-over operating condition at $n_0 = 0.05 \times 10^{17} \text{m}^{-3}$ of the seven-apertures and the single aperture optics. From the figure, we notice the difference in the shape of the plasma sheath of a central aperture and a circumferential aperture due to the edge effect. Therefore, exterior beamlets become asymmetric and diverted towards the axis of the gridlet which explains the larger impingement current predicted by a seven-aperture model than that predicted by a single aperture model as reported in [36].

Figure 6.18 shows the perveance condition at $14.0 \times 10^{17} \text{m}^{-3}$. The figure shows slight difference in potential in the neutralization zone as well.
Raed I. Kafafy  Chapter 6. Ion Optics Simulations

(a) Seven Apertures

(b) Single Aperture

Figure 6.17: Beamlet plasma normalized potential and ion density at the nominal operating condition. The ion density is normalized by the nominal upstream plasma density, $n_0 = 1.0 \times 10^{17} \text{m}^{-3}$, and the potential is normalized by $T_{e0} = 5 \text{ eV}$.

6.9.4 Impingement Current Limits

The operation envelope of the NEXT subscale ion optics gridlet in terms of accel impingement current is estimated. The ratio of the impingement current of the accel grid to the beamlet current, $I_i/I_b$, is plotted against the upstream density, beamlet
Figure 6.18: Beamlet plasma normalized potential and ion density at cross-over. The ion density is normalized by the nominal upstream plasma density, $n_0 = 1.0 \times 10^{17} \text{m}^{-3}$, and the potential is normalized by $T_{e0} = 5 \text{ eV}$.

Current and normalized perveance per unit gridlet area in figure 6.20 Impingement current curves of the two-quarter aperture model are shown for comparison. From the plot, we notice that operation envelope of the whole gridlet model is generally similar to that of the two-quarter aperture model, although the whole gridlet impingement current curve is shifted upward at both ends. This is more noticeable in the cross-
Figure 6.19: Beamlet plasma normalized potential and ion density at perveance. The ion density is normalized by the nominal upstream plasma density, $n_0 = 1.0 \times 10^{17} \text{m}^{-3}$, and the potential is normalized by $T_{e0} = 5 \text{ eV}$.

over region which indicates higher impingement current. This conclusion agrees with our observation from the beamlet behavior of the whole gridlet model, in which we noticed more severe impingement on outer apertures. The cross-over and perveance limits predicted by the two-quarter aperture model and the whole gridlet model are compared with each other in figure 6.21 which focuses on the cross-over and perveance regions. From figure 6.21 we notice that the cross-over limit for the whole gridlet
Figure 6.20: Impingement current curve for the whole gridlet (seven apertures) model as compared with the two-quarter aperture (single aperture) model.

model occurs approximately at $I_b = 0.02$ mA as compared with 0.017 mA for the two-quarter aperture model and to 0.035 mA as estimated by the CSU experiments [91]. Also, the perveance limit for the whole gridlet model occurs approximately at $I_b = 0.469$ mA as compared with 0.464 mA for the two-quarter aperture model. These results addresses the importance of including actual aperture geometric effects, especially when subscale ion optics gridlets are considered.
6.9.5 Electron Backstreaming Limit

The voltage drop along the centerline of each of the three apertures considered in the whole gridlet simulation is plotted against the accel grid voltage in figure 6.23. The voltage drop along the centerline of one of the two apertures considered in the two-quarter aperture model is also shown for comparison. From the figure, we notice that
the electron backstreaming limit the central aperture (aperture 1) in a whole gridlet model occurs at $V_0 \approx -69.75$ which is quite close to the limit of a single aperture in two quarter model. On the other hand, the backstreaming limit of the circumferential apertures (apertures 2 and 3) in the whole gridlet model occurs approximately at $-68.05$ and $-67.95$ which is about 2 V lower than the limit of a single aperture. The potential profile along the centerline of each aperture of the whole gridlet model as well as along the centerline of the single aperture model are shown in figure 6.22. The CEX ions are not considered in the prediction of the electron backstreaming limits as discussed above.

![Potential profile along aperture centerlines.](image)

**Figure 6.22:** Potential profile along aperture centerlines.

### 6.10 Summary and Conclusion

In this chapter, we used the HG-IFE-PIC Ion Optics model to perform ion optics simulations on the NEXT ion optics. Two ion optics models were adopted. A standard HG-IFE-PIC model was first applied to a two-quarter aperture domain, in which ions were injected from the upstream boundary with a Maxwellian velocity distribution and a drift Bohm speed. An adaptive time stepping was employed to speed up the simulation. The CPU time ranged from less than three hours to above
The model was able to predict both perveance and cross-over as well as nominal ion optics performance.

Second, a streamline HG-IFE-PIC was applied to a two-quarter aperture domain. In the streamline HG-IFE-PIC, particles are injected only at the beginning of each simulation loop. No random motion is assumed for the particles. Particles are pushed and deposit their charge at each time step, but the electric field is not update until the end of the simulation loop. The results of the streamline HG-IFE-PIC mode were shown to be in very good agreement with the standard HG-IFE-PIC model. The advantage of the streamline HG-IFE-PIC model is that it employs only a small fraction of the number of particles employed by a standard HG-IFE-PIC. This results in a drastic reduction in CPU time. Moreover, the memory requirements for a streamline HG-IFE-PIC is much less than that of a standard HG-IFE-PIC which allows the tackle of larger size problems. The CPU time ranged from less than 40 minutes to more than 3 hours. The streamline HG-IFE-PIC is capable of predicting the whole operation envelope of an ion optics system using a two-quarter domain in less than one day on a regular workstation! The cross-over and perveance limit were predicted for the studied NEXT ion optics. The predicted cross-over limit was in reasonable agreement with the experimental measurements at CSU using a 37-aperture subscale NEXT gridlet.
Finally, we further applied the streamline HG-IFE-PIC in a whole gridlet seven aperture domain. Due to geometric symmetry, only one quarter of the domain was considered in simulation. The CPU times which was also dependent on plasma density ranged from less than three hours to more than 20 hours. The whole operation envelope of an ion optics system using a whole gridlet seven aperture domain may be obtained in a matter of few days on a regular workstation! The results of the seven aperture model were compared with those of the two-quarter aperture model. It was shown that the central aperture in the seven aperture model was generally in a good agreement with the two-quarter aperture model. On the other hand, the circumferential apertures were different due to the edge effects. The electron backstreaming limit of the circumferential apertures is approximately 2 V lower than the limit of a single aperture, while the electron backstreaming of the central aperture is very close to that of the single aperture model. The cross-over limit predicted by the streamline HG-IFE-PIC model on a whole gridlet domain was in a better agreement with the experimental data than that predicted by the same code on a two-quarter aperture domain.
Chapter 7

Ion Thruster Plume Simulations

7.1 Introduction

In this chapter, we introduce the physical and mathematical modeling of spacecraft–thruster interactions, and the results of the plume interactions simulations performed on the Dawn spacecraft. The basic sources of the spacecraft contamination are first addressed with emphasis on those caused by ion thrusters. Modeling contamination is discussed. The major components of ion thruster near-plume environment are modeled. These are Xe\(^+\) propellant beam ions, Xe propellant efflux neutrals, Xe\(^+\) CEX ions, Mo NPE neutrals, Mo\(^+\) CEX ions, and neutralization electrons. A HG-IFE-PIC Plume model for the simulation of spacecraft-ion thruster interaction is developed.

Simulations of the ion thruster interaction with the Dawn spacecraft are also performed using the HG–IFE–PIC Plume code. The exact geometric and system details of the Dawn spacecraft are still unrevealed. A simplified Dawn model was selected in such a way to retain as much of the known or expected geometrical details as possible and still keep the numerical simulation feasible. Three possible ion thruster arrangements have been studied; a single central thruster, three in-plane thrusters, and a two–by–two thruster array. The effect of each thruster arrangement with selected possible firing options on the surrounding plasma is investigated. The effect of thruster operation in each thruster arrangement and firing option on the deposition of NPE species on spacecraft bus surfaces and solar arrays is also assessed.

All simulations in this chapter were performed on a Dell\(^\circledR\) PC which has a single
Intel® Pentium® 4 processor of 2.2 GHz, and has 1 GB of memory.

7.2 Spacecraft Contamination

A contaminant is defined as any foreign substance in front of or on a spacecraft surface. In the context of spacecraft design, a contaminant can be either a molecular effluent or a solid or liquid particles. Although, it is sometimes difficult to clearly differentiate between gaseous and particulate surface contaminants (e.g. frost and water vapor), gaseous contaminants are assumed here to be individual neutral atoms, molecules, or ions that are either outgassed by materials in space or condense out on spacecraft. Contamination can occur at all phases of a spacecraft mission [24].

The contamination processes include particulate contamination, gaseous-contamination spacecraft charging effects, and spacecraft glow and surface erosion. These processes may significantly degrade the performance of spacecraft systems exposed to the space environment by reducing resolution, light transmission, thermal control, and surface conductivity. These in turn could lead to reductions in mission lifetime and data quality [24].

In this study, we are only concerned with the contamination of spacecraft surfaces due to the backflow of ion thruster plume species. Xe is a gas at typical temperatures of spacecraft surface. Thus, no contamination hazard is expected due to the backflow of Xe$^+$ CEX ions. Mo, on the other hand, is a solid metal at the same temperature range. Condensation of Mo atoms is not likely to happen because the vapor pressure of Mo is extremely low at the spacecraft typical surface temperatures. Desorption of Mo atoms is too small due to the same reason. Hence, the flux of Mo atoms onto spacecraft surfaces will completely stick [65].

7.2.1 Modeling of Contamination

Modeling of contamination on spacecraft is conventionally divided into four steps [24]:

1. determination of the source of the contamination,
2. transport of the contaminants,
3. accommodation of the contaminants on the surface, and
4. determination of the effects of the contaminants on the surface and on the spacecraft operations.

Among all sources of contamination, thruster plumes have an enormous potential for contamination on a spacecraft because of the large amount of mass released during a thruster firing. Thruster exhaust products may reach spacecraft surfaces by different mechanisms.

Collisional transport mechanisms are not significant for an ion thruster where momentum transfer collisions in the plume is very unlike to happen due to the very low plume density. Backflow of plume charge-exchange (CEX) ions, both propellant ($\text{Xe}^+$) and non-propellant ($\text{Mo}^+$), is primarily due to the electric field within and around the thruster plume and around the spacecraft.

Once a flux of contaminant molecules reaches the receiving surface, molecules may either be reflected or adhere to that surface. Contaminants that adhere to the surface can either condense or be adsorbed onto the surface. Condensation occurs when the pressure exerted on by a contaminant gas over a surface exceeds the saturated vapor pressure of the gas at the surface temperature of the material. Adsorption occurs because of surface attraction between individual atoms of the substance and those of the contaminant.

Condensation can be a very serious problem because it easily forms a thick layer on a surface. However, it is usually avoided by using materials emitting a very small fraction of volatile condensible material (VCM). Therefore, adsorption acts as the major adherence mechanism in spacecraft contamination. The degree of adsorption of any individual particle depends on the gas species, the surface temperature, the composition of the substrate, and the amount of surface coverage [24].

Assuming that adsorption is the only source of deposition, the net rate at which mass builds up on a surface is simply

$$\frac{dm}{dt} = \dot{m}_i$$  \hspace{1cm} (7.1)

where $\dot{m}_i$ is the incoming mass flux of contaminants.

The presence of a thin contaminant film on the surface of a material will alter its solar absorptance according to the relation

$$\alpha_s^x = \frac{\int [1 - R_s(\lambda) \exp (-2\alpha_c(\lambda)x)] S(\lambda)d\lambda}{\int S(\lambda)d\lambda}$$  \hspace{1cm} (7.2)
where $\alpha_c(\lambda)$ is the absorptance of the contaminant film (in units of m$^{-1}$), $x$ is the film thickness and $R_s(\lambda) = 1 - \alpha_s(\lambda)$ is the solar reflectance of the uncontaminated material. The contaminant layer will increase the absorptance of the surface material and consequently its equilibrium temperature [74].

In addition to the concern of contamination of thermal control surfaces, there is also the possibility for contamination buildup on optical sensors or solar arrays. The presence of a contaminant film on a lens, mirror or focal plane will degrade the signal to noise ratio (SNR) of the detector and limit the dynamic range by absorbing light from the target of interest. If the contaminant film becomes too thick the sensor will cease to function properly. For IR sensors, which utilize cryogenically cooled surfaces, the problem may be reserved somewhat if the optics can be warmed up so the contaminants will evaporate from the surface [74].

A contaminant film will degrade the output of a solar cell according to the relation

$$F(x) = \frac{\int S(\lambda) I_s(\lambda) \exp \left(-\alpha_c(\lambda)x\right) d\lambda}{\int S(\lambda) d\lambda}$$

(7.3)

where $I_s(\lambda)$ (W/m) is the response of the cell, a measure of how effectively the cell converts that color of light into power. As a rule of thumb, the power output from a cell is decreased by approximately 2% times the contaminant thickness in microns [74].

### 7.2.2 Measuring Molecular Film Thickness

The procedure used to determine the amount of deposited material on a surface, the non-volatile residue (NVR), is denoted by ASTM E 1325. The physical mass deposition of a molecular film may be measured by a Quartz Crystal Microbalance (QCM). Essentially, the QCM operates by comparing the resonant frequencies of two quartz crystals. One crystal is exposed to the environment and the other is shielded. The resonant frequency of the exposed crystal will change if mass is deposited on its surface. Consequently, by measuring the change in resonant frequency, mass deposition can be inferred [74].

### 7.3 Modelling of Ion Thruster Plume

An ion thruster plume consists of the following major components [67]:

---

**Raed I. Kafafy**  
*Chapter 7. Ion Thruster Plume Simulations*  
136
1. fast propellant beam ions that provide thrust,
2. unionized propellant neutrals that originate from the discharge chamber or the neutralizer,
3. slow propellant ions created by CEX reaction collisions between beam ions and propellant neutrals,
4. non-propellant efflux (NPE) which is made of the eroded and sputtered material from thruster grids and discharge chamber (a fraction of these neutrals is ionized mainly by CEX with beam ions),
5. neutralizing electrons which are emitted from the neutralizer.

Each of these components will be discussed in more details below. Other components may also exist in an ion thruster plume such as propellant and non-propellant ions created by electron impact or photo-ionization. Doubly and higher charged propellant ions are typically less than 5% of the total beam ions population [65]. Ions produced by photo-ionization can be easily neglected since the light exposure of the ion thruster plume does not provide enough energy to generate a pronounced ion population as compared to the ions created by CEX reactions. Ions created by electron impact may be an important component depending on the electron temperature, they will not be considered in this study which focuses on beam ions; and propellant and non-propellant ions created by CEX reactions.

### 7.3.1 Ion Beam

The singly-charged propellant ions (Xe\(^{+}\)) are the basic species in an ion thruster plume which travel at velocities greater than 10 km/s. After exiting the thruster grids, beam ions follow nearly straight paths since the fields in the neutralization zone are too weak to disturb their trajectories. A beam expansion cone of 15° – 20° was reported which is primarily due to the fringe electric fields in the grid apertures and secondarily due to the curvature of the grid surface [65].

In this study, we consider a typical NASA Solar Electric Propulsion Technology Application Readiness (NSTAR) ion thruster which utilizes two dished grids with the convex surface facing downstream. As in [65], we assume that the thruster grid is a spherical segment with \(R_c\) radius of curvature, beam ions leave the thruster exit as if originating from a point source located at a distance \(R_c\) behind the thruster.
exit plane. The thruster exit plane is defined as the plane tangent to the accel grid downstream surface at its center point, as illustrated in figure 7.1. The beam divergence angle $\alpha$ can be related to the thruster beam radius $r_T$ and the radius of curvature $R_c$ through

$$R_c = \frac{r_T}{\sin \alpha} \approx \frac{r_T}{\alpha \cos(\alpha/2)}$$

The divergence angle can be determined from the beam radius and the thruster grid depth $d$ using the trigonometric relation

$$\alpha = 2 \cot^{-1}(r_T/d)$$

Over the length scales of interest, the beam ions migrate freely at the initial beam exit velocity $v_b$. The radial beam current density profile may be approximated, in spherical coordinates, by a parabolic axisymmetric core and an exponential wing. The profile is given by

$$j_b(R, \theta) = \begin{cases} 
  j_{b0} \left( \frac{R_c}{R} \right)^2 \left[ 1 - \left( \frac{\theta}{\alpha} \right)^2 \right], & \theta < \gamma \alpha \\
  j_{b0} \exp\left( -\frac{r}{\lambda} \right) \left( \frac{R_c}{R} \right)^2 \left[ 1 - \left( \frac{\theta}{\alpha} \right)^2 \right], & \theta \geq \gamma \alpha 
\end{cases} \quad (7.4)$$

where $R$ is the distance from the point of interest to the beam source point, $r$ is the normal distance to the beam centerline, and $j_{b0} = ev_b n_{b0}$ is the beam current density at thruster exit center point, i.e. the point with coordinates $(R_c, 0)$. Typically, $\gamma$ is taken to be 0.95 and $\lambda$ is taken as a fraction of $r_T$. 

Figure 7.1: Geometry of beam profile.
7.3.2 Neutral Propellant Plume

Due to inefficiencies in the discharge chamber, some of the propellant leave the discharge chamber unionized. The efficiency of a discharge chamber to ionize the propellant is represented by the propellant utilization efficiency $\eta_p$ which is defined as the ratio of the beam ion mass flow rate to the propellant mass flow rate, or

$$\eta_p = \frac{I_b(m_i/e)}{m_p}. \quad (7.5)$$

The density distribution of the neutral plume $n_n(x)$ is modeled analytically as that of a free molecular flow from a point source located at one thruster radius $r_T$ behind the thruster exit. The density distribution of the neutral plume is given by [80].

$$n_n(R, \theta) = a \frac{n_{n_0}}{2} \left(1 - \left[1 + \left(\frac{r_T}{R}\right)^2 \right]^{-1/2}\right) \cos \theta. \quad (7.6)$$

where $R$ is the distance to the point source, $\theta$ is the angle between $R$ and the downstream axis, and $a$ is a correction factor. The geometry of the neutral profile is illustrated in figure 7.2. From which, we notice that $R = \sqrt{r^2 + (r_T + z)^2}$ and $\theta = \tan^{-1}\left(\frac{r}{r_T + z}\right)$. The correction factor $a$ is calculated such that the neutral density at the thruster exit center point, i.e. the point $(r_T,0)$, equals $n_{n_0}/2$. It was shown in [65] that the simplified single point source model given by (7.6) is in reasonable agreement with a more detailed numerical model that accounts for the neutrals flow from all the holes in the grid.


7.3.3 Charge-Exchange Ions

CEX collisions occur between the fast beam ions and the slow neutrals. These collisions result in fast neutrals which follow a line-of-sight, and slow ions which become affected by the strong radial electric field in the beam. Using the analytical profiles for the beam ions and the neutral plume, one could obtain the spatial distribution of the CEX ion production rate \[80]\: 

\[
\frac{dn_{\text{ce}x}}{dt} = \sigma_{\text{ce}x} v_b n_b(x) n_n(x) = \frac{n_b(x) n_n(x)}{n_{b0} n_{n0}} \frac{dn_{\text{ce}x0}}{dt} \tag{7.7}
\]

where \(n_{b0}\) is the average beam ion density at the thruster exit, \(n_{n0}\) is the average neutral efflux density at the thruster exit, \(v_b\) is the beam ion velocity, \(\sigma_{\text{ce}x}\) is the CEX ion collision cross section, and \(\frac{dn_{\text{ce}x0}}{dt}\) is the average charge exchange ion production rate at the thruster exit which is given by 

\[
\frac{dn_{\text{ce}x0}}{dt} = \sigma_{\text{ce}x} v_b n_{b0} n_{n0} \tag{7.8}
\]

Cross-sections of CEX collisions between monatomic ions, such as \(\text{Xe}^+\), and their parent gases can be calculated as a function of the ion velocity \[61\] which may be expressed as 

\[
\sigma_{\text{ce}x} = (k_1 \ln v_b + k_2)^2 \tag{7.9}
\]

where \(k_1 = -0.8821 \times 10^{-10}\text{m}^2\) and \(k_1 = 15.1262 \times 10^{-10}\text{m}^2\) for Xenon \[67\]. The values \(n_{b0}\), \(n_{n0}\) and \(\frac{dn_{\text{ce}x0}}{dt}\) can be obtained from ion engine operation parameters, as suggested by \[80\]. The application of the Direct Simulation Monte Carlo (DSMC) method to compute the neutral plume for the UK-10 ion thruster and examine the accuracy of the analytical neutral plume model of \(7.8\) has shown reasonable agreement between equation \(7.8\) and the DSMC results in the near-axis region but noted substantial errors in the off-axis region \[8\]. Here, equation \(7.8\) is only used to calculate the CEX ion production rate in equation \(7.7\). Because the beam ion density concentrates along the plume axis, CEX collisions occur primarily in the near-axis region \[80\].

7.3.4 Non-Propellant Efflux

The non-propellant metallic species in an ion thruster plume represents a major contamination hazard because of the very low vapor pressure of these species. The
primary source of these metallic species is the sputtered material from the thruster grids which are typically made of molybdenum (Mo). Sputtered Mo neutrals are subject to ionization either by CEX collisions with beam ions, electron impact, or ion impact. The latter two mechanisms are much less likely to happen than the first mechanism\cite{65} and hence will be neglected. The production rate of Mo CEX ions is governed by\cite{7.7}. The cross-section of CEX between Mo and Xe\(^+\) is nearly constant over the range (0.1 – 10 keV) and is measured to be around \(6 \times 10^{-20}\text{m}^2\) [62]. The neutral density of Mo is typically four orders of magnitude less than the neutral propellant density. Thus, we expect the Mo CEX ion production rate to be orders of magnitude less than that of the Xe CEX ions. Hence, the perturbations caused by the Mo CEX ions to the electric field is minimal and can be easily neglected with respect to the field effect due to the Xe CEX ions.

### 7.3.5 Electrons

Electrons are commonly assumed to be isothermal in plasma plume modelling and their density is described by a Boltzmann distribution\cite{80}. Full particle simulations using both ion and electron simulation particles typically requires the use of an artificial ion/electron mass ratio because of the huge difference between ion and electron timescales. Since the plume plasma is mesothermal, that is the directed ion velocity is much larger than the thermal ion velocity, but much less than the electron thermal velocity, one has to use an extremely large ion/electron mass ratio which is believed to be prohibitively expensive\cite{80}. Hence, the assumption of a Boltzmann electron distribution is adopted here as well. The electron density is then given by

\[
n_e = n_{e_0} \exp \left[ \frac{\Phi - \Phi_0}{T_{e_0}} \right]
\]

(7.10)

where \(n_{e_0} = n_{b_0}\) is the average plasma density at thruster exit and \(\Phi_0\) is the plume potential near thruster exit. The electron is assumed to be constant. It was shown in\cite{65} that allowing the electron temperature to vary has a little effect on simulation results. The reference electron density \(n_{e_0}\) is taken here at the thruster exit as in\cite{80} which implies that neutralization is mainly driven by neutralizer electrons. Neutralization due to background electrons can be easily neglected since the interplanetary plasma environment, where our case study satellite is supposed to operate, is observed to have a median density of \(n_\infty \approx 6.9 \times 10^{-6}\text{m}^{-3}\) due to solar wind.

In our model, following\cite{80}, the electron temperature \(T_{e_0}\) and the plume potential
with respect to spacecraft $\Phi_0 - \Phi_{sc}$ are assumed to be known from onboard measurements.

### 7.3.6 Electrostatic Field

The plume plasma may be considered electrostatic. Thus, the Maxwell’s equations governing the transport of the CEX ions reduce to the Poisson’s equation given by

$$-\nabla \cdot \epsilon \nabla \Phi = e(n_i - n_e)$$  \hspace{1cm} (7.11)

where $n_i$ is the total ion number density and $n_e$ is given by the Boltzmann relation (7.10). The components that contribute to the total ion density are the beam ion density, the Xe CEX ion density, and the background ion density. NPE CEX ion density is neglected as discussed above.

### 7.4 Normalization

We apply here the same normalization applied in chapter 6. The Poisson’s equation, the Boltzmann’s distribution, and the Newton’s second law are normalized in exactly the same way. We use the same set of reference variables, in addition to the reference frequency $\omega_{ref} = 1/t_{ref}$. Our selection for the set of reference variables for the plume model is

$q_{ref} = e, \quad m_{ref} = m_i, \quad T_{ref} = T_0 e_{ref} = e_0, \quad n_{ref} = n^*$

where, $n^*$ is the CEX ion density at a reference point outside the ion beam. Then,

$$L_{ref} = \sqrt{\frac{e_0 k T_0}{n^* e^2}} = \lambda_D^*$$

Here, we choose our reference density at a point in the region around the ion beam and near thruster exit as in [80]. This choice allows us to avoid the very high density region in the ion beam which results in a too small Debye length and hence a too fine PIC mesh.
7.5 Simulation Model

7.5.1 Simulation Domain

Our simulation domain includes a geometrically simplified model of the spacecraft under study. This simplification is meant to keep all the important details of the spacecraft that would affect the electric field surrounding the spacecraft and hence the trajectories of the CEX ions backflow. Geometric symmetry is incorporated when available such that only one-half of the spacecraft may be included in the simulation.

7.5.2 Simulation Algorithm

The HG-IFE-PIC plume model solves self-consistently the particle trajectories and space charge for the CEX plasma and the electric field surrounding the spacecraft. Our emphasis here is on studying the backflow transport of CEX ions, hence only the CEX ions are represented by simulation particles.

The number of simulation macro-particles is selected to be statistically large enough; typically few million particles. The distribution of particle population ranges from few thousands in the cells adjacent to the thruster exit plane to few particles in the cells far away from spacecraft.

The simulation is run in three phases. In the first phase, we run the HG-IFE-PIC to generate and trace Xe CEX ions and solve for the electric field until steady-state field and trajectories are obtained. In this phase we ignore the insignificant effect of the Mo$^+$ CEX on the electric field. In the second phase, we fix the electric field and rerun the HG-IFE-PIC code to generate and trace Mo$^+$ CEX ions until steady-state trajectories are obtained. In the final phase, we run the code to generate and trace Mo$^+$ CEX ions for further time steps to collect object-particle intersection information required for the post-processing deposition analysis.

Simulation particles representing CEX ions are generated at each time step according to the volumetric production rate of (7.7). These particles, when created, are given an initial Maxwellian velocity distribution with a temperature corresponding to that of the un-ionized neutral propellant ($\approx 0.04$ eV). Because the initial energy of the CEX ions is negligible compared to the plume potential, the initial velocity distribution plays no role in the final results [80].
Boundary Conditions

The outer boundary conditions is assumed Neumann with $\frac{\partial \Phi}{\partial n} = 0$ on all the sides of the simulation domain either because of symmetry or to impose open boundaries. The potential inside solid objects are fixed to their known values.

Spacecraft bus, payloads, antennas and the backside of the solar arrays are conductors. Hence, they are given fixed potential equal to the spacecraft uniform ground potential $\Phi_{sc}$. The front face of the solar array is covered by glass and, thus is an insulator. It typically has a surface voltage of $-T_e$ (in electron volts), where $T_e$ is temperature of the electrons surrounding the solar array. Solar arrays are modeling as an infinitesimally thin plate. Hence, to distinguish between the potentials of the front and back sides of the solar array is not possible. Instead, we set the potential of the solar array to the spacecraft ground potential.

Absorption particle boundary conditions are applied on all outer surfaces of the domain except on planes of symmetry where reflection boundary conditions are applied. At spacecraft surfaces, hitting ions are assumed to be adsorbed or neutralized then desorbed. Either way, particles hitting solid surfaces are taken away from the particle array.

Domain and Mesh Size

In order to preform a proper PIC simulation, the PIC mesh resolution is determined by the Debye length of the CEX plasma in the region outside of the ion beam as mentioned above, i.e. $\lambda_D^*$. The mesh size is chosen to be of the same order of magnitude as the local $\lambda_D^*$ in the region surrounding the ion beam near the thruster exit [80]. This mesh size is always smaller than the local $\lambda_D$ in the backflow region. Typically, the local CEX plasma density is not known a priori and the mesh size needs to be determined by trial and error.

As mentioned before, the HG-IFE-PIC model allows us to separate the PIC mesh used for sampling particle charge from the IFE mesh used for sampling and solving the electric field. Therefore, the size of the IFE mesh cells can be varied according to the potential gradients and local plasma conditions. In our simulation, we keep the IFE mesh cell size equal to the PIC mesh cell size in the region surrounding spacecraft where CEX plasma density is rather high and potential gradients are very large in the CEX plasma sheath region. In the backflow region, where CEX plasma density and potential gradients are much smaller, we stretch the IFE mesh in all
directions away from the spacecraft keeping the maximum mesh cell size smaller than the reference Debye length $\lambda_D^*$. 

**Time Step**

The simulation time step is calculated based on a Courant condition to assure stable calculations and minimize trajectory errors as described in chapter 5. Since the velocity of the CEX ions increases as they move along the electrostatic field lines, the constant simulation time step should be calculated based on the maximum velocity of CEX ions. The maximum velocity of CEX ions can be simply estimated from the conservation of energy as

$$v_{\text{max}} = \sqrt{2(\Phi_0 - \Phi_{sc}) + (v_{i0})_{\text{max}}^2}$$

where $(v_{i0})_{\text{max}}$ is the maximum initial velocity a CEX ion is given at birth. For the pseudo-Maxwellian distribution we are using, $(v_{i0})_{\text{max}} = 3v_t$.

### 7.5.3 Deposit Calculation

The calculation of the particles deposition on spacecraft bus and solar array surfaces is done after the steady-state trajectories of the traced deposition particles are obtained. Here, the deposition particles are Mo$^+$ CEX ions. The potentials are fixed and the Mo$^+$ CEX ions are traced for further tens of time steps to collect statistically enough number of particles. The particles data are output to a file for the deposition post-processing. The deposit analysis code reads in the surface mesh information of the spacecraft and the data of the collected particles.

The code checks the intersection of the collected particles with the provided spacecraft surface mesh and counts the number of particles passing through each triangle in the surface mesh of each entity of the spacecraft. The flux of particles deposition on each surface can be calculated from

$$Q = \frac{N}{\tau A}$$  \hspace{1cm} (7.12)

where $Q$ is the average flux of $N$ particles through an area $A$ within a time $\tau$. The deposition of solid particles on a specific surface can be related to the flux of the particles through this surface using the following relation

$$\frac{d\xi}{dt} = Q \frac{M}{\rho},$$  \hspace{1cm} (7.13)
where \( \frac{dc}{dt} \) is the deposition rate of particles, \( M \) is the molecular weight of the deposit particles, and \( \rho \) is the density of solid phase of the deposited particles material. Note that we assumed here that the particles are uniformly deposited on the surface and the density of the deposit material is determined by surface temperature. The engineering unit commonly used to measure particle deposition rate on surfaces is Å/khr where 1 Å/khr = \( \frac{1}{3.6 \times 10^{16}} \) m/s.

### 7.6 Dawn Spacecraft

Dawn is a planetary space science mission funded by NASA's Discovery Program. The primary scientific objective of Dawn is to advance our understanding of the origin and evolution of the solar system by studying two of the largest asteroids, Vesta and Ceres, which appear to have remained intact since their formation 4.6 billion years ago. The Dawn mission will launch in May 2006 [75]. It will study Vesta beginning in July 2010, and Ceres beginning in August 2014. Powered by solar electric ion propulsion, the Dawn spacecraft is planned to cruise to the main asteroid belt between Mars and Jupiter where it rendezvous with and orbits Vesta for one year.

The Dawn spacecraft will be the first purely scientific mission powered by solar electric ion propulsion. Ion propulsion will provide the additional velocity needed to reach Vesta once the spacecraft separates from its Delta rocket. Ion propulsion will also be used during asteroid proximity operations to raise and lower the orbit altitude. Dawn’s ion propulsion system is based on the technology successfully demonstrated by Deep Space 1 (DS-1). Figure 7.3 [50] illustrates the Dawn spacecraft with deployed solar arrays.

#### 7.6.1 Spacecraft

The Dawn spacecraft bus and the propulsion module is nearly a cube. In the local coordinate system, the spacecraft orientation is such that the thrust vector is along the \( z \) axis and the solar array is almost always parallel to the \( y-z \) plane. The solar concentrator arrays with refractive linear element technology (SCARLET) solar array have a voltage span of 100 V.
7.6.2 Ion Thruster

The Dawn spacecraft has three gimballed ion engine modules. Each of them is a typical 30-cm NASA Solar Electric Propulsion Technology Application Readiness (NSTAR) Xe ion thruster which was successfully used on DS1 [53]. An NSTAR thruster has an input power range of 500-2300 W. The propellant Xe ions are accelerated through a Mo grid to form a beam with an energy up to 1100 eV (exit beam velocity of $v_b \approx 3.5 \times 10^6$ cm/s) and a current $I_b$ up to 1.8 A. At full thrust level, the thruster produces a thrust of 92 mN and a specific impulse of about 3100 s. Measurements have shown that the propellant ions form a divergent beam with a divergence half angle of about $15^\circ - 20^\circ$ due to the fringe electric fields in the grid apertures and the curvature of the thruster exit surface. The ion beam is kept quasi neutral by emitting electrons from the neutralizer [81].

The propellant that remains un-ionized flows out of the thruster exit in free molecular flow with a thermal speed corresponding to the thruster wall temperature of $T_w \approx 500$ K (0.04 eV). The density of the neutral plume typically remains quasi steady due to the low charge-exchange collision rate. One can estimate the average neutral density at thruster exit $n_{n0}$ from the measured main flow rate, cathode flow rate, and the discharge propellant efficiency $\eta_d$. If the un-ionized propellant exits through the grids in free-molecular flow with a temperature close to that of the thruster
discharge chamber walls $T_w$, $n_{no}$ can be calculated from [80]

$$n_{no} = \frac{\dot{N}_n}{A_n \sqrt{\frac{8kT_w}{\pi M_{Xe}}}}$$

(7.14)

where $\dot{N}_n$ is the number of Xe atoms flowing out of the thruster per second and is obtained from the discharge chamber flow rate. $A_n$ is the flow-through area through the grids and is about 0.24 of the thruster exit area for the NSTAR thruster.

### 7.7 Spacecraft Model

In simulation, the local coordinates of the model spacecraft is rotated by 180° around the +$y$ axis. Thus, the thruster module becomes on the +$z$ face of the spacecraft bus and the antenna becomes on the -$x$ axis face of the spacecraft bus. This rotation does not affect the simulation results in any way. The spacecraft is geometrically modeled to include as much significant details as possible. The spacecraft bus is modeled as a 1.32 × 1.32 × 1.32 m cube. Each ion thruster in its shroud is modeled as a circular cylinder with a radius equal to the outer radius of the shroud. The plume of a firing ion thruster is modeled as discussed above. Major payloads are modeled as three boxes of different sizes. The communication antenna is modeled as a spherical segment which is pointing to the -$x$ direction.

We have studied three ion thruster configurations. The first configuration (CASE 0) is considered as the reference configuration. This configuration has one NSTAR ion thruster attached to the spacecraft and pointing in the +$z$ axis direction. In the second configuration (CASE 1), we add two more thrusters which are inclined by $+/- 45°$ to the +$z$ axis and lie in the same $x$-$z$ plane as the central main thruster. In the last configuration (CASE 2), we attach an array of four thrusters to the spacecraft which are all pointing in the +$z$ axis. Figure 7.4 illustrates the three studied spacecraft-thruster configurations. For each of these configurations, we operate the thrusters in all significant possible firing options.
7.8 Simulation Domain

The simulation domain is setup to include the region of significant physical phenomena. The PIC mesh is a uniform Cartesian mesh with $105 \times 54 \times 90$ cells. Each cell is a 6-cm cube. The cell size is selected to resolve the Debye length of the CEX plasma in the wake region. The IFE mesh is a multi-zone, stretched Cartesian-based tetrahedral mesh which has 830,465 tetrahedral elements. The inner zone, which surrounds the spacecraft bus, has no stretching and has a Cartesian cell size equal to the PIC cell size; each of the IFE cells is further partitioned into five tetrahedrons. The outer zones are stretched according to the above stretching rule in all directions. The stretching parameter is limited in such a way to preserve an acceptable accuracy of the IFE field solver. Figure 7.5 shows a 3-D view of the outer boundaries of the IFE mesh used which is cut away to show the relative location of the spacecraft model.
Figure 7.5: A 3D view of the IFE mesh. The mesh is cut away to illustrate relative spacecraft position.

7.9 Simulation Setup

The input parameters for our simulation model are $n_{b0}$, $n_{n0}$, $\frac{dn_{ex}}{dt}$, $T_{e0}$, and $\Phi_0 - \Phi_{sc}$. We obtain these parameters from engine operating parameters. We setup The spacecraft components and solar arrays are setup to have a negative bias voltage of $\Phi_{sc} = -10$ V and the plume potential near thruster exit is setup as $\Phi_0 = 9$ V. The plume plasma parameters are selected as:

$$T_{e0} = 1.25 \text{eV}, \quad n_{b0} = 1.5220 \times 10^{15} \text{m}^{-3}, \quad n_{n0} = 2.3 \times 10^{17} \text{m}^{-3},$$

$$\frac{dn_{ex}}{dt} = 10^{19} \text{m}^{-3}.\text{s}^{-1}, \quad \text{and} \quad n_\infty = 7.61 \times 10^6 \text{m}^{-3}$$

The number of simulation macro-particles was selected to be statistically large enough; typically 5–6 million particles. The simulation is run in three phases. First, we run the HG-IFE-PIC to generate and trace Xe$^+$ CEX ions and solve for the electric field
until steady-state field and trajectories are obtained. In this phase we ignore the insignificant effect of the Mo\(^+\) CEX on the electric field. Secondly, we fix the electric field and rerun the HG-IFE-PIC code to generate and trace Mo\(^+\) CEX ions until steady-state trajectories are obtained. Finally, we run the code to generate and trace Mo\(^+\) CEX ions for further time steps to collect object-particle intersection information required for the post-processing deposition analysis. The first phase takes less than 4 hours on a Dell\(\textsuperscript{\textregistered}\) PC which has a single Intel\(\textsuperscript{\textregistered}\) Pentium\(\textsuperscript{\textregistered}\) 4 processor of 2.2 GHz, and has 1 GB of memory, while the second phase takes less than an hour and the third takes less than 20 minutes.

### 7.10 Results: CASE 0

CASE 0 has only one possible firing option (denoted by 0A). The history of simulation particles is shown in figure [7.6](#). The steady state number of Xe\(^+\) CEX simulation particles is about 5.5 million particles.

![Figure 7.6: History of number of particles for Xe\(^+\) CEX ions simulation for CASE 0A. The simulation time is given in units of time steps.](image)
7.10.1 Plasma Diagnosis

The plasma potential and total ion charge density contours on two cutting planes, the \(x-z\) plane which is the plane of symmetry and the \(y-z\) plane, are shown in figure 7.7. Potential and total ion charge density iso-surfaces at selected values are also shown to illustrate the 3D structure of the plasma parameters.

![Plasma properties for CASE 0A. Potential is normalized by 5 eV and ion charge density is normalized by 7.61 x 10^{10} m^{-3}.](image)

(a) Potential  
(b) Ion charge density

Figure 7.7: Plasma properties for CASE 0A. Potential is normalized by 5 eV and ion charge density is normalized by \(7.61 \times 10^{10} \text{m}^{-3}\).

From figure 7.7, we notice the mushroom-like shape of the potential and ion density iso-surfaces around the thruster beam axis. This shape is due to the combined effects of the ion beam core and the charge distribution of the CEX ions. If we examine the potential plot, we can easily distinguish the plasma sheath around the spacecraft components. The sheath is very thin in the regions close to the thruster exit and becomes very large in the wake region. Also from the potential plot, we can see a wind-like effect of the backflowing CEX ions which pushes the potential iso-surfaces around the spacecraft in the opposite direction to thruster exit.

The trajectories of the \(\text{Xe}^+\) CEX ions are shown in figure 7.8. From this figure, we notice the backflow of the CEX ions toward spacecraft. We also notice the asymmetric effect of the antenna, which has a potential of \(\Phi_{sc}\), on the CEX ion trajectories.
7.10.2 Deposition Diagnosis

The deposition rate, measured in Å/khr, of Mo\(^+\) CEX ions on the spacecraft bus surfaces and solar array are shown in figure 7.9. From the plot, we notice that a large deposition rate occurs on the bus surface where the thruster module is attached with the maximum value concentrated just behind the thruster module. Significant deposition is also collected by the external surface of the thruster module itself. The solar arrays collects NPE contaminant mostly in the region close to the closest corner to the thruster exit giving an L-shaped deposition profile. The maximum deposition rate on the solar arrays is 0.01442 Å/khr and the average is 0.00028 Å/khr. The antenna seems to be also affected by the NPE contaminant especially on the closest side to the thruster exit. All payloads have nearly zero contamination which indicates the safe location they are placed at.

7.11 Results: CASE 1

In CASE 1, the spacecraft has three thrusters; a central main thruster (T1) pointing in the \(+z\) axis direction, and two side thrusters (T2 and T3) which are inclined by \(+/- 45^\circ\) to the \(+z\) axis and lie in the same \(x-z\) plane as the central main thruster.
This configuration has three possible firing options as illustrated in figure 7.10. In the first firing option (denoted by 1A), only the main thruster T1 is fired. In the second firing option (denoted by 1B) only the side thruster T2 is fired. Finally, in the third option (denoted by 1C), all thrusters are fired.
The history of simulation particles for all firing options is shown in figure 7.11. The steady state number of Xe$^+$ CEX simulation particles is about 5.5 million particles.

(a) Case 1A

(b) Case 1B

(c) Case 1C

Figure 7.11: History of number of particles for Xe$^+$ CEX ions simulation for CASE 1 with all possible firing options. The simulation time is given in units of time steps.
7.11.1 Plasma Diagnosis

CASE 1A

Figure 7.12 shows the plasma properties of CASE 1A. From the figure, we notice that the potential and ion charge density are very close to those of CASE 0A. The potential and ion density have the same mushroom-like profile around the middle firing thruster (T1). The existence of the side thrusters which are negatively biased at $\Phi_{sc}$ attracts CEX ions toward the side thrusters shroud. The sheath structure is very similar to the sheath in CASE 0A. The same wind-like effect of the backflowing CEX ions is still noticed here.

![Figure 7.12: Plasma properties for CASE 1A. Potential is normalized by 5 eV and ion charge density is normalized by $7.61 \times 10^{10} \text{m}^{-3}$.](image)

The trajectories of the Xe$^+$ CEX ions are shown in figure 7.13. From the figure, we notice the backflowing of the CEX ions toward spacecraft. We also notice the slight asymmetric effect of the antenna, which is biased to $\Phi_{sc}$, on the CEX ion trajectories such that they are twisted toward the antenna. Both sides of the solar arrays have nearly the same likelihood to get contaminated.
Figure 7.13: Xe\(^{+}\) CEX ion trajectories for CASE 1A.

**CASE 1B**

Figure 7.14 shows the plasma properties of CASE 1B. It is shown that the potential and ion density are very distorted from those of CASE 0A and CASE 1A. The potential and ion density still have the same mushroom-like profile around the side firing thruster (T2). The sheath structure is completely distorted by firing the side thruster. The sheath on the solar arrays side which faces the thruster is much thinner than on the other side. The CEX ions seem to be backflowing in the opposite direction to thruster exit. The wake region also seems to be rotated about \(-45^\circ\) around the \(+y\) axis.

Figure 7.15 shows the trajectories of the Xe\(^{+}\) CEX ions which look strongly twisted due to the firing of the side thruster. Trajectories of many particles are heading directly into the \(+x\) axis direction. Thus, the solar arrays side facing the \(-x\) axis is exposed to NPE contamination much more than the other side.

**CASE 1C**

Figure 7.16 shows the plasma properties of CASE 1C. We notice the combined effects of all thrusters on the potential and ion charge density. The ion charge density is larger than CASE 1A and CASE 1B meaning that more CEX ions are produced which
Figure 7.14: Plasma properties for CASE 1B. Potential is normalized by 5 eV and ion charge density is normalized by $7.61 \times 10^{10} \text{m}^{-3}$.

Figure 7.15: Xe$^+$ CEX ion trajectories for CASE 1B.

is expected. The mushroom-like shapes of the potential and ion density around the fired thrusters are overlapped. The sheath structure is affected by the higher CEX ion density which results in a thinner and nearly closed sheath. The wake region behind the spacecraft becomes much smaller. A much stronger wind-like effect of
the backflowing CEX ions is noticed here.

![Figure 7.16: Plasma properties for CASE 1C. Potential is normalized by 5 eV and ion charge density is normalized by $7.61 \times 10^{10} \text{m}^{-3}$.](image)

The trajectories of the Xe\(^+\) CEX ions are shown in figure 7.17. The trajectories of the backflowing CEX ions are nearly symmetric because of the symmetric firing. However, the antenna still distorts this symmetry and attracts the CEX ions towards its side. Both sides of the solar arrays seem to suffer from large NPE impingement.

### 7.11.2 Deposition Diagnosis

**CASE 1A**

The deposition rate of the Mo\(^+\) CEX ions on the spacecraft bus surfaces and solar array for CASE 1A is shown in figure 7.18. From this figure, we observe that deposition pattern is quite similar to that of CASE 0A. The L-shaped deposition profile is nearly the same. The maximum deposition rate on the solar array is 0.01585 Å/khr and the average is 0.00029 Å/khr. These values are a little larger than those of CASE 0A, which may be explained by the attracting effect of the side thrusters. The same conclusions can be inferred, here, for the contamination of the antenna and payloads. The inner sides of the side thrusters are suffering from noticeable NPE contamination.
Figure 7.17: \( \text{Xe}^+ \) CEX ion trajectories for CASE 1C.

Figure 7.18: Deposition flux of Mo atoms for CASE 1A.

CASE 1B

Figure 7.19 shows the deposition rate of \( \text{Mo}^+ \) CEX ions for CASE 1B. For this case,
we notice that a large deposition rate occurs on the bus edge closest to the firing side thruster. Significant deposition is also collected by the external surface of the thruster module itself. The solar arrays contamination occurs mostly on the $-x$ side. The maximum deposition rate on the solar array is 0.00163 Å/khr and the average is 0.00036 Å/khr. It is interesting to note that solar arrays contamination in this case is less than the reference case and CASE 1A. The middle thruster have little contamination and the other side thruster is nearly shielded from contamination. Larger antenna contamination is noticed in this case which indicates the possible danger of this firing mode on spacecraft communications.

CASE 1C

Figure 7.20 shows the deposition rate in CASE 1C where all thrusters are firing. Large contamination by the NPE is noticed on the whole bus surface where the thrusters are attached. The bus side surfaces are also affected by this firing mode which is also noticed in the antenna. However, the payloads still look uncontaminated, or negligibly contaminated which is explained by the twisting effect the antenna has on CEX ion trajectories. Thruster modules are severely contaminated. The solar arrays also suffers from much larger contamination which is still concen-
trated towards the lower front corner of the solar arrays. The maximum deposition rate on the solar array is 0.0212 Å/khr and the average is 0.0010 Å/khr.

### 7.12 Results: CASE 2

In CASE 2, the spacecraft has four thrusters arranged in a two-by-two array. All thrusters are parallel and pointing into the +z axis direction. The thrusters are denoted by T1, T2, T3, and T4. We study two possible firing options of this configuration as illustrated in figure 7.21. In the first firing option (denoted by 2A), the thruster pair T1 and T2 are fired. In the second firing option (denoted by 2B), all the thrusters are fired altogether.

The history of simulation particles for all firing options is in shown in figure 7.22. The steady state number of Xe+ CEX simulation particles is slightly less than 6 million particles for CASE 2A and about 5.5 million for CASE 2B.
Figure 7.21: Firing options of CASE 2.

Figure 7.22: History of number of particles for Xe$^+$ CEX ions simulation for CASE 2 with all possible firing options. The simulation time is given in units of time steps.
7.12.1 Plasma Diagnosis

CASE 2A

The plasma properties of CASE 2A is shown in figure [7.23]. From which, we notice that the potential and ion charge density is distorted from those of the reference configuration. The potential and ion density still have the same mushroom-like profile around the fired thrusters (T1 and T2) but now it looks wider. The asymmetry of the firing mode results in asymmetry in the sheath structure which looks much thinner on the bus side and solar arrays closest to the fingering thruster pair. Larger CEX ion density is noticed and expected since two thruster are firing.

![Figure 7.23: Plasma properties for CASE 2A. Potential is normalized by 5 eV and ion charge density is normalized by $7.61 \times 10^{10} \text{m}^{-3}$.](image)

The trajectories of the Xe$^+$ CEX ions are shown in figure [7.24] which shows asymmetry in the backflowing of the CEX ions because of the antenna in addition to the firing asymmetry. Both sides of the solar arrays are likely to get contaminated with larger possibility on the $+x$ side. Payloads seem to be off the trajectories of CEX ions.
CASE 2B

Figure 7.25 shows the plasma properties of CASE 2B. The potential and ion density are similar to CASE 2A but the symmetry is nearly revived. The potential and ion density still have the same mushroom-like profile around the firing thrusters. The sheath structure is nearly symmetric and much thinner than CASE 2A and the reference case. The sheath surface also seems to be closed. The wake region also seems to much smaller.

Trajectories of the Xe\(^+\) CEX is shown in figure 7.26. They are more symmetric and less twisted. Both sides of the solar arrays are likely to get contaminated. The payloads are also off the contaminant trajectories.

7.12.2 Deposition Diagnosis

CASE 2A

The deposition rate for CASE 2A is shown in figure 7.27. We notice that a large deposition rate occurs on the bus surface behind the firing thrusters. The non-firing thrusters are severely contaminated. More contamination on the antenna is observed,
Figure 7.25: Plasma properties for CASE 2B. Potential is normalized by 5 eV and ion charge density is normalized by $7.61 \times 10^{10} \text{m}^{-3}$.

Figure 7.26: Xe\textsuperscript{+} CEX ion trajectories for CASE 2B.

in this case, than the reference case. The solar arrays looks less contaminated than the reference case. The maximum deposition rate on the solar array is 0.00454 \AA/khr and the average is 0.00034 \AA/khr. The payloads are nearly clean.
CASE 2B

Figure 7.28 shows the deposition rate in CASE 2B. Here, the whole bus surface behind the thrusters is badly contaminated which is similar to CASE 1C. The antenna is also severely contaminated, as well as the thruster shrouds. Contamination is even more than in CASE 2A. Significant deposition is also collected by the external surface of the thruster module itself. The solar arrays early have the same L-shaped deposition profile as in the reference case but the values are much higher. The maximum deposition rate on the solar array is 0.02404 Å/khr and the average is 0.00140 Å/khr. These are the highest values in all cases. Despite the extremely severe contamination observed in this case, the payloads are quite clean and look well protected from the NPE contaminants.

7.13 Summary and Conclusion

In this chapter, we used the HG-IFE-PIC Plume model to study the interaction between a model for the Dawn spacecraft and the plume of the attached ion thruster(s). Three thruster configurations of the model Dawn spacecraft were studied with the
associated thruster firing options. It was shown that a firing ion thruster creates a CEX plasma environment around the thruster which has a mushroom-like shape with its head surrounding the thruster exist and its leg surrounding the plume axis. This distinguished profile of the CEX plasma of an ion thruster results from the interaction between the beam ion density and the CEX ion density. The backflow of CEX ions results in a wind-like effect on the CEX plasma which affects the CEX plasma sheath profile around the spacecraft and solar arrays. Symmetric firing results in a quasi-symmetric sheath. This symmetry is distorted by the existence of the large antenna body which is negatively biased at the spacecraft ground potential. Firing more than one thruster at the same time results in a more complicated CEX plasma profiles especially when the thrusters are not parallel.

The trajectories of the CEX ions show that the CEX ions created in the thruster plume suffers from a strong outward radial electric field. As the CEX ions move away form the influence zone of the thruster plume, they become more affected by the outside plasma potential gradients which turn the trajectories toward the sheath of the negatively biased spacecraft. Including the large antenna body in the simulation was shown to have a noticeable effect on the CEX ion trajectories which are generally attracted toward the bus side where the antenna is located. Firing only the side thruster in CASE 1B, results in directing most of the CEX ion trajectories
away from the spacecraft bus and solar arrays. Firing more than one thruster at the
same time results in increasing the intensity of the ion trajectories which intersect
with the spacecraft bus and solar arrays.

We have also assessed the contamination of the spacecraft solar array by NPE con-
taminants resulting from ion thrusters in selected firing options for the studied con-
figurations. It was shown that the solar arrays zone which is closest to the firing
thruster suffers from larger contamination than the rest of the solar arrays surface.
It was also shown that contamination of the solar arrays greatly depends on the
firing option. Symmetric firing options results in higher contamination levels than
asymmetric firing option. We also found that the more thrusters are fired, the larger
the contamination of the solar arrays is. Table 7.1 summarizes the maximum and
average deposition rates of Mo\(^+\) CEX ions on the solar array for each spacecraft-
thruster configuration and firing option. The numbers shown are based on the 1
Å/khr average deposition derived for DS1 ML–71 [81]. The antenna, in general, is
more likely to get contaminated on the part of the surface which is closest to the
fired thruster. Firing the side the closes side thruster to the antenna in CASE 1C
results in the worst contamination situation. The payloads, because of their location
and size, are generally quite clean and free from any significant NPE contamination.
The bus surface on which the ion thrusters are installed suffers from severe NPE con-
tamination which becomes much worse when more than one thruster is firing. The
ion thrusters themselves are liable to severe contamination on their shrouds which
is not of great danger as for the sensitive spacecraft components such as the solar
arrays, the payloads and the communication antenna.

<table>
<thead>
<tr>
<th>CASE</th>
<th>0A</th>
<th>1A</th>
<th>1B</th>
<th>1C</th>
<th>2A</th>
<th>2B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum [Å/khr]</td>
<td>0.01442</td>
<td>0.01585</td>
<td>0.00163</td>
<td>0.0212</td>
<td>0.00454</td>
<td>0.02404</td>
</tr>
<tr>
<td>Average [Å/khr]</td>
<td>0.00028</td>
<td>0.00029</td>
<td>0.00036</td>
<td>0.0010</td>
<td>0.00034</td>
<td>0.00140</td>
</tr>
</tbody>
</table>

Table 7.1: Maximum and average deposition rates of Mo\(^+\) CEX ions on the solar
array.
Chapter 8

Conclusions

8.1 Introduction

In this chapter, we review the research work that has been conducted in this study. We summarize the results found and describe the contributions that have been made to the different aspects of science and engineering related to this work. Recommendations for future research work are, finally, provided.

8.2 Summary of Research

In this work, a new plasma simulation algorithm was developed. This new algorithm is capable of handling large scale simulations which involve complex geometric details while retaining the accuracy of body-fitting grid codes. The new model is based on the three-dimensional immersed finite element method which has been developed in this work and a modified legacy particle-in-cell code. The model also applies a new meshing technique that separates the field solution mesh from the particle pushing mesh in order to increase the computational efficiency of the model.

The new simulation model was used to study two problems of great importance to the development of ion propulsion technology: the ion optics performance and the interaction between spacecraft and the ion thruster. The new model was used to perform ion optics simulations on the NEXT ion optics as well as spacecraft–ion thruster interaction simulation on the Dawn spacecraft. The results of the ion optics
simulations demonstrated good agreement with the available experimental data. It also emphasized the importance of involving circumferential apertures in subscale ion optics simulations. The spacecraft–ion thruster interaction simulations investigated the contamination effect of firing a variety of ion thruster configurations on the sensitive components of the Dawn spacecraft.

8.3 Contributions to Finite Element Analysis

The contribution of this study to finite element analysis is summarized in the following:

- A new three-dimensional immersed finite element (IFE) method was developed. The new method is capable of solving interface boundary value problems on structured, even Cartesian, domains regardless of the shape and location of the interface.
- The existence and uniqueness of the new associate IFE basis functions were proven. It was also shown that the new IFE method possesses the property of partition of unity, and is consistent with the classical finite element method.
- It was shown, through numerical experiments, that the new IFE method provokes a second order convergence of the $L^2$ error and a first order convergence of the $H^1$ error.

8.4 Contributions to Plasma Simulation

The contribution to plasma simulation includes:

- A new electrostatic field solver was developed based on the IFE method. The new field solver is capable of solving the Poisson’s equation with complex boundaries on a Cartesian mesh while retaining a second order accuracy.
- A new Particle-in-Cell algorithm (IFE-PIC) was developed based on a modified legacy PIC code and the recently developed IFE field solver. The new PIC model retains the computational efficiency of a standard Cartesian PIC code as well as the second order accuracy of a finite element field solver.
• A new meshing technique was introduced to enhance the computational efficiency of the IFE-PIC model to be able to tackle even larger scale problems. In the new model, the hybrid-grid IFE-PIC (HG-IFE-PIC), the IFE and PIC meshes are separated. The PIC mesh is a uniform Cartesian to retain the efficiency of a Cartesian PIC code. The IFE mesh is a stretched Cartesian-based tetrahedral mesh to reduce the computational time and memory requirements. IFE mesh stretching follows the potential gradients and local plasma conditions as imposed by physics.

8.5 Contributions to Ion Optics Modeling

The following lists the contribution to ion optics modeling:

• Two ion optics models have been developed: a standard HG-IFE-PIC ion optics model, and a streamline HG-IFE-PIC ion optics model.

• Ion optics simulations on the NEXT ion optics was performed using both models in a two-quarter aperture domain. The results from both models were very close to each other, although the streamline model was much superior in terms of computational time. The computational time of the streamline model is about one order of magnitude less than that of the standard model.

• The whole operation envelope of the NEXT ion optics was assessed by the streamline HG-IFE-PIC model as well as electron backstreaming limits. The cross-over limit predicted by the model was about one-half that reported by experiments.

• The streamline HG-IFE-PIC model was also used to perform ion optics simulations on a whole subscale NEXT ion optics model. The whole operation envelope as well as electron backstreaming of all apertures were assessed. The cross-over limit prediction of the whole ion optics simulation was in a better agreement with experimental data than predictions of the two-quarter aperture ion optics model.
8.6 Contributions to Spacecraft-Ion Thruster Interaction Modeling

The contribution to spacecraft-ion thruster interaction modeling is listed below:

- A HG-IFE-PIC plume model was developed. The model is used to perform spacecraft-ion thruster plume interaction simulation on a model Dawn spacecraft. The model spacecraft was included most of the details of the Dawn spacecraft such as bus, payloads, solar arrays, and antenna. Several ion thruster configurations and thruster firing options have been investigated.

- Using the HG-IFE-PIC plume model, the charge-exchange plasma environment of the ion thruster plume was diagnosed.

- A contamination model was also developed to estimate the deposition rate of Mo CEX ions on spacecraft surfaces and solar arrays. Maximum and average deposition rates of Mo on solar arrays have been assessed.

8.7 Recommended Future Work

Beyond what has been achieved in this work, there remains opportunity for improvements. In the following, our recommendations for future research related to this work are summarized:

- **Parallel Implementation.** The current hybrid-grid immersed-finite-element particle-in-cell (HG-IFE-PIC) model was efficient enough that we were able to tackle 3D plasma simulation problems using cheap computing resources like regular PCs or workstations. However, the HG-IFE-PIC model is still incapable of handling more challenging problems which is larger in size. Examples include ion optics simulations which involve more apertures such as 19 or 37 apertures. To include the whole downstream region of charge-exchange production may need to extend the current domain length in the beam direction two or three times. Another example is the simulation of spacecraft-ion thruster interaction in a domain large enough to include the whole length of the solar arrays and as large of the surrounding space as possible. The mentioned examples suggest problem sizes about ten times the size of the currently tackled problems which
seem impossible using current serial computing resources. Parallel computing is strongly recommended here. The whole problem may be decomposed into many smaller size problems to be distributed on many processing nodes. Initial results that have been obtained but not provided in this work show that there is a great potential there.

- **Ion Optics Physical Model Enhancement.** The ion optics physical model applied in this work has shown reasonable agreement in predicting the current impingement limits of the studied ion optics grid with experimental results. However, to compare other aspects of performance of ion optics grids as well as the life time of the grid system, one needs to involve charge-exchange ions and a sputtering erosion model.

- **Spacecraft Ion-Thruster Interaction Model Enhancement.** The spacecraft ion-thruster interaction model that can be used to predict spacecraft floating ground potential would be a remarkable improvement. In addition, better Modeling of the thruster plume is recommended. The neutral non-propellant efflux which is generated by sputtering erosion needs to be further investigated. The role of the ion thruster neutralizer needs also to be better understood. The details of the surrounding space environment may need also to be carefully studied.

- **Code Flexibly.** The HG-IFE-PIC Plume code developed in this work use a library of basic geometric shapes to define the complex geometry of a specific spacecraft. These objects are defined in the code using analytic geometric functions. Modeling the more realistic and complicated geometries of real spacecraft require the use of solid Modeling packages. Thus, the HG-IFE-PIC needs to be able to handle this type of object definitions and its implications on defining particle and boundary conditions.
Bibliography


175


Vita

Raed Ismail Kafafy was born on June 17th, 1973 in Cairo, Egypt. He graduated from Cairo University, Egypt in June 1996 with a B.Sc. degree with distinction in Aerospace Engineering. After graduation, he worked as a Research and Teaching Assistant at the Aerospace Engineering Department of Cairo University. During his research work, he contributed to the preliminary analysis and design of the first Egyptian satellite Desert Sat. He also received his M.Sc. degree from Cairo University in Aerospace Engineering with honor in June 2000.

Raed arrived at the United States in August 2000 to enrol the Ph.D. program at the Aerospace and Ocean Engineering Department of Virginia Polytechnic Institute and State University, Blacksburg, Virginia. During his Ph.D. program, he has served as a Graduate Teaching Assistant and a Graduate Research Assistant. As a member of the research team at the Computational Advanced Propulsion Laboratory (CAPLab) of Virginia Tech, he has been conducting research in the areas of Plasma Simulations and Advanced Numerical Methods. This dissertation completes the requirements for his Ph.D. degree in Aerospace Engineering from Virginia Tech.

Raed Ismail Kafafy

Blacksburg, Virginia

in August 2005