Multiscale Modeling of Hall Thrusters

by Lubos Brieda

B.S. in Aerospace Engineering, May 2003, Virginia Tech
M.S. in Aerospace Engineering, June 2005,Virginia Tech

A dissertation submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

May 20, 2012

Dissertation Directed by

Michael Keidar
Associate Professor of Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Lubos Brieda has passed the Final Examination for the degree of Doctor of Philosophy as of March 12, 2012. This is the final and approved form of the dissertation.

Multiscale Modeling of Hall Thrusters

Lubos Brieda

Dissertation Research Committee:

Michael Keidar, Associate Professor of Engineering and Applied Science, Dissertation Director

Chunlei Liang, Assistant Professor of Engineering and Applied Science, Committee Member

Taeyoung Lee, Assistant Professor of Engineering and Applied Science, Committee Member

Justin Koo, Electric Propulsion Modeling and Simulation Lead, Air Force Research Laboratory, Committee Member

Yevgeny Raitses, Principal Research Scientist, Princeton Plasma Physics Laboratory, Committee Member

Acknowledgments

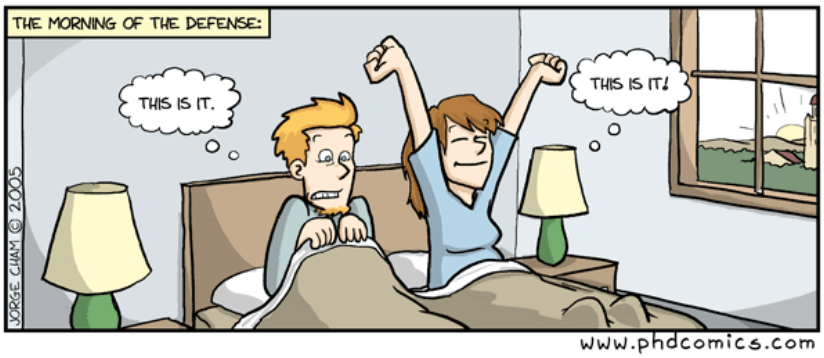
I still remember when back in 2007 I returned to work after spending 16 days on California's John Muir Trail and found that email. In it, Dr. Keidar was telling me that he has just accepted a faculty position at the George Washington University and was wondering if I would be interested in joining his new research group. It was bit of a dilemma at first - I fell in love with the west coast and was also contemplating changing research areas (atmospheric modeling, especially a project allowing me to conduct an experiment from a tent perched in some mountain valley, seemed quite attractive). Yet despite the fact I had to consolidate my life into what fit in my car on its transcoastal journey, and that my new dwelling became a windowless office with that fresh smell of a not-too-recently deceased rat, I am really glad I made the decision to accept the offer. It's been a hard journey, but definitely a good one.

None of this would be possible without Prof. Michael Keidar as my dissertation advisor. Dr. Keidar is truly one of the most brilliant people I know. At the same time, he is also one of the nicest. It's amazing how he never loses his cool, even while supervising a handful of non-related projects, meeting with students, teaching several courses each semester, and also serving as the director of various research institutes. But a similar story could be said about each of my committee members, Prof. Chunlei Liang, Prof. Taeyoung Lee, Dr. Yevgeny Raitses, and Dr. Justin Koo. I am really grateful for their feedback and comments on my dissertation work. In addition, several professors made my GWU experience unique. First, there is Professor Emeritus Michael Myers. Without realizing, my first semester I signed

up for a true class from hell: Dr. Myers' Fluid Mechanics. This was one of those classes where your entire life revolves around completing the homework, and the time between assignments simply does not seem to have enough hours to get everything done. Yet while really painful, I did learn a lot. It was this class that after all my years of schooling finally gave me the required understanding of the governing relations that form the basis of all fluid dynamics research. Next, I want to thank Dr. Jonathan Silver. His course on biological nanomachines, although not related to my actual research, was quite fascinating, and definitely helped broaden my horizons. I really appreciated his rigor analyzing, and often discovering mistakes in respected peer-reviewed publications. I also need to thank Prof. Michael Plesniak, the MAE department head, for allowing me to serve as the department webmaster. This to me was a much more rewarding way to make few extra bucks than grading homeworks.

Next, there is the MpNL and Staughton Hall crew, especially Madhu Kundrapu, Brent Duffy, and Jarrod Fenstermacher. I really enjoyed our various trips to the "Clam" and our discussions on a wide range of topics. Hopefully I did not bore you guys too much with my various business ventures and other crazy ideas. I am excited that we all managed to graduate the same year - I am looking forward to our last Clam trip after the hooding ceremony. Then there is Sandra, previously my girlfriend, now my fianceé, and soon to be my wife. Thank you baby for putting up with my constant "I have to work", late nights, and weekends that look no different than a regular work day. I bet you didn't know what you were signing up for when you decided to date a grad student! I think Jorge Cham from phdcomics.com summarized it the best in his cartoon:[1]

[1] `http://www.phdcomics.com/comics/archive.php?comicid=570`



Finally, I take this opportunity to gratefully acknowledge the financial support of the Air Force Office of Scientic Research, the Air Force Research Laboratory, The George Washington University Fellowship, and the D.C. Grant Program. It would be very difficult to complete my dissertation work without this financial assistance.

Abstract

Multiscale Modeling of Hall Thrusters

Hall thrusters are efficient space propulsion devices that produce thrust by accelerating quasineutral plasma. The thruster consists of a channel with one end open to the ambient environment. The neutral propellant is injected at the closed end and is ionized by electrons produced by an externally mounted cathode. A radial magnetic field is applied across the channel to restrict the electron flow and increase the ionization efficiency. The magnetic field also become lines of constant potential and induce an electric field that accelerates the ions out of the device.

This thesis introduces a novel approach for modeling these devices. Hall thrusters have been operated in space for over 40 years, yet no code yet exists that can self-consistently predict their operation and lifetime. This shortcoming is to a large extent driven by the presence of significantly different spatial scales that dominate the internal plasma dynamics. One such a scale is the microscopic scale of an electron orbiting about a magnetic field line. This scale determines the rate with which electrons diffuse across the field line which becomes an important parameter in modeling the internal thruster discharge. It is not numerically feasible to include electron motion when dimensions of a thruster are considered. As such, Hall thruster codes typically utilize a hybrid approach in which electrons are modeled as a fluid with electron transport given by an analytical expression. These expressions do not correctly capture the experimentally-observed behavior. In this work, a kinetic code was developed to de-

termine transport at the desired spatial locations self-consistently by considering the rate with which electrons diffuse across a magnetic field line. This kinetically- determined transport can then be utilized in the discharge code instead of the analytical model. This code is utilized in this work to study fundamentals of electron transport. Among other findings, simulation results indicate that the linear combination of classical and wall induced transport is not valid due to synergistic effects.

Thruster codes also do not accurately capture the near-wall region known as the sheath which drives ion flux to the walls. In addition, often the plume produced by these devices needs to to be analyzed in order to quantify any possible spacecraft contamination events. On the scale of the spacecraft, the thruster internal details are no longer resolved and the thruster acts as a source of ions and neutrals into the plume analysis. These two topics describe the other two spatial scales studied in this work. The near-wall region is analyzed with a particle code which resolves the near-wall non-neutral region and also estimates wall erosion rates based on ion impacts. This code also takes into account magnetic field line inclination. At highly-inclined magnetic field line angles, the code predicts that ions become repelled from the wall, resulting in a sheath collapse. Finally, a novel contribution to the plume modeling effort is a time-dependent source that injects particles based on a time-resolved kinetic sampling of particles leaving the thruster simulation. Hall thrusters are inherently unsteady devices, however, typical plume codes assume steady operation. Preliminary results indicate that the unsteady operation of these devices is demonstrated in a corresponding oscillating current of backflowing ions.

Table of Contents

Acknowledgments iv

Abstract vii

List of Figures xii

List of Tables xvi

Nomenclature xvii

Chapter

1. Introduction 1

1.1 Overview of Hall Thrusters 1
1.2 Novel Thruster Designs 3
1.2.1 Hall thruster analysis 6

2. Hall Thruster Theory and Modeling 9

2.1 Electron Mobility 9
2.1.1 Classical Mobility 10
2.1.2 Near-Wall Conductivity 13
2.1.3 Azimuthal Oscillations 14
2.2 Hall Thruster Modeling 15
2.2.1 Fluid Modeling 16
2.2.2 Kinetic Modeling 16
2.2.3 Hybrid Modeling 18
2.3 Thermalized Potential Model 18
2.4 Plume Modeling 20
2.5 Multiscale Approach to Modeling Hall Thrusters 21

3. Kinetic Modeling of Electron Transport 27

3.1 Overview . . . 27
3.2 Computational Domain . . . 29
3.3 Potential Solver . . . 31
3.3.1 Boundary Conditions . . . 33
3.4 Particles . . . 35
3.4.1 Particle Loading . . . 35
3.4.2 Loading a Prescribed VDF in Magnetized Plasmas . 39
3.4.3 Particle Motion . . . 42
3.4.4 Wall Model . . . 45
3.4.5 Particle Boundaries . . . 47
3.4.6 Current Collection . . . 48
3.5 Collisions . . . 49
3.5.1 Momentum Transfer Collisions . . . 50
3.5.2 Ionization and Excitation Collisions . . . 52
3.5.3 Coulomb Collisions . . . 53
3.6 Transport Calculation . . . 55
3.7 Implementation . . . 57

4. Code Validation and Mobility Studies . . . 61

4.1 Introduction . . . 61
4.2 Unit Tests . . . 61
4.2.1 Single Particle Motion . . . 61
4.2.2 Wall Interactions . . . 67
4.2.3 Particle Loading . . . 69
4.2.4 Potential Solver . . . 71
4.2.5 Collisions . . . 71
4.2.6 Random Number Generator . . . 74
4.2.7 Transport Calculation . . . 75
4.3 Parametric Studies . . . 79
4.3.1 Velocity Distribution Function . . . 80
4.3.2 Secondary Electrons Under a Magnetic Mirror . . . 81
4.3.3 Dielectric Walls . . . 84
4.3.4 Azimuthal Oscillations . . . 85
4.3.5 Synergistic Wall Effects . . . 86

5. Studies of the Princeton Cylindrical Hall Thruster . . . 91

5.1 Overview of the Thruster . . . 92
5.2 HPHall Modeling . . . 93
5.2.1 Inability of HPHall to resolve the CHT Magnetic Field Topology . . . 96
5.2.2 Initial Results with Classical Mobility . . . 98
5.2.3 Kinetic Code Inputs . . . 98
5.3 Kinetic Transport Calculation . . . 100

5.3.1 Transport Analysis 100
5.3.2 Results with Self-Consistent Mobility 105
5.4 Particle Sampling 106

6. Plasma Wall Transition in the Presence of an Inclined Magnetic Field 109

6.1 Introduction 109
6.2 Computational Model 110
6.2.1 Simulation Domain 110
6.2.2 Particle Injection 112
6.2.3 Potential Solver 113
6.2.4 Boundary Conditions 115
6.2.5 Quasineutral Solver 116
6.2.6 Implementation 116
6.3 Results 117
6.3.1 Potential Distribution at Uniform Density 117
6.3.2 Sheath variation with magnetic field angle 119
6.3.3 Influence of Secondary Electron Emission 121
6.3.4 Magnetic Mirror Effect 122
6.4 Discussion 124
6.4.1 Sheath Collapse 124
6.4.2 Erosion and Lifetime 126
6.4.3 Sheath Stability 128

7. Plume Modeling 130

7.1 Introduction 130
7.2 Draco Plume Simulation Code 132
7.2.1 Mesh Splitting 132
7.2.2 Mesh Intersection 137
7.2.3 Time-Dependent Source Model 142
7.2.4 Quasineutral Potential Fix 147
7.3 Results 149

8. Conclusion and Future Work 153

8.1 Conclusions 153
8.2 Future Work 155
8.3 Source Code 156

List of Figures

1.1 Hall thruster schematic 2

1.2 Illustrative photos of an (a) annular and a (b) cylindrical Hall thruster from the Princeton Plasma Physics Laboratory. Images courtesy of [1] 4

1.3 Plumes of 100W annular and cylindrical Hall thruster 5

2.1 Collisional electron drift 12

2.2 Wall reflection of electrons 14

2.3 Graphical representation of the multiscale approach. 22

2.4 Schematic of the multiscale approach 25

3.1 Lynx computational domain 30

3.2 Electric field at the wall interface 33

3.3 Electron and ion densities, and charge density for $Q_e = 0.9Q_i$. . . . 36

3.4 The effect of particle resampling on the tail of the velocity distribution function 37

3.5 Electron count and wall current vs. iteration time step 38

3.6 The perpendicular, parallel, and combined velocity distribution functions 39

3.7 Trace of a magnetized particle 40

3.8 Wall yield 46

3.9 Collision cross-section vs. energy 50

3.10 Electron velocity thermalization 54

3.11 Schematic of guiding center shift due to scattering events. 55

3.12 Plot of typical simulation inputs 57
4.1 Larmor-radius unit test results 62
4.2 Particle trace with simple and improved loading in the presence of $E_\perp$ 64
4.3 Particle trace for a particle in a potential well 65
4.4 Electron reflecting in a linearly strengthening magnetic field 67
4.5 Trace of a particle bouncing between two completely elastic walls . 68
4.6 Velocity distribution functions obtained with different number of computational particles 70
4.7 Close up of the high energy tail 70
4.8 Prescribed charge densities and the computed potential for solver test 72
4.9 Velocity Distribution Function Evolution 73
4.10 Comparison of system kinetic energies 73
4.11 Random number histogram comparison 75
4.12 Influence of orbit check on transport calculations 76
4.13 Transport properties along a field line 77
4.14 Simulation setup for mobility studies 80
4.15 Typical plasma parameters at steady state 81
4.16 Modification of velocity distribution function by secondary electrons 82
4.17 Near wall transport due a magnetic pressure 82
4.18 Near wall transport of secondary electrons under a strong mirror . . 83
4.19 Time evolution in collected wall charge for a fixed and a floating wall 84
4.20 Plasma potential and electron density for a floating dielectric (solid) and a fixed Dirichlet (dashed) wall 85
4.21 Guiding center drift to an azimuthal electric field 86
4.22 Mobilities due to collisions and/or wall effects 89
4.23 Comparison of velocity distribution functions for the three considered cases for initial electron temperatures 10 and 35 eV. 90

5.1 HPHall simulation domain 94
5.2 Neutral injection 94
5.3 The desired (right) and the used (left) cathode λ line 97
5.4 HPHall simulation results obtained using teh analytical mobility model 99
5.5 Inputs to the kinetic code 101
5.6 Additional parameters interpolated onto the magnetic field mesh . . 102
5.7 Simulation results from kinetic Lynx simulations, interpolated onto the HPHall mesh 102
5.8 Magnetic field lines plotted in Figure 5.9 103
5.9 Kinetic code simulation results 104
5.10 Temporal evolution in average current density for the four selected field lines 105
5.11 Centerline plasma parameters 106
5.12 Particle Sampling 108
6.1 Schematic of the computational domain for near-wall analysis . . . 111
6.2 Potential with uniform plasma density 118
6.3 Ion density profiles for three different magnetic field line angles . . . 120
6.4 Potential profile for 40° magnetic field line inclination 122
6.5 Ion density contours in the presence of a magnetic mirror 123
6.6 Variation in the maximum sheath thickness with the angle of magnetic field 125
6.7 Plots of normalized radial velocity and ion velocity streamlines at $\theta = 85°$. 125
6.8 Comparison of wall flux and emitted sputtered yield for 5 selected configurations. 127
6.9 Sheath profile for $\theta = 60°$ as a function of normal electric field. . . . 128
7.1 Profiling studies from previous version of Draco 134

7.2 Potential solution obtained for a single and multiple (16) zones . . . 136
7.3 Thin feature resulting in multiple cell/edge cuts 139
7.4 Typical problem arising in earlier version of Volcar resulting in non-physical spikes 140
7.5 Node control volume used to classify node location 142
7.6 Discharge current from a typical HPHall simulation of the CHT showing a strong temporal dependence 143
7.7 Particle sampling using a time-dependent source 144
7.8 Comparison of QN and QN-Switch potential solutions 148
7.9 Simulation results showing the surface deposition rate, normalized by thruster mass flow rate, and the plume density in m^{-3}. 150
7.10 Comparison between the simulation particle count at steady state and the instrument collected current. 150

List of Tables

2.1 Relevant properties of a representative Hall thruster 10

4.1 Summary of errors for a single particle motion 63

4.2 Errors in collision pair selection algorithm 74

4.3 Sigmas and timing results for 3 random number generators 75

4.4 Mobility without orbit check 77

4.5 Mobility with orbit check 77

4.6 Mobility variation with σ, without orbit check, 5000 time steps, and $E_\perp = 20$ kV/m. 79

4.7 Mobility variation with $E_\perp$ and $\sigma = 10^{-17}$ m^2. 79

5.1 Summary of HPHall simulation inputs 98

Nomenclature

B	magnetic field (T)
E	electric field (V/m)
f	distribution function (number)
it	time step (iteration number)
I	current (A)
j	charge density (A/m^2)
k	Boltzmann constant, $1.3806\ldots \times 10^{-23}\,m^2\,kg\,s^{-2}\,K^{-1}$
m	mass (kg)
M	macroparticle weight (number)
n	number density ($\#/m^3$)
N	particle count (number)
p	pressure (Pa)
Q	charge (C)
r	radius (m)
R	random number, [0:1)
Δt	time step (s)
T	temperature (K)
v	velocity (m/s)
v_{th}	thermal velocity (m/s)
W	kinetic energy (J or eV)
x	position (m)
Z	average ionization of ions, $Z = 1$ assumed

Subscripts

e	electrons
gc	guiding center
i	ions
a	atoms
$\perp$	direction perpendicular to the magnetic field line
$\parallel$	direction parallel to the magnetic field line

Greek Symbols

ϵ_0	permittivity of free space, $8.854\ldots \times 10^{-12}\,\mathrm{F/m}$
ϕ	potential (V)
ρ	charge density ($\mathrm{C/m^3}$)
μ	mobility (1/T)
ν	collision frequency (#/s)
ω_c	cyclotron frequency (Hz)
σ_c	conductivity (S/m)
σ	collision cross-section ($\mathrm{m^2}$)

Abbreviations

AFRL	Air Force Research Laboratory
CHT	Cylindrical Hall Thruster
DSMC	Direct Simulation Monte Carlo
EP	Electric Propulsion
ES	Electrostatic
HET	Hall Effect Thruster
MCC	Monte Carlo Collisions
PIC	Particle In Cell
PPPL	Princeton Plasma Physics Laboratory

Chapter 1

Introduction

1.1 Overview of Hall Thrusters

Hall thrusters are efficient space propulsion devices that utilize an applied magnetic field to accelerate quasineutral plasma. The typical Hall thruster consists of an azimuthally-symmetric channel having one end open to the ambient environment. The upstream end is capped by the anode and the downstream end is open to the ambient environment. Figure 1.1 shows a cross-section of a particular thruster design. This thruster consists of an annular upstream region and a cylindrical downstream part. The neutral propellant is injected into the channel at the upstream end, typically via ports in the anode. Xenon is used as the standard propellant for space applications due to it's high mass and non-reacting properties. A cathode mounted outside the thruster produces electrons. Fraction of these electrons enter the channel and ignite the discharge via electron-neutral ionization collisions. In order to increase the electron transit time, and hence increase the ionization efficiency, a radial magnetic field is applied across the channel. Several representative field lines are sketched with red lines and magnetic coils are also indicated. The strength of the magnetic field is selected such that electrons become magnetized - trapped in helical orbits about the field lines. This particular trapping is not shown in th image. Ions, on the other hand, are not magnetized and freely cross the magnetic field. The magnetic

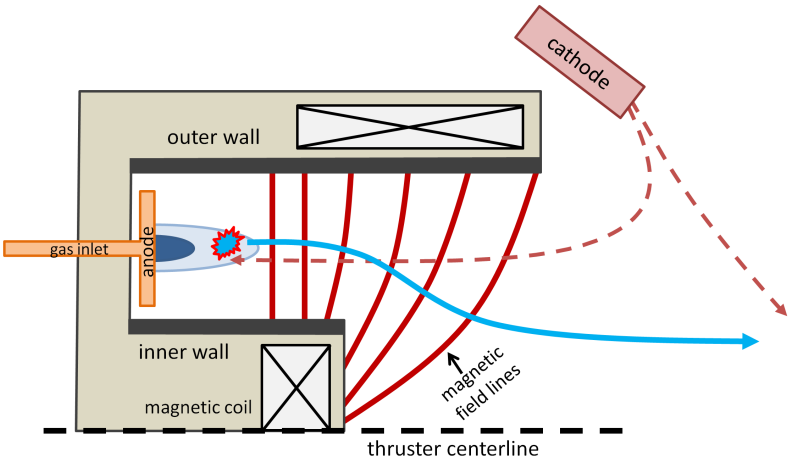


Figure 1.1: Cross-section of a cylindrical Hall thruster. Ionized propellant is accelerated by an electric field that develops in direction perpendicular to the applied magnetic field lines.

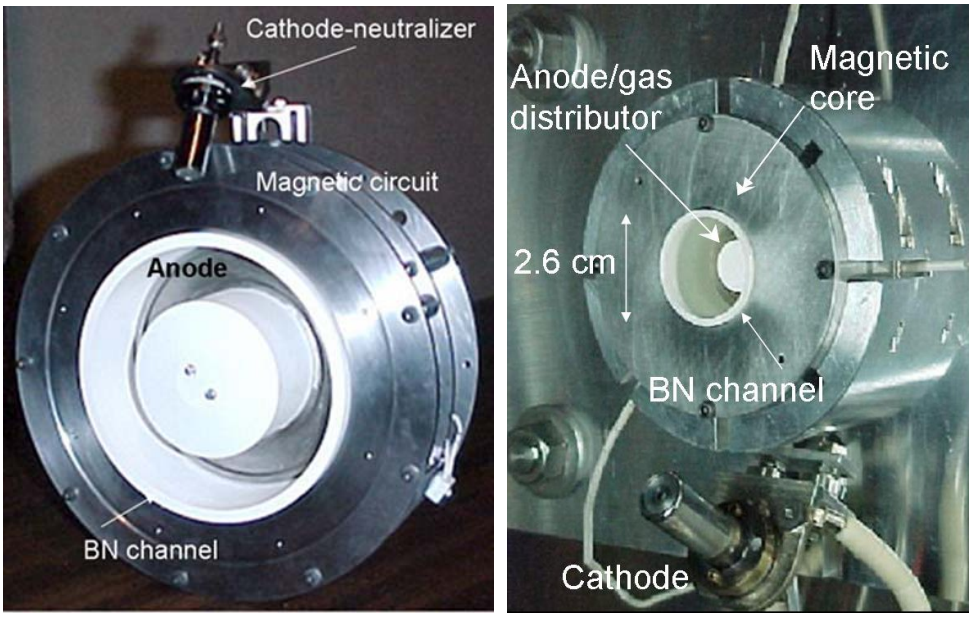
field thus serves another important role. The motion of electrons is unrestricted in the direction tangential to the magnetic field and thus electrons are free to redistribute in the tangential direction according to the spatial variation in n_i, the ion density. The effect is the neutralization of space charge, $\rho = e(Zn_i - n_e) \sim 0$. Here e is the elementary charge, Z is the effective ion charge, assumed to be 1 in this work, and n_e is the electron number density. The magnetic field lines thus become lines of constant potential ϕ. An electric field $\vec{E} = -\nabla\phi$ then develops in the ∇B direction. Since the magnetic field is predominantly radial, the electric field will consist of a large axial component. It is this field that accelerates ions out of the device. The electric field also induces an azimuthal $\vec{E} \times \vec{B}$, or Hall, drift of electrons about the thruster centerline. The final picture of electrons in a Hall thruster thus consists of electrons orbiting about a field line and oscillating between the walls, while at the same time drifting about the thruster centerline. Due to this drift, Hall thrusters are also commonly known as closed-drift thrusters.

Historically, Hall thrusters have utilized an annular design which naturally leads itself to devices with a radial magnetic field. An example of such a thruster is shown in Figure 1.2a. This figure shows a 3 kW thruster from the Princeton Plasma Physics

Laboratory [1]. The outer diameter of this thruster is 13 cm. The thrust scales with input power and mass flow rate, but typical Hall thruster thrust levels are in the 50-200 mN range. The hollow cathode responsible for producing electrons can be seen above the thruster. Another prominent feature is the dielectric Boron Nitrite wall lining the discharge channel. The presence of the dielectric, as well as the relatively long discharge channel, places this thruster in a class known as a "thruster with a magnetic layer and an extended acceleration zone", however generally referred to as Stationary Plasma Thruster (SPT) for the obvious brevity reasons. The dielectric walls play an important role in the operation of this device. Electron impacts with the wall result in emission of low-energy secondary electrons. These cold electrons cool the primary electron population, and keep the electron temperature low [2]. This in turn results in a more gradual ion acceleration than what is achieved with the alternate design, the "thruster with an anode layer" or TAL. TAL thrusters use conductive walls and short channels. In this work only the SPT-type thrusters are analyzed, for the reasons that SPT thrusters enjoy a much wider usage in the industry and research community.

1.2 Novel Thruster Designs

Figure 1.2b shows a novel hybrid design, the 100 W Cylindrical Hall Thruster (CHT) from PPPL [3]. The outer channel diemeter is 2.6 cm. The schematic in Figure 1.1 was based on this thruster. The theory described in the opening paragraph provided a simplified insight into the operation of Hall thrusters and neglected the presence of walls. However, walls play an important role in these devices and their role becomes more prominent as thruster designs are miniaturized. Imagine that a wall is placed instantaneously into an unperturbed plasma. Even if we assume that plasma species are in a thermal equilibrium, $kT_e = kT_i$, the thermal velocity of electrons $v_{th,e} = \sqrt{2kT_e/m_e}$ will exceed the ion velocity by several orders of magnitude due

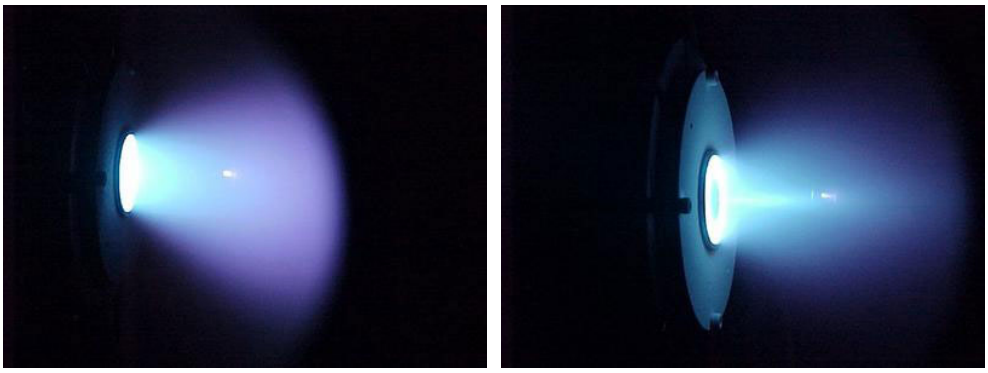


(a) Annular 3kW HET (b) Cylindrical 100W HET

Figure 1.2: Illustrative photos of an (a) annular and a (b) cylindrical Hall thruster from the Princeton Plasma Physics Laboratory. Images courtesy of [1]

to the large mass ratio, $\sqrt{m_i/m_e} \sim 489$ for Xenon ions. This results in initially a large flux of electrons to the wall, $\Gamma_e \gg \Gamma_i$. In order to restore equilibrium, the potential in the undisturbed plasma sufficiently removed from the wall increases to make the wall appear negatively charged in respect to the bulk plasma. The negative potential reduces the electron flux while at the same increasing the ion flux due to the electrostatic $\vec{F} = q\vec{E} = -q\nabla\phi$ Lorentz attraction. The potential increases until the net wall current disappears, $j_{wall} = e(\Gamma_i - \Gamma_e + \Gamma_s) = 0$. Here the three fluxes on the right side correspond to incident ions, incident electrons, and emitted secondary electrons.

The potential drop between the plasma and the wall occurs in a non-neutral region known as the sheath. In some conditions, the sheath thickness can reach dimensions comparable to the width of the channel [4]. Ions located in the sheath will be accelerated by the potential drop and subsequently lost to the wall neutralization. The loss of ions to walls is one of the main inefficiencies of Hall thrusters. This



(a) Cylindrical (b) Annular

Figure 1.3: Plume of a 100W Hall Thruster with a (a) cylindrical channel and (b) annular channel. Images from [1].

issue becomes especially problematic as the size of the thruster is decreased and the relative ratio between the bulk and the sheath populations decreases. Ion wall impacts also cause sputtering (bombardment erosion) of the wall material. Erosion of thruster walls is the primary factor limiting the lifetime of these devices. For this reason, several novel Hall thrusters, such as the one shown here, have started experimenting with non-traditional geometries. This particular thruster utilizes a hybrid channel in which the upstream ionization section is annular, however the downstream acceleration zone is completely cylindrical. This The lack of the inner wall in the acceleration region reduces wall losses. It also introduces new interesting physics due to the non-radial structure of the magnetic field. The convergence of magnetic field lines near the center pole results in a magnetic pressure which is not present if the simplified radial configuration is assumed. This thruster is studied in more detail in Chapter 5 of this thesis. It can be seen operating in Figure 1.2b. Figure 1.3b shows the plume of the same thruster, however modified such that the annular zone extends the entire length of the thruster to give it the standard annular geometry (this thruster can also be operated in a fully cylindrical configuration). As can be seen from this figure, while the hybrid geometry reduces wall losses, it also increases the plume divergence, hinting at a lower thrust efficiency.

1.2.1 Hall thruster analysis

Hall thrusters are sometimes also referred to as *gridless ion thrusters*. Unlike their counterparts, the gridded ion thrusters, Hall thrusters do not contain a physical electrode responsible for accelerating the ionized propellant. The lack of ion optics makes these devices more robust and less-susceptible to the harsh launch and space environment. In addition, since Hall thrusters accelerate quasineutral plasma, they are not subject to the Child-Langmuir law which limits the current density in ion thrusters. This allows Hall thrusters to generate higher current densities, allowing for a smaller required thruster cross-section for a desired thrust level. The omission of grids and the second cathode used in ion thrusters further reduces the mass of these devices.

Unfortunately, the features that make Hall thruster attractive for space propulsion also make them difficult to analyze. The acceleration in ion thrusters can be studied with a relative ease if the geometry of the ion grids is known. Yet despite over 40 years of flight heritage - the first Hall thruster, SPT-50, was launched aboard the Soviet Meteor in December 1971 - no codes yet exist that can predictively model the plasma dynamics inside these devices. One of the primary difficulties involves the transport of electrons across the magnetic field. As will be discussed in the following chapter, the rate at which electrons diffuse across magnetic field lines plays a crucial role in establishing the internal potential profile. According to the classical theory, magnetized electrons scatter across magnetic field lines by undergoing collisions. Unfortunately, collisions alone do not reproduce the anode currents observed experimentally. This additional cross-field transport is known as *anomalous diffusion* and a quick literature search will identify tens of papers published in recent years attempting to tackle its origin. A brief, but far from an exhausting introduction to the subject are the works cited in [5–20].

In the absence of an analytical model capable of capturing the mobility variation,

Hall thruster simulation codes rely on user-defined coefficients controlling the strength of various anomalous terms. The magnitude of these coefficients varies with the thruster and the operating condition, and is generally not known *a priori*. The typical Hall thruster simulation sequence consists of performing parametric studies to find the set of input conditions that offer the best agreement with experiments. Such an approach clearly does not lend itself to a truly predictive code.

At the heart of the problem is the fact that electron transport is fundamentally a kinetic phenomenon. However, for computational reasons, typical Hall thruster codes utilize a fluid model for electrons. It simply is not practical to resolve the electron time and spatial scales on the scales of the thruster. Although fully-kinetic Hall thruster codes have been developed [21, 22], these typically require weeks if not months of supercomputer time and non-physical mass ratios. Furthermore, analysis of the thruster discharge channel is not the only area in need of a modeling analysis. An important engineering issue that arises during integration of electric propulsion (EP) on spacecraft is assuring that the thruster plume does not negatively impact the operation and lifetime of the spacecraft and instruments. Plume models operate on the spatial scale of the satellite. It is no longer necessary or practical to model the internal thruster physics on such a scale. Instead the thruster can be assumed to be an injection source into a larger plume simulation. For the plume analysis, the unknowns of interest are the fluxes and velocities of ions as they leave the thruster.

This thesis describes a novel approach for modeling Hall thrusters, in which the thruster back-to-back problem is subdivided into several individual spatial scales. Each spatial scale is treated with a numerical method most appropriate to compute the desired properties. The different spatial scales are coupled via shared parameters. The ultimate goal of this methodology is to develop a simulation code capable of predictively modeling Hall thrusters and their spacecraft interaction using a common desktop workstation in the relatively rapid timeframe of hours to days. The approach

is described in more detail in the next chapter. The next chapter also introduces the fundamental physics of Hall thrusters and summarizes common approaches used in Hall thruster modeling. The thesis then continues with chapters devoted to each of the different spatial scales. The kinetic code developed to study the electron transport along a single magnetic field line is described in Chapter 3. Chapter 4 presents validation studies and also presents results from transport studies performed using a simplified geometry. In Chapter 5, analysis is performed on the Princeton 2.6 cm Cylindrical Hall Thruster (CHT). A standard Hall thruster code HPHall is used to obtain the global discharge parameters, which are subsequently used to compute mobility with the kinetic code. The self-consistent mobility is then used to update the initial HPHall results. Chapter 6 introduces a code designed to study the near-wall ion dynamics in thrusters with an inclined magnetic field topology, such as the CHT. Finally, Chapter 7 tackles plume modeling and investigates the ambient environment around a particle sensor on a hypothetical satellite flying the CHT. The thesis is concluded with a brief summary and topics for future work.

Chapter 2

Hall Thruster Theory and Modeling

2.1 Electron Mobility

An important characteristic of Hall thrusters is the magnetic confinement of electrons. To illustrate this, let's consider a fairly typical Hall thruster such as the 1kW laboratory Hall thruster from the Princeton Plasma Physics Laboratory analyzed in [23]. Some relevant properties are summarized in Table 2.1. This thruster has an outside diameter of 12.3 cm and a channel width of 2.5 cm. The channel is 4.3 cm long. The magnetic field has a peak value of 113 Gauss (0.0113 Tesla) at the exit plane. The electron temperature and the electric field also reach their peak value at or near the exit plane. First, let's consider the Larmor radius r_L, the size of the orbit a charged particle makes about a magnetic field line. In order for a particle to be considered magnetized, the orbit must be smaller than some characteristic dimension of the problem, $r_L \ll l$. The Larmor radius is given by

$$r_L \equiv \frac{v_\perp}{\omega_c} = \frac{m}{|q|B} v_\perp \tag{2.1}$$

where $v_\perp$ is the tangential velocity about the orbit centerline. The angular rate of rotation ω_C is known as cyclotron frequency. We can estimate the electron $v_\perp$ from the thermal energy, $v_\perp \sim v_{th} \equiv \sqrt{2kT_e/m_e} = 3.2 \times 10^6$ m/s. The cyclotron frequency

Table 2.1: Relevant properties of the 2kW annular PPPL thruster. The distances are measured from the exit plane. The anode is located at -4.3 cm.

Parameter	Value
d_{outer}	12.3 cm
d_{inner}	7.3 cm
B_r (0 cm)	113 Gauss
B_r (-2 cm)	25 Gauss
E_z (0 cm)	15 km/s
T_e (0 cm)	30 eV
I_d	1.6 A
V_D	250 V

$\omega_c = |q|B/m \sim 2 \times 10^9$ Hz, yielding $r_{L,e} = 0.16$ cm, or 3% of the channel length. On the other hand, ion velocities can be estimated from energy conservation. An interesting feature of many Hall thrusters is that a significant portion of acceleration occurs outside the thruster. For instance, in this device, measurements indicate that $\Delta\phi = 100$ V inside the thruster, and $\Delta\phi = 150$ V outside. Hence at the exit plane, the ion velocity is $v_i = \sqrt{2e\Delta\phi/m_{Xe+}} \sim 12.1$ km/s, with the remaining acceleration occurring in the near plume. The ion cyclotron frequency is 8304 Hz. This gives us $r_{L,i} = 1.46$ m, or $\sim 3000\%$ of the thruster length. Since $r_{L,e} \ll l$ and $r_{L,i} \gg l$, we can see that indeed electrons are magnetized while ions are not. It should be noted that thruster length was used as the characteristic length instead of the commonly used diameter since the motion in the axial direction is of importance when considering electron transport.

2.1.1 Classical Mobility

Current conservation requires electrons entering the channel from the cathode to eventually reach the anode. The discharge current measured in the experiment was 1.6 A. At the thruster exit plane, this current consists of an electron and ion component. Assuming uniform current density, we have $j \equiv env = I/A = 208$ C/m^{2}s. As analyzed by Koo [15], correctly computing this current requires taking

into account the size of the ionized region. Such an approach is not used here, instead the ion plume is assumed to uniformly span the entire channel. Unfortunately, ion density was not reported directly. Hall thruster plasma densities are generally of the order of 10^{17} m^{-3} and using this general value and 12.1 km/s ion velocity, the electron current density j_e becomes 400 C/m^2. From Ohm's law, we have $\vec{j} = \sigma_c \vec{E}$, where σ_c is the plasma conductivity. An additional term appears in the equation if pressure gradients are present, these are ignored here for simplicity. Another form of the equation can be obtained by dividing through by charge density, $\vec{v} = (\sigma_c/en)\vec{E}$ or

$$\vec{v} = \mu \vec{E} \tag{2.2}$$

This very important relationship relates the macroscopic velocity of electrons $\vec{v}$ to the electric field $\vec{E}$ via a proportionality factor μ. This factor is called mobility. In the case of a magnetized plasma, μ is a tensor. The motion can be decoupled into two modes, the motion along and perpendicular to the field line,

$$v_\parallel = \mu_\parallel E_\parallel \tag{2.3}$$

$$v_\perp = \mu_\perp E_\perp \tag{2.4}$$

In the perpendicular direction, $v_\perp$ is the speed with which electron guiding centers move across the magnetic field. The value of $\mu_\perp$ can be estimated if the plasma density is known. With $n_i = 10^{17}$ m^{-3}, the drift velocity is $v_\perp = 25$ km/s, and the value of mobility becomes $\mu_\perp \sim 1.7$.

This value can next be compared with the analytical estimate. In the absence of any scattering events, electrons will continuously gyrate about a magnetic field line and $\mu_\perp = 0$. The classical theory of electron transport is based on diffusion via a random walk due to particle collisions. The step size of the diffusion process is r_L,

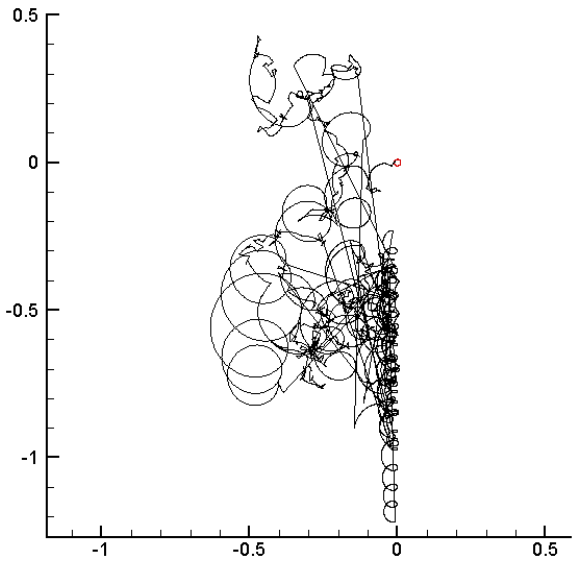


Figure 2.1: Simulation of an electron diffusing across a magnetic field line due to collisions. The small red circle shows the initial orbit. The electric field goes from left to right.

the Larmor radius. The cross-field mobility is given

$$\mu_\perp = \frac{\mu_\parallel}{1 + \omega_c^2 \tau^2} \tag{2.5}$$

where $\mu_\parallel$ is the mobility in the absence of a magnetic field (or in the direction along the field). The parallel mobility is completely collision-driven and is given by

$$\mu \equiv \frac{|q|}{m\nu} \tag{2.6}$$

where ν is the collision frequency. It is given by $\nu = n_n \overline{\sigma v}$, where $\overline{\sigma}$ is the average collision cross-section, $\overline{v}$ is the average relative velocity, and $n_n = 10^{19}$ m^{-3} is the neutral density. Again using typical values, we have $\sigma = 10^{-19}$ m^2. Then by assuming neutrals to be stationary, the collision frequency for electron-neutral collisions becomes $\nu = 3.2 \times 10^6$. Then $\mu = 54,963$ T^{-1} and $\mu_\perp = 0.14$ T^{-1}. This rather simplified analysis results in the analytical model underpredicting the experimental measurements by approximately one order of magnitude. Such a finding is typical.

In addition, as indicated by numerous experimental studies, the discrepancy between the expected and measured mobilities is a non-linear function of position. While in some parts of the discharge, the analytical model agrees well with theory, in other parts it may underpredict the actual values by one or several orders of magnitude. For this reason, the EP research community has spent a considerable effort to investigate possibly sources of this anomalous mobility, and to derive analytical expressions that can be used to augment the classical mobility models.

2.1.2 Near-Wall Conductivity

A simple concept explaining the extra mobility was put forth by Morozov [8]. The walls of the discharge chamber play an important role in Hall thrusters. The general idea of near wall conductivity is that the extra mobility arises from electron wall impacts. This is outlined schematically in Figure 2.2. This picture shows the closeup of a rough thruster wall and several near-wall potential contour lines. A low energy particle without energy sufficient to overcome the potential drop will be reflected. As shown here, depending on the surface details, the sheath may exhibit non-planar topography and even particles that do not come into actual contact with the wall may be scattered. Particles impacting the wall may be absorbed or reflected back. In addition the electron impact can induce emission of secondary electrons from the wall. These interactions, with the exception of a mirror-like elastic reflection from a smooth surface will result in the particle shifting onto a new magnetic field line. This resulting shift in guiding centers then contributes to transport. Another important characteristic of NWC is that, assuming a symmetric and steady sheath, electrons born at one wall with just a small finite velocity will enter the opposite wall with the same velocity. Depending on the actual energy of these particles, this back-and-forth beating of electrons can result in a significant population of secondary electrons. Eventually, the sheath will become saturated by the additional electron population,

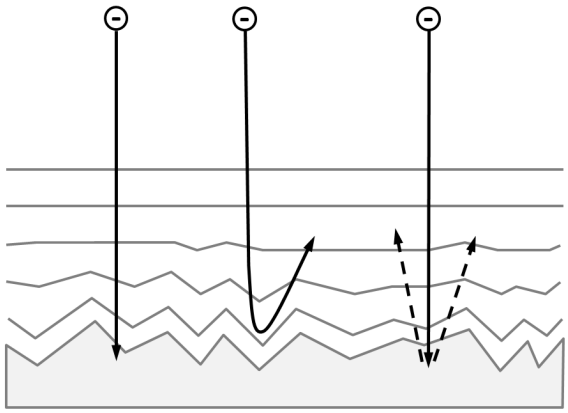


Figure 2.2: Electrons impacting the wall are either reflected back into the domain, are absorbed by the wall, or result in emission of secondary electrons.

resulting in a non-monotonic sheath. Such a sheath can then trap electrons emitted from the wall in a small near wall region and lead to a significant electron current due to electrons bouncing between the wall and the inflected sheath. It is this near wall trapping from which this method derives its name. In this work, a more general treatment of NWC is considered. NWC is taken to be a contribution to transport due to the presence of walls regardless of whether the actual mobility occurs in the vicinity of the walls. In addition, in the present work, non-laminar sheath is not included, and only electrons actually impacting a wall are considered.

2.1.3 Azimuthal Oscillations

A competing model to explain the anomalous mobility is plasma turbulence and azimuthal plasma waves. As analyzed by Choueiri [24], Hall thrusters exhibit a wide range of oscillations. Azimuthal oscillations were analyzed by Darnon [25], among others. The azimuthal oscillations are of interest since the mere presence of an electric field in the azimuthal direction will induce current in the axial direction due to the $\vec{E} \times \vec{B}$ drift. Such an electric field could be associated with the so-called spoke mode noted by several researchers [19, 26, 27]. This mode is believed to correspond to an ionization wave propagating in the azimuthal direction and generating a region of

high plasma density.

However, even in the absence of steady electric fields, a non-steady $\partial E_\theta/\partial t$ field will induce transport. This effect is somewhat analogous to the presence of a constant field, but is due to particles spending asymmetric time in the high and slow velocity regions [28]. This effect scales with the E/B drift velocity. Experimental measurements of transport also indicate a $1/B$ dependence. This type of "anomalous" transport is known as Bohm mobility. Since it is not clear which of these models dominates, mobility in Hall thrusters is typically modeled as

$$\mu = \mu_C + \alpha_1 \mu_w + \alpha_2 \mu_B \tag{2.7}$$

where the terms on the right hand side correspond to the classical, wall, and Bohm-type terms and the α's are user defined parameters that must be determined by parametric studies.

2.2 Hall Thruster Modeling

Despite Hall thrusters having over 40 years of flight history, the computational modeling is still in early years of maturity. Partial responsibility lies in the lack of research into Hall thrusters in the western world during the early years of development. Hall thrusters were initially studied in 1960's in both the US and the Soviet Union. However, these efforts soon split, with USSR taking an active role in developing Hall thrusters (primarily via experimental techniques), while the United States concentrated on gridded ion thrusters. It was only in the 90's and 2000's that the US began to revisit Hall thrusters as a viable space propulsion technology. The research effort culminated with the launch of the first US-built Hall thruster, the 200 W annular Busek BHT-200 aboard the TacSat-2 Air Force Research Lab demonstrator mission in 2006, 35 years after the launch of SPT-50. The first US-built Hall thruster flown

on an operational mission was the Aerojet BTP-4000, which was launched in August 2010 on the DoD Advanced Extremely High Frequency (AEHF) GEO satellite. At 4.5 kW, it is also the highest powered Hall thruster ever flown in space. This thruster is credited with saving the mission after the main liquid rocket designed to insert the spacecraft in its final orbit malfunctioned [29].

2.2.1 Fluid Modeling

The renewed in interest in Hall thrusters has also led to development of simulation codes. The codes developed to study Hall thrusters can be generally divided into three primary categories: fluid, kinetic, and hybrid. The codes in the first category treat plasma as fluid and solve a set of magnetohydrodynamics equations in a manner similar to the standard computational fluid dynamics. The benefit of this approach is its speed. These kinds of codes can often complete in minutes. However, in general they are not particularly useful beyond determining global thruster parameters such as thrust. The reason is that fluid codes assume a certain (Maxwellian) distribution function for the plasma and ignore the temperature variation in the device. Some examples of fluid modeling of Hall thrusters is the effort of Keidar[30], Yim[31], and Roy[32].

2.2.2 Kinetic Modeling

At the opposite end of the simulation spectrum are kinetic codes. Kinetic codes represent plasma by a collection of computational macroparticles. In essence, the particles are discrete representation of the velocity distribution function. In a fully-kinetic code, both ions and electrons are treated in this manner. Interparticle forces are computed with the help of a computational grid. Particle positions are scattered to the mesh to obtain the spatial variation in charge density, ρ. The electric potential is then computed by solving the Poisson's equation, $\nabla^2\phi = -\rho/\epsilon_0$ and the electric

field is recovered from $\vec{E} = -\nabla\phi$. Particle positions are updated by first integrating velocity from $\vec{F} \equiv m d\vec{v}/dt = e(\vec{E} + \vec{v} \times \vec{B})$ using field values gathered onto the particle position. The integration is performed through a small finite time step Δt. The same time step is used to push particles to a new location by integrating the position $\vec{x}^1 = \vec{x}^0 + \vec{v}\Delta t$. This computational technique is known as particle-in-cell[33].

The benefit of the kinetic approach is that it is based on fundamental physical laws and is free, in general, of simplifying assumptions. The kinetic approach can thus self-consistently determine the VDF of the individual species instead of prescribing a distribution *a priori* as is the case with fluid codes. The drawback of these codes is their computational cost. Solution of the Poisson's equation requires mesh densities fine-enought to resolve the Debye length. Fundamental frequencies on electron time scales, such as the cyclotron and plasma frequency must also be resolved, leading to the necessity to use very small time steps. This makes these codes practical only for microscopic scales, and geometry or electron mass scaling must be utilized if real-sized devices are to be considered.

Yet, even given these difficulties, fully-kinetic modeling has been applied in the past with some success to Hall thrusters. Of interest are the codes of Hirakawa[22] (which interestingly was one of the first codes designed specifically for modeling Hall thrusters), Szabo [21], and Smirnov [7]. These codes typically requires supercomputers and weeks of computational time. Kinetic codes have also been used to study just a subset of the Hall thruster problem. Some such examples include the work of Wu [34] and Yu [35]. Of significant importance is also the code of Sydorenko [36]. Sydorenko developed an implicit 1D kinetic code for modeling electron diffusion in bounded magnetized plasma in a planar geometry, EDIPIC. The concepts put forward in [36] form a basis for the kinetic analysis presented in this research.

2.2.3 Hybrid Modeling

The most popular approach for modeling Hall thrusters relies on hybrid modeling. In this approach, ions and neutrals are represented by kinetic particles, while a fluid model is used for the electrons. These codes operate in 2D and typically consider the $r-z$ plane. Azimuthal symmetry is assumed. Early effort in this category included the HPHall code of Fife [37] and a code by Boeuf and Garrigues [38]. These codes were developed in late 1990s. Since then, the underlying principles of these codes have been adapted by a wide range of researchers. These include efforts of Koo [39], Parra and Ahedo [40], Hofer [41], and Scharfe [42]. The codes in this category generally run on the order of hours to days. A primary issue with these codes is that they assume quasineutrality and uniform Maxwellian temperature along each field line. Because of the first assumption, these codes do not resolve the near-wall plasma sheath and as such do not self-consistently calculate correctly the velocities with which ions strike the wall. As such, analytical models, such as the one of Ahedo [43] must be applied to correct the ion velocity. These codes have been used with some success in the past to model Hall thruster discharges [44] and internal erosion [45].

2.3 Thermalized Potential Model

The hybrid codes typically follow the approach outlined by Morozov [46] and divide the electron dynamics into directions along and perpendicular to the field line. First, in the direction along the magnetic line, the temperature is assumed to remain constant. Since plasma is assumed isothermal, any magnetic mirror drops out [47], and the force balance may be written,

$$\nabla_{\parallel} P_e = \nabla_{\parallel}(n_e k T_e) = e n_e \nabla_{\parallel} \phi \tag{2.8}$$

which is integrated

$$\phi - \frac{kT_e}{e}\ln(n_e) = \phi_0 - \frac{kT_e}{e}\ln(n_0) \tag{2.9}$$

In HPHall, the following expression is used instead, alleviating the need to define a reference density

$$\phi - \frac{kT_e}{e}\ln(n_e) = \phi^*(\lambda) \tag{2.10}$$

The ϕ^* term is the thermalized potential. Knowing ϕ^* and kT_e along each field line allows us to determine the spatial variation in potential, ϕ from ion density n_i. Quasineutrality is assumed, $Zn_i \approx n_e \approx n$, which is a valid assumption in the bulk region removed from the near-wall sheath. This approach reduces the dimensionality of the electron solve from the 2D $r-z$ domain to a 1D λ domain, where λ is the magnetic field line.

The other term that is necessary in order to determine potential is the electron temperature. In fact, the electron fluid model implemented in HPHall can be primarily thought of as a temperature solver. The electron temperature is given by the energy equation,

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_e kT_e\right) + \nabla\cdot\left(\frac{5}{2}n_e kT_e\vec{u}_e + \vec{q}_e\right) - \vec{u}_e\cdot(n_e kT_e) = S_e - S_i \tag{2.11}$$

where S_e and S_i are energy losses due elastic and inelastic collisions with neutrals and ions, and the heat term $\vec{q}_e$ also includes energy losses to the wall. These properties are area averaged over the magnetic field line.

The electron cross-field velocity is obtained from the force balance presented in Equation 2.8

$$j_{e,\hat{n}} = -en_e u_{e,\hat{n}} = \sigma_{e,\perp}\left(E_{\hat{n}} + \frac{1}{en_e}\frac{\partial p_e}{\partial \hat{n}}\right) \tag{2.12}$$

or

$$u_{e,\hat{n}} = -\mu_{e,\perp}\left(E_{\hat{n}} + \frac{1}{en_e}\frac{\partial p_e}{\partial \hat{n}}\right) \tag{2.13}$$

This expression is further simplified using $p = n_e k T_e$ and $\vec{E} = -\nabla\phi$ to obtain

$$u_{e,\hat{n}} = \mu_{e,\perp}\left(\frac{\partial\phi}{\partial\hat{n}} - \frac{kT_e}{en_e}\frac{\partial n_e}{\partial\hat{n}} - \frac{k}{e}\frac{\partial T_e}{\partial\hat{n}}\right) \tag{2.14}$$

Fife further simplified this expression by differentiating 2.10 to obtain an expression in terms of the thermalized potential

$$u_{e,\hat{n}} = \mu_{e,\perp}\left(\frac{\partial\phi^*}{\partial\hat{n}} + \frac{k}{e}\left(\ln(n_e) - 1\right)\frac{\partial T_e}{\partial\hat{n}}\right) \tag{2.15}$$

Here $\mu_{e,\perp}$ is the mobility term which in HPHall is evaluated following a model analogous to Equation 2.7. This expression also highlights an important relationship. In the channel current is conserved, $\nabla \cdot (n\vec{u}) = 0$, assuming steady state and no losses to the dielectric walls. Then $\int_{\lambda_1} n_{e,1} u_{e,1} dA = \int_{\lambda_2} n_{e,2} u_{e,2} dA$. Quasineutrality dictates $n_e \approx z N_i$ and since n_i of the unmagnetized ions is not coupled to the field, the electron cross-field velocity must adjust in order to satisfy current conservation. This in turn requires an increase in electron temperature. The increased electron temperature increases the ionization rate. When combined with neutral density, this coupled behavior results in an oscillatory steady-state in which the discharge parameters adjust self-consistently.

2.4 Plume Modeling

The hybrid kinetic ions / fluid electrons model is also used to study plumes of these devices. Even though Hall thrusters are open to the ambient environment, influence of the downstream properties is typically ignored due to many orders of magnitude difference in the respective densities. For comparison, in LEO with a relatively dense background ion population compared to GEO, the background density of ions is of O(12). The density of ions produced by the thruster is of O(17). The

magnetic field also decays rapidly outside the thruster, reducing the Lorentz force to the electrostatic form $\vec{F} = q\vec{E}$, unless Earth's magnetic field needs to be included. It is not practical to solve Poisson's equation in the plume and instead potential is obtained from a model such as the one given in Equation 2.9, except that in this case the values ϕ_0, kTe, and n_0 are reference parameters based on properties at single known location. Of importance in plume codes are interparticle collisions, as well as the initial particle VDF and the interaction of the thruster plume with spacecraft components. Some examples of codes in this category include the work of Brieda [48], Boyd [49], VanGilder [50], and Santi and Cheng [51]. The plume modeling field is more mature than the thruster modeling effort. However, as demonstrated in [52], the initial VDF with which the ions are injected into the plume simulation plays a crucial role in the downstream plume properties. Since ions in Hall thruster are generally non-Maxwellian, it is difficult to provide the correct description using analytical models.

2.5 Multiscale Approach to Modeling Hall Thrusters

This thesis describes a novel approach for modeling Hall thrusters and their plumes based on a multiscale formulation. The approach is based on the fact that hybrid HPHall-like codes offer a valid environment for studying the discharge and obtaining inputs for plume codes, except that they suffer from two limitations. First, due to the fluid implementation of electron motion, electron mobility is not captured correctly. Secondly, since potential is obtained assuming quasineutrality, the sheath is not resolved, resulting in incorrect near-wall ion dynamics. Related to the first item is the realization that since in Hall thrusters electrons are magnetized, their dynamic along each magnetic field lines can be decoupled. This then allows us to propose a novel approach for determining electron mobility. Instead of computing mobility from an analytical expression, mobility can be computed self-consistently from a kinetic

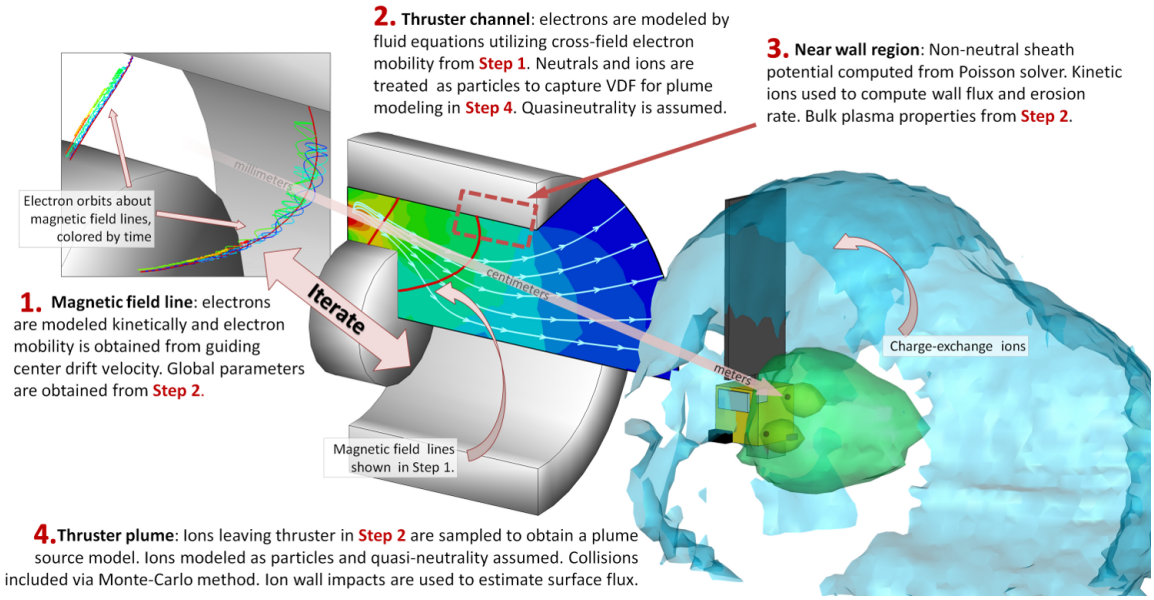


Figure 2.3: Graphical representation of the multiscale approach.

simulation of electrons. Since mobility needs to be known only along a small number of magnetic field lines (the computational grid lines for the quasi-1D electron fluid model), the computational effort is significantly reduced compared to fully-kinetic codes. Similarly, the near wall ion motion can be analyzed by considering a small subdomain along the wall. Then due to the reduced domain dimension, the non-neutral potential can be solved and ion motion can be determined self-consistently. The hybrid code can furthermore be used to obtain the source model for the plume code by sampling ion particles entering the plume.

The multiscale approach is shown visually in Figure 2.3. This approach somewhat diverges from the multiscale approach utilized in fields such as fluid-dynamics in that it considers the different spatial scales independently, but couples them via shared data. Each spatial scale is treated such that the important physics is captured in an efficient manner. The ultimate goal of this effort is a simulation tool capable of modeling the entire internal-to-external behavior of Hall thrusters in a relatively rapid time frame of hours to days utilizing only a standard desktop workstation. Although the actual implementation of the combined simulation tool remains as future work,

this thesis outlines the physics that needs to be captured by each component of interest.

The spatial scales used in this work are enumerated in Figure 2.3 and are also summarized below. The smallest spatial scale is the scale of a single magnetic field line. On this scale, electron motion is described by the kinetic cyclotron orbit about field lines, and electron transport can be determined directly by considering the speed with which electron guiding centers travel. The second spatial scale is the scale of the thruster. Here it is no longer feasible to treat electrons as particles. Instead they are modeled as a fluid. Ions and neutrals on the other hand can be treated kinetically, which allows the code to sample exit plane properties for use in a plume model. A subset of the thruster spatial scale is the near-wall region. By considering the near-wall segment it is possible to directly compute the flux of ions to the wall and the resulting sputter rate. Finally, there is the spatial scale of the satellite. On this scale, the thruster is no longer modeled, instead it is used solely to specify the injection source properties. The spatial scales are summarized below:

Magnetic Field Line: On the spatial scale of a magnetic field lines, dynamics is driven by the cyclotron motion of electrons. Electrons are magnetized, and individual field lines can be considered independent of each other. Heavy particle and properties normal to the field lines are assumed frozen. This approach allows us to rapidly simulate electrons and recover mobility self-consistently. Leveraging modern multi-core architectures via multithreading allows for multiple field lines to be studied simultaneously.

Thruster Channel: On the spatial scale of the thruster, plasma is assumed to be quasineutral, and electron density can be obtained from kinetic ions. Electron temperature and plasma potential is obtained by solving the quasi-1D equations.This approach is identical to standard models, except that mobility is

determined self-consistently by the kinetic analysis. Ions exiting the thruster at steady state are sampled to obtain a discretized source term for plume modeling.

Near-Wall Region: In order to predict thruster lifetime, it is necessary to capture the ion dynamics in the near-wall sheath region. However, resolving the sheath is not practical on the spatial scale of a thruster since it requires that very fine computational mesh is used everywhere. However, once the bulk plasma properties are known, a wall interaction simulation can be performed on the near-wall subdomain.

Plume Environment: Outside the thruster exit, the magnetic field plays a negligible role. The plume is quasineutral except in low density sheath regions around the spacecraft. Of interest here is the radial expansion of the charge exchange ions and their impact on spacecraft components. Electron density is obtained from quasineutrality, and potential can be obtained by inverting the Boltzmann relationship or by solving Poisson's equation.

A significant contributor to this multiscale model is a code developed to model the motion of magnetized electrons and determine electron transport, Lynx. Many of the concepts for this code are based on the previous efforts of Sydorenko. However, as outlined in the following chapter, Lynx differs from EDIPIC in several important areas. First, only electrons are simulated. Ions and neutrals are assumed to remain stationary. Secondly, the explicit formulation is used. The benefit of an implicit formulation, namely larger time step that can be used for stability, is often lost in time-dependent simulations in which high-frequencies need to be resolved. In that case, an implicit formulation may lead to a reduced code performance due to the added computational cost of inverting matrix equations. Finally, while EDIPIC was developed for the radial field-line topology, the current code is applicable to curved magnetic field lines found in real-world Hall thrusters.

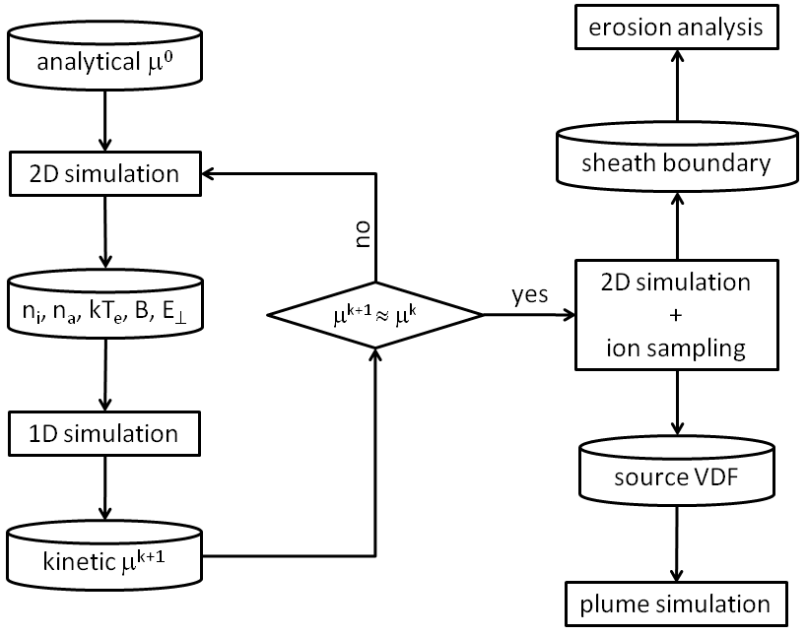


Figure 2.4: Schematic of the multiscale approach

The multiscale approach requires the components to utilize a set of shared data such that the individual results can be correlated. The kinetic simulation require as inputs the densities of heavy particles interpolated onto the grid line, as well as the variation in the perpendicular electric field component, $E_\perp$. These properties correlate the field line with a state in the global discharge model. The discharge solution in the near-wall region is also used to provide the bulk plasma properties for the wall analysis. This conceptual methodology is illustrated in Figure 2.4. First, global discharge parameters are obtained via a hybrid code such as HPHall. The classical mobility terms are used in this initial run. Mobilities are then recalculated by running kinetic simulations of electron transport for the required magnetic field lines. Since the kinetic treatment is applied to only a small subset of the thruster (a predefined number of field lines), transport can be computed in very short time periods (minutes) utilizing common desktop computers. This updated map of $\mu_\perp$ is then used in the hybrid code to obtain a correction for the global parameters. This procedure is repeated until convergence, given by $\partial\mu/\partial t \sim 0$. Although not computed

in the present version of the kinetic code, the kinetic analysis could also be used to calculate the wall heat losses. This heat flux is one of the terms used in the electron energy equation, Equation 2.11.

Once convergence is achieved, another hybrid simulation is performed at the converged set of $\mu_\perp$ values. This last run is used to sample data for both the wall erosion and plume analysis. The required inputs for the wall analysis are the plasma densities in the bulk plasma region outside the sheath as wells as the potential profile. This integration forms a one way coupling, since it can be assumed that ions impacting the wall recombine and are removed from the simulation. The inputs for the plume code are obtained by sampling particles as the cross some defined plane. These particles form a spatially-resolved discretized velocity distribution function of beam ions. Again, only one way coupling is assumed. Even though Hall thruster discharge channel is open to the ambient environment, the ambient conditions are not expected to play an important role on the internal thruster dynamics due to the rapid decrease in plume densities outside the thruster.

The actual implementation of the iterative approach as shown in Figure 2.4 remains as future work. Instead, only a single coupling between the kinetic and thruster codes was attempted, and the present near-wall analysis was performed using an analytical configuration. In this thesis the details of the individual constituents are described instead. Although the code is applied to the CHT, this case serves primarily to demonstrate the concept and does not constitute a detailed analysis of this device. One of the reasons for this shortcoming was a difficulty in utilizing HPHall for the non-annular geometry used in the CHT. This issue is discussed in more detail in Chapter 5. The kinetic code used to model electron transport is introduced in the next chapter.

Chapter 3

Kinetic Modeling of Electron Transport

3.1 Overview

This chapter describes the kinetic code developed to analyze the microscopic spatial scale of electrons. The objective of the code, named Lynx, is to self-consistently determine mobility of magnetized electrons orbiting about a one or more magnetic field lines. The field line can be curved and can also be subject to a magnetic mirror. Scattering events, such as collisions and wall interactions, are simulated directly. Transport is determined from the translational speed of electron guiding centers.

The code uses a hybrid version of the particle-in-cell (PIC) technique introduced in Chapter 2. In this formulation, only electrons are simulated. Ions and neutrals are assumed to be stationary and are described by a fixed background density. The reason for this approach is two fold. First, it reduces the computational overhead since a smaller number of particles needs be considered. Secondly, it allows the kinetic simulation to correlate with the global discharge simulation. In a Hall thruster, ions are not magnetized, and are continuously streaming across the field line. Treating ions kinetically and allowing them to redistribute along the field line according to potential gradients would result in smoothing of ion density variation, and a solution no longer corresponding to the global discharge ion density. Keeping ions stationary imposes the assumption that mobility develops instantaneously in response to global

discharge parameters. This is believed to be a valid assumption, since in the frame of reference of ions, electrons respond instantaneously to disturbances due to their light mass.

The simulation begins by loading from an input file plasma properties relating the individual field line to the global discharge. Namely, ion and neutral number densities, n_i and n_a, the cross-field electric field $E_\perp$, the magnetic field strength B, and the initial electron temperature kT_e are loaded. The code assumes that n_i is the total charge averaged ion density, i.e., ion charge density $\rho_i = en_i$. In addition, the code reads in ion velocity components $v_\perp$ and $v_\parallel$, however these are not utilized in the current version. They are included for a possible future correction of ion wall flux. The initial electron temperature is assumed to be isotropic and uniform following the model utilized in HPHall. The code then creates electron particles following the prescribed ion density, $n_e = n_i$ everywhere. The parallel component of electric field $E_\parallel$ is determined self-consistently by solving Poisson's equation, $\nabla^2\phi = -\rho/\epsilon_0$ at each time step. Electrons are advanced according to the leapfrog method. The code initially runs in a sheath-forming mode in which electrons impacting the walls are absorbed and collisions are ignored. This mode continues until a well-formed sheath with $j = 0$ at both walls is formed. Since the high-energy electrons will be preferentially lost during this sheath formation stage, the code subsequently resamples the electron population.

The simulations then continues in a normal mode in which the user-specified wall and collision model is utilized. The simulation continues for a user defined number of time steps and properties of interest are averaged starting with time step $it = it_{steadystate} + \Delta it_{sampling}$. Mobility is recovered from the speed with which the electron guiding centers diffuse. Three primary transport mechanisms are considered: particle collisions, wall interaction, and electric field instabilities. Collisions are modeled with a Monte Carlo approach and several collisional processes, such

as momentum transfer (scatter), Coulomb collisions, ionization, and excitation, are included. Wall interactions are modeled following the approach of Sydorenko [36]. Support for modeling azimuthal waves is also included, however, this mechanism is generally not used.

As will be detailed in the Implementation section 3.7, a typical Lynx analysis consists of multiple threads running in parallel, with each thread analyzing a single magnetic field line. The following sections detail the implementation of each computational thread. As such, each simulation is one dimensional. By running several simulations for different field lines, the code is able to compute the transport variation in the channel. Computational times are on the order of minutes for a typical simulation consisting of 100,000 particles and 20,000 time steps. Generally, steady state is formed in the first couple hundred steps, and averaging begins with step 1000. Typical value of Δt is 2×10^{-11} s. At this Δt, the simulation completes approximately 10 ion steps in the time used to average data over 20,000 iterations. 100,000 particle simulations result in several hundred particles per cell. Cell spacing is determined automatically such that Debye length is resolved. Typical simulations contain between several hundred to a thousand nodes.

3.2 Computational Domain

The general computational domain for each 1D analysis is is sketched in Figure 3.1. The domain consists of a curved magnetic field line connecting the inner and the outer thruster wall. Along this domain, the coordinate system consists of two components: the distance s along the field line, and a coordinate system $x - y$ in the plane normal to the line. The distance is measured from the inner wall. The other two coordinate directions, $\hat{\vec{e}}_x$ and $\hat{\vec{e}}_y$ are normal to $\hat{\vec{e}}_s$. These coordinates are defined such that $\hat{\vec{e}}_x$ lies in the $r - z$ plane of the thruster, while $\hat{\vec{e}}_y$ lies in the $r - \theta$ plane. In the numerical implementation, these coordinates are also defined as $\hat{\vec{e}}_x \equiv \hat{\vec{e}}_0$, $\hat{\vec{e}}_y \equiv \hat{\vec{e}}_1$,

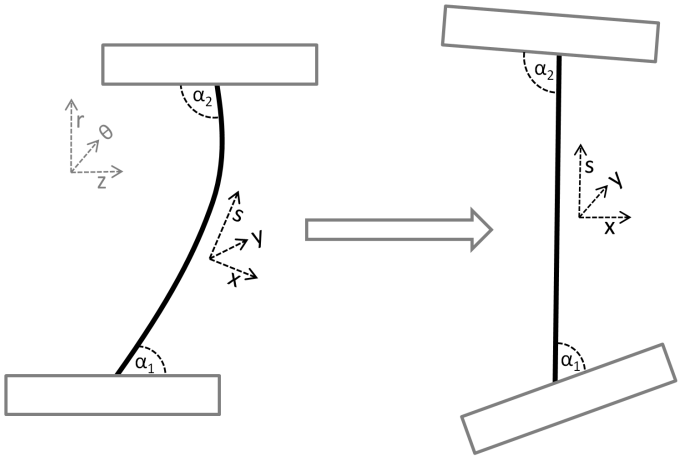


Figure 3.1: Schematic of the computational domain. The image on left shows the magnetic field line in physical coordinates. The image on right corresponds to the internal representation.

and $\hat{\vec{e}}_s \equiv \hat{\vec{e}}_2$. This 0-2 index notation is used to match the indexing with the internal representation in the underlying programming code. The simulation is implemented in the Java programming language, in which array indexing starts with 0. This semi-planar approach results in the particle $\vec{E} \times \vec{B}$ drift occurring in the $\pm y$ direction. In order to allow for a rotation of this motion into the original $r - z$ system for visualization purposes, the code also keeps track of particle's angular displacement, θ.

In the code, it is assumed that electron orbital planes rotate automatically with the magnetic field line. As such, the actual profile of the field line is not required in the particle push. The shape of the line is utilized only to determine the variation in electric and magnetic fields along s. Magnetic field gradient, $\partial B/\partial s$ is computed by differentiating the supplied $B = B(s)$ magnetic field profile. The shape of the magnetic field line is also of importance when considering injection of secondary electrons at the wall. From Figure 3.1 we can see that the field line intersects the wall at some angle α_1 and α_2, which can be computed from $\tan \alpha_1 = \Delta r/\Delta z$. These angles comes into play in the wall emission model which rotates emitted secondary electrons onto the magnetic field line according to the incidence angle.

The field line is discretized into a number of equidistant computational segments (one-dimensional cells) in accordance with the particle-in-cell (PIC) method. The length of each segment is Δs. This cell spacing is set such that $\lambda_{D,min} = h\Delta s$, where $\lambda_{D,min} = \sqrt{\epsilon_0 kT_e/n_{p,max}e^2}$ is the minimum Debye length along the entire simulation domain, and h is a user provided parameter. Typically, $h = 2$ is used, resulting in two cells per Debye length. Debye length is calculated using the constant initial electron temperature, which is assumed to be constant along the field line $\partial kT_e/\partial s = 0$ and isotropic, $kT_{e,\perp} = kT_{e,\parallel}$. It should be noted that this and prior analysis [53] indicates that electron distribution function in Hall thrusters becomes anisotropic due to the presence of cold secondary electrons. In the simulation, the kinetic EEDF is characterized, however this value is used only for diagnostic purposes.

Simulation parameters as well as various computed results are stored on the grid nodes. Linear interpolation is used to obtain values at intermediary points. The stored values include the fixed properties describing the global environment such as n_a, n_i, kT_e, B, and $E_\perp$, as well as number of parameters determined by the code. The second group includes the plasma potential ϕ, the parallel component of electric field $E_\parallel$, the magnetic field gradient $\partial B/\partial s$, electron density n_e, the average drift velocity v_d, the collision rate ν, mobility μ, and the cross-field electron and total currents j_e and j. At the completion of the analysis, these properties are interpolated onto a coarse HPHall-like mesh for subsequent use in the 2D code. The detailed meshes are also exported at a user defined interval. In addition, the code outputs a random collection of particles, velocity distribution functions, particle traces, and velocity phase plots.

3.3 Potential Solver

Lynx solves the electric field in the parallel direction, $E_\parallel$ self-consistently from the Poisson's equation, $\nabla \cdot \vec{E} = -\nabla^2\phi = \rho/\epsilon_0$. The discretized form of this equation is

determined from the finite volume approach. We can define a control volume around each simulation node and perform the integration

$$\begin{aligned} \int_V \nabla \cdot \nabla\phi dV &= \int_V -\frac{\rho}{\epsilon_0} dV \\ \int_S \nabla\phi \cdot \hat{n} dA \cong \sum_{i=1}^{4} \nabla\phi \cdot \hat{n} dA &= -\frac{\rho}{\epsilon_0} r\Delta x \Delta y \Delta s \\ -\nabla\phi_1 r_1 + \nabla\phi_3 r_3 &= -\frac{\rho}{\epsilon_0} r\Delta s \end{aligned} \tag{3.1}$$

where $r = f(s)$ and $\nabla\phi_2$ and $\nabla\phi_4$ are zero since for the one dimensional case we have $\partial\phi/\partial x = 0$. Axisymmetry dictates $\partial\phi/\partial y = 0$. Here the indexes correspond to the control volume face, starting with the bottom face and continuing around the control volume counter-clockwise. We next need an expression for the two derivatives $\nabla\phi_1$ and $\nabla\phi_3$. These are obtained using the scalar form of the divergence theorem, $\int_S \nabla\phi dA = \oint \phi \hat{n} dl$. For our formulation this yields the standard finite difference derivate, $\nabla\phi_1 = (\phi_j - \phi_{j-1})/\Delta s \vec{\hat{e}}_s$ and $\nabla\phi_3 = (\phi_{j+1} - \phi_j)/\Delta s \vec{\hat{e}}_s$. Substituting these expression into Eq. 3.1 gives

$$\frac{\phi_{j+1} - \phi_j}{\Delta^2 s} \frac{r_{j+1/2}}{r_j} - \frac{\phi_j - \phi_{j-1}}{\Delta^2 s} \frac{r_{j-1/2}}{r_j} = -\frac{\rho}{\epsilon_0} \tag{3.2}$$

This expression can be checked for the case of a fully radial magnetic field line where $s \equiv r$ and $\Delta s \equiv \Delta r$, yielding

$$\frac{\phi_{j+1} - 2\phi_j + \phi_{j-1}}{\Delta^2 s} + \frac{1}{r} \frac{\phi_{j+1} - \phi_{j-1}}{2\Delta r} = -\frac{\rho}{\epsilon_0} \tag{3.3}$$

which is the standard discretization of $\nabla_r^2 = \partial^2/\partial r^2 + (1/r)\partial/\partial r$. The code includes support for both this cylindrical approach and a planar mode in which the standard $\nabla^2\phi = \phi_{j-1} - 2\phi_j + \phi_{j+1} = -\Delta^2\rho/\epsilon_0$ is used. The planar mode is useful for testing purposes, and for analyzing discharges with a planar geometry. The electric field $\vec{E} =$

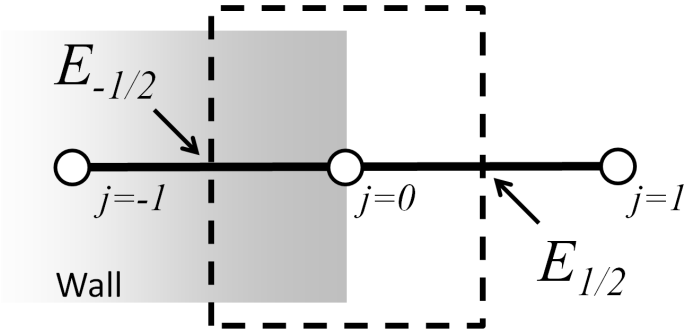


Figure 3.2: Electric field at the wall interface

$-\nabla\phi$ is obtained by numerically differentiating the potential, $E = (\phi_{j-1} - \phi_{j+1})/2\Delta s$. This differentiation is independent of the geometry model. A one sided model is used along the walls.

3.3.1 Boundary Conditions

The expression given in Eq. 3.2 holds inside the simulation domain. This expression needs to be modified along the walls to take into account the boundaries. Two types of boundary conditions are implemented in the code: conductor and dielectric walls. A conductor wall is assumed to be connected to a power supply keeping the potential constant. This wall type is approximated by applying the Dirichlet boundary condition, $\phi_0 = \phi_{inner}$ and $\phi_{nj-1} = 0$. The boundary condition becomes more complicated if dielectric walls are present. The discretization along the inner wall can be determined by considering a small volume elements centered at the $j = 0$ grid node. From Gauss' law, we have

$$\int_V \nabla \cdot \vec{E} ds\, dA = \frac{Q}{\epsilon_0} \tag{3.4}$$

where Q is the total charge enclosed by the volume. Applying the divergence theorem and considering a domain with the field varying in only a single direction, we obtain $(E_{1/2} - E_{-1/2})dA = (\sigma + 0.5\Delta s\rho)dA$. This formulation assumes that $dA_0 = dA_{1/2} =$

$dA_{-1/2} = dA$, i.e. a planar geometry. The variation in dA is not considered in the code due to the small Δs spacing between nodes and hence a negligible difference in r. The area element can then be eliminated, giving the boundary condition

$$E_{1/2} = \frac{\phi_0 - \phi_1}{\Delta s} = \sigma + \frac{1}{2}\rho\Delta s \tag{3.5}$$

The internal electric field, $E_{-1/2}$ is assumed to be zero inside the dielectric. This integration scheme is illustrated in Figure 3.2. A more detailed treatment, including the correction due to the varying r is found in [54]. Since a potential solution is unique only up to a fixed constant, at least a single Dirichlet boundary condition must be applied to define the problem. This condition is applied by fixing the potential on the outer wall to 0 V, following the formulation in [54] and [36]. It should be noted that this approach appears to neglect the surface charge accumulated on the outer wall. Including the electric field gradient on the outer wall would over-specify the problem as it would result in $N+1$ equations for N unknowns. The outer surface charge is instead included by the condition of zero total plasma charge, $\int \rho dV + \int(\sigma_0 + \sigma_{nj-1})dA = 0$

The resulting system of equations produces a tri-diagonal matrix that can be easily solved using the direct Thomas Algorithm [55]. Electric field is then retrieved from $\vec{E} = -\nabla\phi$, or $E_s = (\phi_{j-1} - \phi_{j+1})/(2\Delta s)$. It should be noted that this method produces just one of three electric field components acting on the particle. The other two components are $E_\perp$ and E_y. The first of these comes directly from the axisymmetric solution and is driven by the potential drop along the thruster channel. The second, E_y is the electric field in the azimuthal direction. HPHall assumes the ideal axisymmetric case and assumes this field be zero. However, azimuthal waves and additional oscillations oscillations [25] have been previously observed experimentally. These waves are believed to be an important contributor to electron transport. To investigate their role, Lynx includes an analytical harmonic oscillator that can be

used to modulate the azimuthal electric field. The magnitude is given by $E_q = E_0 + A\cos(-\omega t + \varphi)$. It should be noted that this modulation is generally not used, it is used only to characterize the influence of electric field fluctuations in the subsequent chapter. Additionally, oscillations in the normal direction are not considered since these drive diffusion in the azimuthal direction which is not of interest to the present analysis.

3.4 Particles

3.4.1 Particle Loading

The code contains two subroutines for generating particles: (a) quiet start and (b) random injection. The quiet start method is used to initially populate the electron species. This method creates particles at node locations such that $\rho = n_i - n_e \sim 0$ (limited by the statistical weight of particles). If total electron and ion charges $Q_e = \int_s e n_e ds$ and $Q_i = \int_s n_i ds$ are initially equal $Q_e = Q_i$ this loading will result in $\rho \sim 0$ everywhere. The initial potential energy of such a system $\int_s \rho\phi \sim 0$ and the energy is determined by electron temperature alone. The simulation begins by loading such a quasineutral system. Lynx uses a constant specific weight (the number of real particles represented by a single simulation macroparticle). Although other researchers [56] indicate that a variable specific weight should be used to represent the high energy tail of the velocity distribution function, such an approach was not used at present due to it imposing its own difficulties. Namely, variable weight models typically rely on methods for combining and splitting particles. Care must be taken when combining two particles of non-equal velocities to assure that numerical changes in the VDF are not accidentally included. For this reason, a constant specific weight model was used, with implementation of the variable weight model identified as a possible future enhancement. The ability of this model to resolve the high energy tail

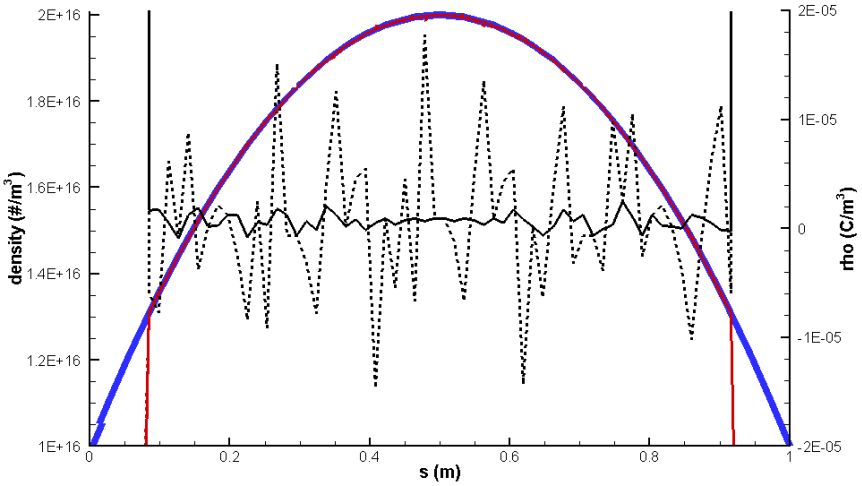


Figure 3.3: Electron and ion densities, and charge density for $Q_e = 0.9Q_i$.

is further discussed in the next chapter.

The specific weight of each electron is determined by integrating the ion density along the field line to determine the total number of ions in the 1D domain, $N_i = \int_s n_i dV$. The specific weight is then set such that a user-defined number of macroparticles M_0 will represent this count, $w_e = N_i/M_0$. The quiet start routine begins loading this number of particles from center of the magnetic field line. Particles are first loaded at the location of the center node(s), and the loading continues towards the boundaries until the specified number of particles is loaded. The number of particles to load at each node is given by $p = n_{i,j}\Delta V/w_e$, where $n_{i,j}$ is the ion density at the $j-$th node and ΔV is the cell volume. $0.5\Delta V$ is used along the boundaries. Since the particle count p will in general be non-integral, the scheme keeps a track of fractional particles and an additional particle is loaded whenever the fractional count reaches unity.

The importance of this scheme is its ability to load a sheath-like electron distribution. In a wall-bounded plasma, $Q_e < Q_i$ due to a faster initial depletion of electrons in the sheath. The ratio of Q_e/Q_i at steady state is not known *a priori* and must be determined self-consistently. The approach used by the code is to first load an electron population with $Q_e = Q_i$ and allow the simulation to establish a steady state

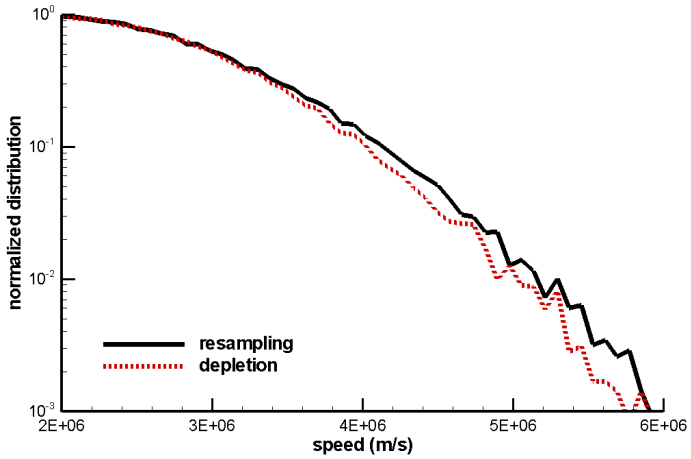


Figure 3.4: The effect of particle resampling on the tail of the velocity distribution function

marked by zero wall current. The charge ratio then becomes N_e/N_i, assuming $Z_i = 1$. The number of electrons at steady state is recorded and subsequently used to reload the electron population. Reloading is accomplished by first destroying all particles and then injecting a new population with a reduced electron count. This reloading allows the simulation to replenish the high velocity tail of the electron EDF that is initially lost to the wall. Since electrons are loaded from the center out, particle loading with $Q_e < Q_i$ will result in quasineutrality in the bulk, and a near-wall region where $Q_e = 0$. This is demonstrated in Figure 3.3. This picture shows electron loading for an analytical quadratic ion density variation. The blue curve shows the ion density, while the red curve shows the neutral density. In this loading, $Q_e = 0.9Q_i$ was used. The black curves show the charge density $\rho = e(n_i - n_e)$. The effect of particle count is shown by the two black curves. Increasing the count reduces the noise in ρ. However, for both cases $\rho \sim 0$ in the bulk. It should be noted that this loading scheme assumes that the sheaths at the inner and the outer wall are symmetric.

The importance of resampling is shown in Figure 3.4. This figure focuses on the high energy tail of the electron velocity distribution function. The dashed red curve shows the velocity distribution function at steady state prior to resampling. The

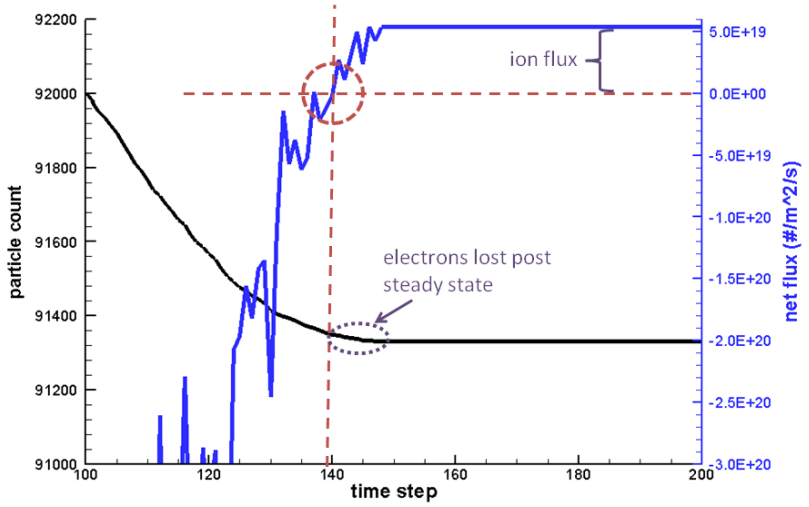


Figure 3.5: Electron count and wall current vs. iteration timestep. Ion density $n_i = 10^{16}$ m^{-3} and $kT_e = 10$ eV.

black curve shows the distribution obtained by resampling the electron distribution. As can be seen from this plot, the two distributions agree in the low energy region, but begin to diverge at approximately 3×10^6 m/s. Resampling allows the distribution function to repopulated the high energy tail.

Related to particle loading is the internal characterization of a fully developed plasma sheath. This is accomplished by calculating the total net current flux to the walls, $\Gamma_{tot} = \Gamma_{outer} + \Gamma_{inner}$, where each flux consists of ion, electron, and secondary electron fluxes, $\Gamma = \Gamma_i - \Gamma_e + \Gamma_s$. The ion flux is obtained by assuming that ions enter the wall with the Bohm velocity, $v_b = \sqrt{kT_e/m_i}$ from the sheath edge with density equal to the average ion density in the simulation. Currently, the actual ion velocity in the near wall region obtained from the 2D code is not taken into account. The sheath is established once $\Gamma_{net} >= 0$. The evolution of a typical simulation is plotted in Figure 3.5. The black curve shows the number of particles while the blue curve plots the net wall flux. The point at which wall flux reaches zero is indicated. This particular simulation was allowed to run past this point resulting a complete depletion of energetic electrons able to reach the wall. The population highlighted in the dashed ellipse corresponds to the electrons continuously flowing to the wall in

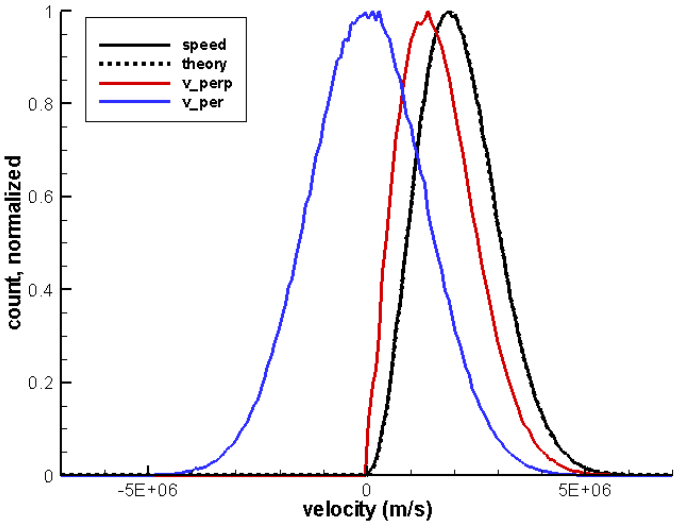


Figure 3.6: The perpendicular, parallel, and combined velocity distribution functions

steady state to compensate for ion current and an important contributor to a possible near wall electron mobility.

The above described method is used to load the initial electron population. It is not used during the normal simulation; another method is implemented to replenish particles lost to walls or domain boundary. This method samples particles at a random spatial location, with the probability following the prescribed ion density. This scheme works by selecting a random position along the field line, $s = RL$ where R is a random number and L is the domain length. The probability of a particle being found at this position is obtained from $P = N_i(s)/\sum N_i$. This probability is compared to another random number R_2. If $P \geq R_2$, a particle is placed at this location. Otherwise, the process continues.

3.4.2 Loading a Prescribed VDF in Magnetized Plasmas

The particle sampling routines described above select the initial s position of the particle. The initial velocity components $v_\perp$ and v_θ are obtained by sampling the Maxwellian distribution function at the specified temperature, kT_e. Isotropic distribution is used in particle loading, following the model used in HPHall. In the

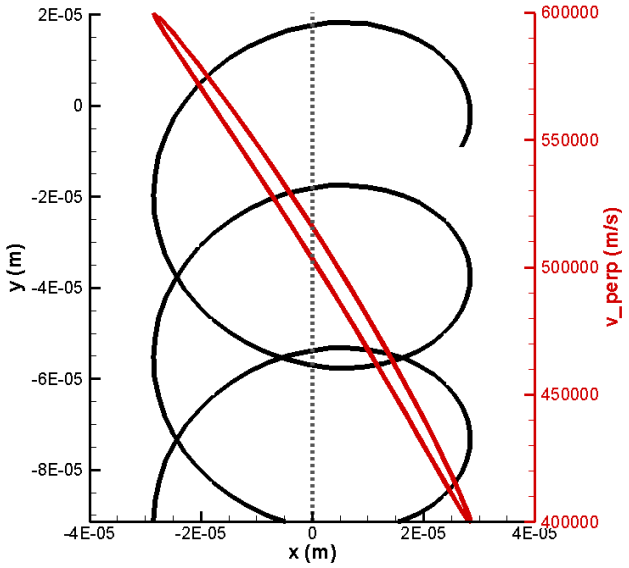


Figure 3.7: Trace of a magnetized particle and velocity vs. position. The spread of the velocity phase plot is directly related to integration Δt.

parallel direction, the velocity component is given by the standard one-dimensional Maxwellian distribution function

$$f_M = (m/2\pi kT)^{1/2} \exp\left(-v^2/v_{th}^2\right) \tag{3.6}$$

Particles are sampled from this distribution by selecting a random velocity in the $-6\sigma : +6\sigma$ range and evaluating the probability $P = \exp(-v^2/v_{th}^2)$. The process continues until $P \geq R$. In the perpendicular direction, the velocity component is obtained by a combination of two independently sampled one-dimensional distributions, $v_\perp = \sqrt{f_{M1}^2 + f_{M1}^2}$. These two velocity distributions can be seen in Figure 3.6. The blue line shows $\mathrm{VDF}_\parallel$ while the red curve shows $\mathrm{VDF}_\perp$. The black curve is the speed distribution of particles. This distribution can be seen to agree with the desired analytical curve, shown by the dashed line. The dashed line is not clearly seen since it is overlapped by the solid black line.

Once the desired $v_\parallel$ and $v_\perp$ velocity components are known, the actual particle is created. As will be discussed next, Lynx uses the position of guiding centers

to determine mobility. As such, it is imperative that particles are correctly placed on an orbit centered at $x_{gc} = 0$. In addition, the loading function must assure that the correct velocity distribution function is retrieved when particle velocities are averaged over an orbital period. Both of these requirements can be satisfied by placing the particle at a random phase location along the analytical orbit. The velocity components of a particle drifting in an $(E_\perp \vec{e}_x) \times (B\vec{e}_s)$ field are given by [28]

$$v_x = v_\perp \cos(\omega_c t) \tag{3.7}$$

$$v_y = - \pm v_\perp \sin(\omega_c t) - \frac{E_\perp}{B} \tag{3.8}$$

These expressions can be integrated to obtain

$$x = r_L \sin(\omega_c t) \tag{3.9}$$

$$y = \pm r_L \cos(\omega_c t) \tag{3.10}$$

where $r_L = v_\perp/\omega_c$ is the Larmor radius and $\pm$ follows the particle charge. The $E_\perp/B$ component is ignored in the position, since the position in the azimuthal direction is not relevant. The typical trace of a particle drifting in a crossed E-B field is shown in Figure 3.7. The black line is the particle trace, while the red curve shows the particle tangential velocity as a function of x position. This particle was loaded with $v_\perp = 5 \times 10^5$ m/s. As can be seen from this figure, the actual velocity of the particle varies significantly during the cyclotron motion and is the factor contributing to the $\vec{E} \times \vec{B}$ drift. The "spread" in the velocity trace is directly related to integration step. A rather large time step was used here to illustrate this effect. The two halves collapse into a single line when the typical $it_{orbit} = 150$ is used. Of importance is the fact that the curve remains closed even for a fairly large Δt, indicating energy conservation.

3.4.3 Particle Motion

Motion of charged particles in an electromagnetic field is governed by the Lorentz force,

$$\vec{F} = m\frac{d\vec{v}}{dt} = q\left(\vec{E} + \vec{v} \times \vec{B}\right) \tag{3.11}$$

Care must be taken when integrating this equation since velocity appears on both sides of the equation. The obvious integration by the forward method, $\vec{v}^{n+1/2} = \vec{v}^{n-1/2} + q/m(\vec{E} + \vec{v} \times \vec{B})\Delta t$ will result in a non-physical energy gain and the particle describing a spiral orbit. The correct closed orbit can be achieved by utilizing an implicit formulation and using matrix inversion to solve for $v^{n+1/2}$. This method is however not computationally efficient. In 1970 Boris [57] described an elegant alternative, which is now known as the Boris method. This method is the *de facto* standard for particle pushing in PIC codes.

We are solving

$$\frac{\vec{v}^{n+1/2} - \vec{v}^{n-1/2}}{\Delta t} = \frac{q}{m}\left[\vec{E} + \frac{\vec{v}^{n+1/2} + \vec{v}^{n-1/2}}{2} \times \vec{B}\right] \tag{3.12}$$

Boris noticed that we can eliminate the electric field by defining

$$\vec{v}^{n-1/2} = \vec{v}^{-} - \frac{q\vec{E}}{m}\frac{\Delta t}{2} \tag{3.13}$$

$$\vec{v}^{n+1/2} = \vec{v}^{+} + \frac{q\vec{E}}{m}\frac{\Delta t}{2} \tag{3.14}$$

When these definitions are substituted into the original equation, we obtain pure rotation

$$\frac{\vec{v}^{+} - \vec{v}^{-}}{\Delta t} = \frac{q}{2m}\left(\vec{v}^{+} + \vec{v}^{-}\right) \times \vec{B} \tag{3.15}$$

Boris next utilized some basic geometry (see for instance Figure 4-4a in [33]) to derive the expression for performing the rotation. The first step is to find the vector bisecting

the angle formed by the pre- and the (to be yet computed) post-rotation velocity. The angle through which the velocity will rotate in the given time step is, from geometry, $\tan(\theta/2) = -(qB)m\Delta t/2$. The vector form of this is $\vec{t} \equiv -\hat{\vec{b}} \tan\theta/2 = (q\vec{B}/m)\Delta t/2$. The bisector vector v' is then

$$\vec{v}' = \vec{v}^- + \vec{v}^- \times \vec{t} \tag{3.16}$$

This $\vec{v}'$ vector is perpendicular to both the magnetic field (the vector $\vec{t}$) and the vector from $\vec{v}^-$ to $\vec{v}^+$, the post-rotation velocity we are looking for. This connecting vector is again obtained from geometry as the cross product of $\vec{v}'$ and a new vector $\vec{s}$. The vector $\vec{s}$ is just the rotation vector $\vec{t}$ scaled to satisfy the requirement that magnitude of velocity remains constant in rotation. Mathematically speaking

$$\vec{v}^+ = \vec{v}^- + \vec{v}' \times \vec{s} \tag{3.17}$$

where

$$\vec{s} = \frac{2\vec{t}}{1+t^2} \tag{3.18}$$

To implement the Boris method, the code first obtains $\vec{v}^-$ by adding half acceleration to the initial velocity, per Equation 3.13. The full rotation is the performed according to Equations 3.16 and 3.17. Finally, another half acceleration is added as given by Equation 3.17.

In the presence of a converging magnetic field, a magnetic mirror force, $F_M = -\mu_M(\partial B_s/\partial s)$ will arise retarding the motion of particles entering a region of an increasing magnetic field strength. The μ_M term in the above equation is the magnetic moment. It is given by $\mu_M = (1/2)mv_\perp^2/B$. If the mirror is particularly strong, the particle will reach a position where $v_\parallel = 0$ and the particle will be reflected. This is the basis for magnetic plasma confinement. In Hall thrusters, mirror arises in thrusters utilizing cusped field configurations, or near the centerpole of the CHT due to the geometry of the device.

The total energy of a particle in a magnetic field must be conserved, indicating that as the parallel component of velocity decreases, the tangential velocity component must increase to keep the total kinetic energy constant,

$$\frac{d}{dt}\left(\frac{1}{2}mv_{\parallel}^2 + \frac{1}{2}mv_{\perp}^2\right) = 0 \tag{3.19}$$

The Boris algorithm by itself does not resolve this effect. The reason is due to the fact that this relationship arises from the conservation of energy, while the Boris method is integrating the momentum equation. As such, it must be superimposed on the integrator. In the code, the $dv_M = (1/m)F_M\Delta t$ parallel component is first computed. The new perpendicular velocity is tcomputed from $v_{\perp}^2 = v_0^2 - (v_{\parallel} + dv_M)^2$. The x and y velocity components are then rescaled accordingly.

Particle position is updated using the Leapfrog method. This method was chosen over Runge-Kutta, as it provides a comparable accuracy at a fraction of the computational cost. We are integrating $d\vec{x}/dt = \vec{v}$. This expression can be easily integrated numerically if the velocity is offset by half Δt from the position. This is accomplished by integrating particle velocities by $-0.5\Delta t$ on loading. Then

$$\vec{x}^{n+1} = \vec{x}^{n} + \vec{v}^{n+1/2}\Delta t \tag{3.20}$$

where the velocity is integrated from $\vec{v}^{n-1/2}$ to $\vec{v}^{n+1/2}$ according to the Boris scheme discussed above. Of importance to axisymmetric simulations is the rotation of particles drifting about the thruster centerline. In the Δt time step, the crossed $E_{\perp} \times B$ fields will give rise to a $-E_{\perp}/B\Delta t$ shift in the y direction. Since in this code, the motion in the $x-y$ plane is decoupled from the motion in the s plane, this shift does not affect the subsequent computation. In other words, rotation due to the $E \times B$ drift is implied. However, for visualization of particle orbits about the thruster centerline, the code retains the angular position of the particle. The angle θ is incremented

according the azimuthal shift, $\theta = \tan^{-1}(\Delta y/r)$, where $r = r(s)$ is the radial position of the particle.

3.4.4 Wall Model

In a bounded discharge, majority of electrons is confined away from walls by the potential drop in the sheath. Only electrons with a sufficient energy to overcome the sheath drop will impact the wall. Electrons with insufficient energy to reach the wall will be reflected by the sheath. Due to surface roughness effects, the near wall sheath is expected to follow wall contours and be generally non-laminar. As such, it is possible that even electrons not actually reaching the wall will be diffusely scattered by the wall. This effect is not modeled in the present work and remains as a topic for future investigation. Instead, only particles actually impacting the wall are considered. In this work, the model of Sydorenko [36] was implemented. This model provides a curve fit to available experimental and analytic yields and takes into account elastic reflection, diffuse reflection, and yield of true secondary electrons. The secondary electron emission yield follows the analytical Vaughan fit.

The emission yields are plotted in Figure 3.8. This plot was generated by colliding 5,000 particles with a constant energy level W for each 1 eV increment in the 0 to 160 eV range. The wall interaction type flag set by the wall handler was used to count the number of interactions in each of the three possible categories (particles lost to the wall were not considered). It should be noted that for $W < 13$ eV the wall yield is dominated by the elastic reflection of the primary electrons. Even though these reflection conserve energy, they can contribute to transport if the reflected electron diffuses due to surface roughness. For $W > 30$ eV, the wall yield is dominated by the emission of secondary electrons, which exceeds unity at $W_1 = 52$ eV. High yield of secondary electrons can result in development of a non-planar saturated sheath which can trap emitted electrons in the near wall region. However, in the discharges studied

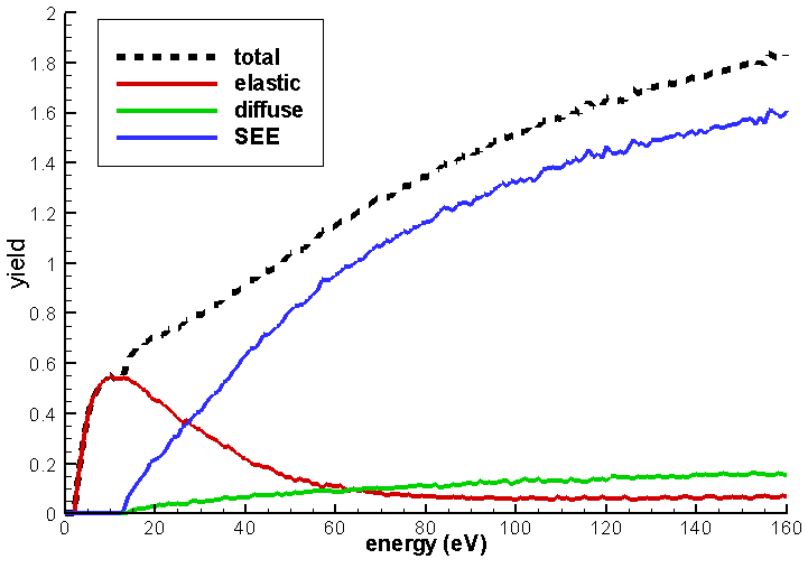


Figure 3.8: Variation in wall yield with incidence energy. This plot can be compared with Figure 3.4(a) in Sydorenko [36].

in this work, $ekT_e < W_1$. In addition to this realistic model, several wall models were developed for testing purposes. These additional wall types include completely absorbing walls, walls elastically and diffusely scattering all impacting particles, and a SEE injecting wall. This last wall type injects a specified $N_{see} = k_{see}N_{primary}$ number of secondary electrons for each incident primary electron.

Secondary electrons were injected into domain following the cosine distribution, $\vec{v} = v_M\hat{n}$ where

$$\hat{n}_0 = \sin(\theta)\cos(\xi) \tag{3.21}$$

$$\hat{n}_1 = \sin(\theta)\sin(\xi) \tag{3.22}$$

$$\hat{n}_2 = \cos(\theta) \tag{3.23}$$

with

$$\theta = \sin^{-1}(R_1) \tag{3.24}$$

$$\xi = 2\pi R_2 \tag{3.25}$$

and v_M is a velocity sampled from the Maxwellian distribution at the wall temperature. Maxwellian distribution was used instead of the perhaps more accurate effusion model for simplicity. R_1 and R_2 are two random numbers.

3.4.5 Particle Boundaries

Particles can leave the simulation domain in one of two ways: (a) by being lost to the walls or (b) by scattering sufficiently far from the magnetic field line. The first method is self-explanatory for a conducting wall. For the case of a dielectric wall, this loss corresponds to the electron becoming attached to the surface, and contributing to the surface charge density, σ. The details of current collection are outlined in the following section.

The second method is merely a computational tool used to repopulated the initial velocity distribution function. In a real device, electrons are continuously traversing from one magnetic field line to another until they arrive at the anode. If the distance between field lines is taken to be r_L, then the energy an electron gains in the transport is given by $W = E_\perp r_L$. For the Hall thruster discharge, this energy gain is on the order of one to several tens of eV and corresponds to the initial electron temperature at the field line. It should be pointed out here that a magnetic field line is merely a convenient mathematical construct, in reality the magnetic field is continuous. If an electron guiding center diffuses a significant distance from the starting position, it can be assumed to no longer correctly represent the population at the studied location. The size of the bounding envelope is somewhat arbitrary and can be set by the user. The size is set as $d = fr_{L,max}$ where $r_{L,max}$ is the Larmor radius computed at the smallest value of magnetic field. Although an argument could be made for using $f = 1$, such a value will immediately (assuming uniform B) remove all particles undergoing a scattering event. As such, f needs to be set to a sufficiently large (i.e. $f > 1$) value to allow secondary electrons to complete at least one oscillation between

walls, and allow colliding electrons enter the sheath. Simulations presented in this work used $f = 5$.

3.4.6 Current Collection

Surface charge σ collected on the walls of a dielectric material contributes to the boundary conditions in the potential equation. The wall current at each wall consists of three terms: $j = j_i - j_e + j_s$ where the terms on the right hand side are the ion, primary electrons, and secondary electron current densities. In this formulation no distinction is made between true secondaries and reflected primary electrons. For a mirror-like wall, $j_e = j_s$ and the contributions cancel. The ion current density can be estimated from Bohm velocity. Note, this assumes that the sheath has not inverted; and that ions are entering the sheath in direction normal to the wall with the Bohm speed. As indicated in Chapter 6, this is often not the case, especially once novel thruster configurations are considered. SEE yield greater than unity can also result in an inverted sheath. Taking sheath inversion and sheath collapse into account remains as future work. In the simple Bohm model, the ion term becomes $j_i = en_0u_i$. For simplicity, $n_0 = \overline{n_i}$, the average ion density, is used. Again, determination of the actual sheath density remains as future work.

The electron terms are obtained from the kinetic electrons. In the code, the one dimensional volume scaling is used, $n = N/\Delta s$ in the bulk and $n = N/0.5\Delta s$ along the boundaries. $N = w_{sp}$ is the number of physical electrons carried by each electron macroparticle. Electron current density then becomes $j_e = 2ew_{sp}/\Delta s\vec{u} \cdot \hat{n}$. Similar expressions are written for the reflected and secondary populations. Surface charge collection begins at steady state and is incremented at each time step from $\sigma = \sigma + j\Delta$.

3.5 Collisions

Collisions are an important process affecting the diffusion of electrons. The collision types considered in the code are momentum transfer, ionization, excitation, and Coulomb collisions. The Monte Carlo Method (MCC) is used to determine collision probability. MCC is a fast method that is applicable in situations where the target population is not significantly affected by the collision events. MCC works by iterating through the list of source particles, and for each particle computing the collision probability. The probability is given by $P = 1 - \exp(-n_0 \sigma_0 g \Delta t_c)$. Here σ_0 is the total collision cross-section, n_0 is the density of the target gas, g is the relative velocity (due to the high velocity of electrons, $g = v_e$), and Δt_c is the time difference between collisions. Typically, collisions are computed once every four time steps, $\Delta t_c = 4\Delta t$ for performance and statistical error reduction reasons.

The collision probability is compared to a random number R. If $P \geq R$, a collision occurs. Next, the code needs to determine which collision type occurred. This is done by calculating relative collision cross-sections, $\sum_{i=1}^{types} \sigma_i/\sigma_0 = 1$. The respective σ_i ranges are stored in an array such that $(\sigma_{i-1} : \sigma_{i-1} + \sigma_i)_i$. A second random number is then drawn. Location of this random number in the array of relative sigmas is then determined, and the collision handler for the respective process is called. It should be noted that the collision handler operates solely on the particle velocities. Particle positions are not affected by the collision. Cross-sections for the considered collision types are plotted in Figure 3.9.

MCC differs from Direct Simulation Monte Carlo (DSMC) in that it collides a particle with a fixed background cloud. In DSMC, particle pairs are selected, requiring the target population to be treated kinetically. The downside of the MCC method is that unless corrections are made to the target population, energy and momentum of the system will not be conserved. For majority of collision types, electrons are collided with massive neutral particles, with the neutral density exceeding plasma

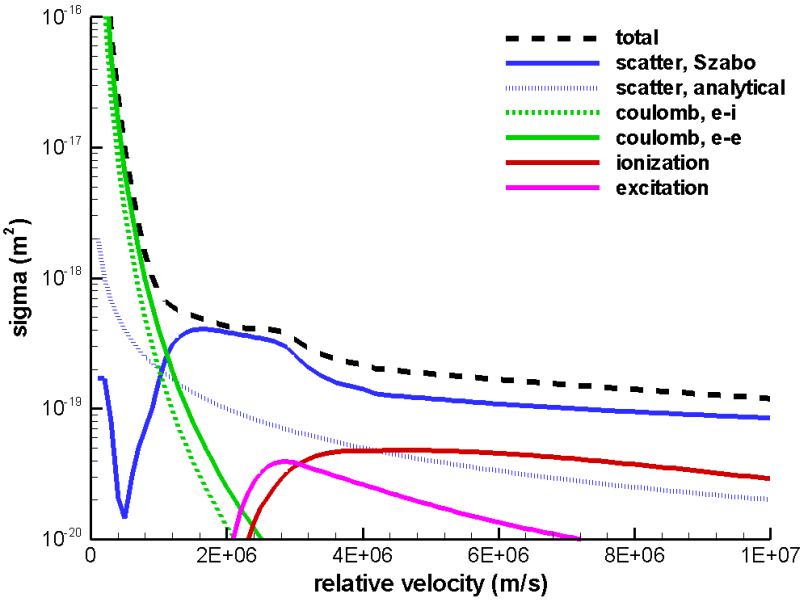


Figure 3.9: Collision cross-sections vs. energy. Scatter, excitation, and ionization models from Szabo [21].

density be several orders of magnitude. As such, collisions have a negligible effect on the target population and the MCC method is applicable. The exception are electron-electron Coulomb collisions. Proper treatment of these interactions would require implementation of the DSMC method. Here however a simplified approach is taken. As described below, a modified MCC algorithm is developed for the electron Coulomb collisions. In this algorithm, collision probability is selected from the MCC method, but the collisional event is performed by sampling a target particle from the electrons occupying the same computational cell as the source particle.

3.5.1 Momentum Transfer Collisions

An important collision process in Hall thruster discharges are the momentum transfer collisions between electron and neutral atoms. These collisions are due to electrostatic interaction between the electron and the atom, where the interaction forces arises from the polarization of the atom. An analytical model for polarization

scattering is given by Lieberman [58],

$$\sigma_L = \pi b_L^2 = \left(\frac{\pi \alpha_p q^2}{\epsilon_0 m_R}\right)^{1/2} \frac{1}{g} \tag{3.26}$$

where $\alpha_p = \alpha_R a_0^3$ is the polarizability (in units of volume), α_R is a proportionality constant, and $a_0 = 53$ pm is the Bohr radius. The reduced mass m_R is given by $m_R = m_e m_a/(m_e + m_a) \approx m_e$ for electron-atom collisions. The relative velocity $g = |v_e - v_a| \approx |v_e|$. The relative proportionality is obtained using available Xenon polarization data, which list $\alpha_p = 4$ Å^3, resulting in $\alpha_R = 26.9$. However, when this model is compared to experimental measurements, it is seen to underpredict the collision cross-section by approximately half order of magnitude except in the low energy region, where it dominates over the experimental fit. The fit to experimental data, as utilized by Szabo [21] is seen in Figure 3.9 in blue. The analytical polarization profile is shown with the dotted blue curve. For this reason, Szabo's model is used in this work to compute the cross-section. The expression for the curve is available in the Appendix in [21].

Electron scatter collisions are modeled by colliding the electron with a virtual target neutral sampled from the background population at the background temperature. The algorithm is based on the approach described by Bird [59]. The algorithm first computes the center of mass and a random deflection angle. The post-collision electron velocity is then updated from

$$\vec{v}^* = \vec{v}_m + \frac{m_2}{m_1 + m_2} g^* \tag{3.27}$$

where $\vec{v}_m$ is the center of mass velocity and g^* is the post collision relative velocity vector. It should be noted that the magnitude of g is preserved in the collision, $|g| = |g^*|$.

3.5.2 Ionization and Excitation Collisions

Szabo's models (as listed in the Appendix in [21]) are also used for the ionization and excitation collisions. As an example, the cross-section for the excitation collision is given by

$$\sigma_{ex} = \begin{cases} 0 & W \leq W_{ex} \\ 10^{-12}(0.00194724369808W^2 - 0.02261576374741W^{1.5} & \\ +0.09807793114366W - 0.18808539260191W^{0.5} & \\ +0.13446494003922) & W \leq 11 \\ 10^{-13}(0.00069390658261W^2 - 0.01241570210985W^{1.5} & \\ +0.08109737428153W - 0.22730324307635W^{0.5} & \\ +0.23122639784590) & W \leq 25 \\ 10^{-14}(0.00000121267639W^2 - 0.00008169557347W^{1.5} & \\ +0.00207211887803W - 0.02409700583197W^{0.5} & \\ +0.11701534311188) & W \leq 500 \\ 3.95 \times 10^{-17} & W > 500 \end{cases} \tag{3.28}$$

An analogous model exists for the ionization event. These two cross-section are plotted in Figure 3.9 with the red and purple lines. As can be seen from this plot, these collisions become dominant only at high energies. This is due to the fact that the impacting electron needs to impart $\mathcal{E}_{iz}$ in order to remove the valence electron for the ionization event to occur. Hence, ionization occurs only for source electrons with $\mathcal{E} > \mathcal{E}_{iz}$. The code does not consider creation of new electron/ion pairs. Instead, both of these events only acts as an energy sink for the impacting electrons. The energy of an electron undergoing an ionization collision is reduced by W_{ionize}, while the

energy in the excitation collision is reduced by $W_{ex} = RW_{ionize}$ where R is a random number. In both events, the original direction of the velocity vector is preserved. These collision events are thus assumed to only operate on the velocity magnitude.

3.5.3 Coulomb Collisions

As indicated by Figure 3.9, Coulomb collisions are an important process at low energies. Unlike the previously considered collisions, Coulomb collisions arise from the electrostatic attraction or repulsion between charged particles and thus the interaction force magnitude scales with the inverse of the distance. Without considering an upper bound, these collisions would result in an infinitely large cross-section. For this reason, λ_D is selected as the typical upper bound over which the interaction force is integrated. This then leads to a very large cross-section, $\sigma \sim \pi\lambda_D^2$. However, vast majority of Coulomb collisions are small angle events. As such, a common approach is to approximate the large number of small angle collisions with a corresponding large angle cross-section. Lieberman [58] gives the large-angle cross-section as

$$\sigma_{90} = \frac{8}{\pi} b_0^2 \ln \Lambda \tag{3.29}$$

where $\ln \Lambda \approx 10$ for typical plasma discharges and b_0 is the distance of closes approach, given by

$$b_0 = \frac{q_1}{q_2} 4\pi\epsilon_0 W_R \tag{3.30}$$

where $W_R = \frac{1}{2} m_R v_R^2$ is the center of mass kinetic energy.

Since this cross section approaches infinity as $v_r \to 0$, the code limits minimum v_r to 10^4 m/s. Electron-ion Coulomb collisions are modeled using the same algorithm utilized for electron-neutral momentum transfer collisions, where a virtual target ion is sampled from the background population. This approach is not valid for electron-electron interactions. Electron-electron Coulomb interactions are the primary mean

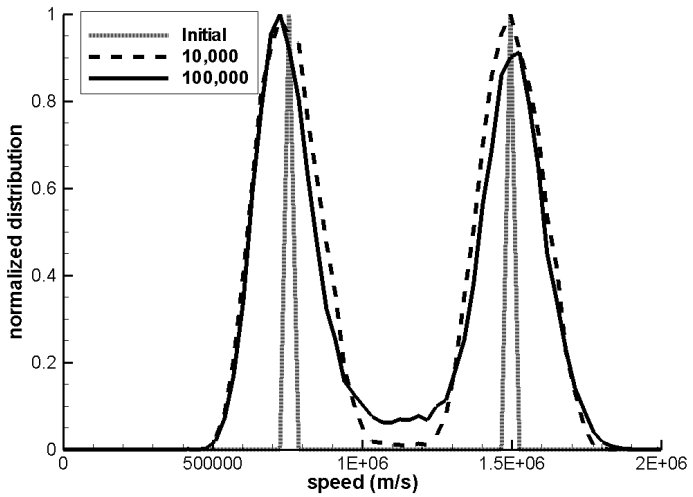
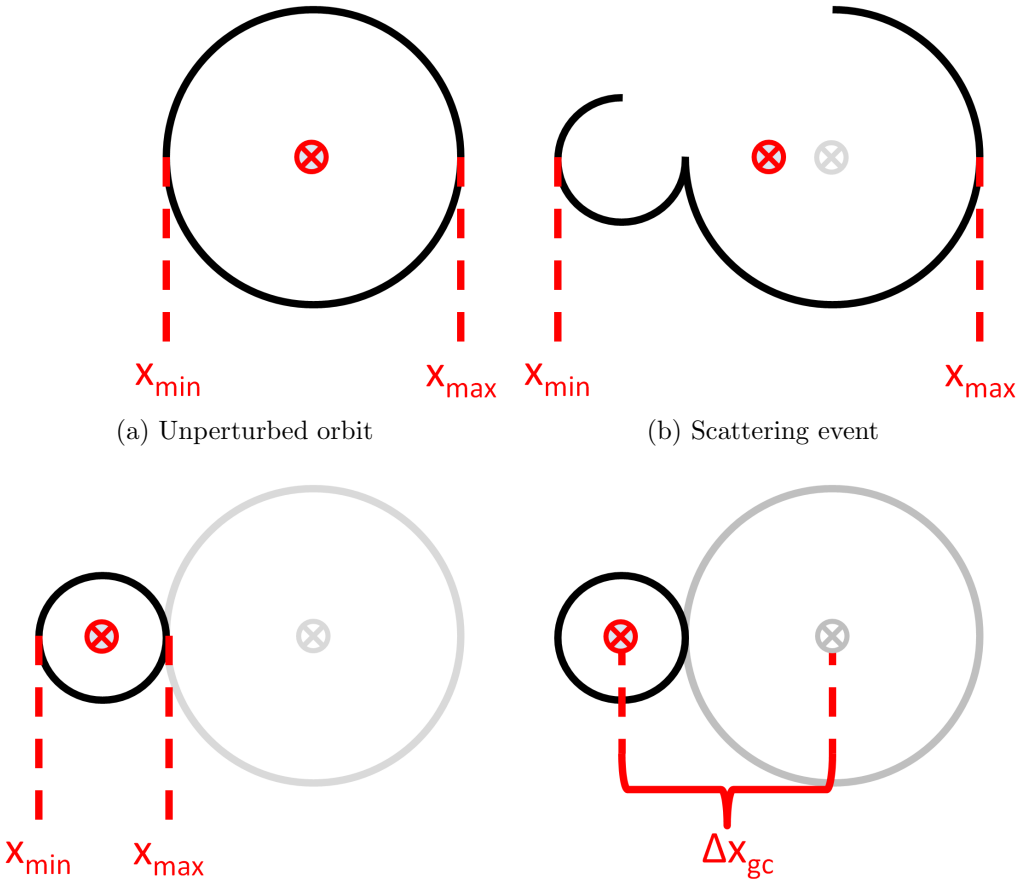


Figure 3.10: Velocity distribution function at three different times showing the thermalization of an initial beam-like population used for code testing

by which electron velocity distribution functions thermalize and hence it is important to consider both source and target particles in the collision event. A modified MCC algorithm was implemented to handle this interaction event. Prior to computing collision probabilities, electrons are grouped by cell. Next, during a collision event, a target electron is picked randomly from the electrons located in the same cell as the source, assuring the target particle is different from the source. Collision is then performed with these two particles, and velocities of source and target are updated. Since this effectively results in a doubling of collisional rate due to the MCC algorithm, the collision cross-section for electron-electron collisions is reduced by half, $b_{0,eff} = 0.5b_0$. The effect of Coulomb collisions is demonstrated in Figure 3.10. These results were obtained by loading two cold electron populations with $v_\perp = 10^5$ m/s, $v_{\parallel 1} = 0.75 \times 10^6$ m/s, and $v_{\parallel 2} = 1.25 \times 10^6$ m/s. The two populations are seen to form a double-Maxwellian distribution in the first 10,000 time steps. Additional thermalization continues, however at a reduced rate. This reduction is related to the preference of Coulomb collisions to occur between particles of similar speed due to the $1/g$ dependence in σ.



(a) Unperturbed orbit (b) Scattering event

(c) Final orbit (d) Guiding center shift

Figure 3.11: Schematic of guiding center shift due to scattering events.

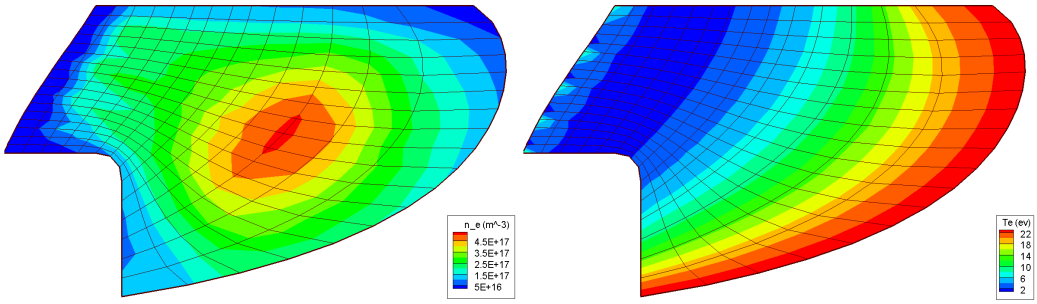
3.6 Transport Calculation

The primary output from a Lynx simulation is the spatial variation in mobility along the magnetic field line, $\mu = \mu(s)$. Conceptually, the average drift velocity can be computed by averaging the particle perpendicular component of velocity. Mobility can then be determined from $\mu = v_d/E_\perp$. However, such an approach does not work in practice due to statistical errors. The simulation noise in the computation of the average drift velocity is on the order of μ. This noise arises due to the discretized nature of the particle push, the particle traverse along the field line, and the averaging of velocities at nodal positions.

Instead, the drift velocity is computed from the speed with which guiding centers

diffuse across the field line. This concept is shown schematically in Figure 3.11. The x position of the guiding center, the point about which the particle orbits in the $x-y$ plane, center can be computed from $x_{gc} = 0.5(x_{min}+x_{max})$, where the two positions on the right hand side are the minimum and maximum x position attained by the particle. These values are continuously updated by the particle integrator. Scattering events, such as collisions or wall interactions, result in the particle orbit shifting to a new guiding center, which will be demonstrated by a shift in the x range. Guiding center positions are checked once per orbit, and only particles that have completed at least one orbit since birth are considered. Particle x_{min} and x_{max} values are reset during this operation. As can be seen from Figure 3.11, particles undergoing a transport event in the previous cyclotron orbit will have a guiding center intermediate of the initial and the final position. The correct shift will be computed upon the subsequent calculation (assuming no additional scattering events) due to particle x range reset. Once the position of the guiding center is known, the drift speed is obtained from $v_{gc} = \Delta x_{gc}/(t_{now} - t_{born})$. The term on the right hand side is the time the particle has been alive. Since in the code the initial $x_{gc} = 0$, the shift is given by the actual guiding center position. Mobility is then computed from $\mu = v_{gc}/E_{\perp}$.

Numerically, this approach is implemented by looping through the particles and for each particle determining the drift velocity. Two valid approaches exist for computing mobility, both of which are implemented in the code. In the first approach, the drift velocity is interpolated to the grid nodes along with a particle counter entry (value of 1). These parameters are accumulated for several orbits, according to a user defined parameter. The average drift velocity at each node is then computed from $v_{gc,j} = \sum v_{gc,j}/c_j$, where c_j is the value of the counter field at node j. Mobility is computed from $\mu_j = v_{gc,j}/E_{\perp,j}$. In this approach, an average drift velocity is determined first and the node value of electric field is used. In the second approach, mobility is computed directly at the particle location, $\mu(s) = v_{gc}(s)/E_{\perp}(s)$. This



(a) Plasma Density (b) Electron Temperature

Figure 3.12: Plot of typical simulation inputs

mobility value is then scatter to the grid, and the final value is obtained from the average, $\mu_j = \sum \mu_j / c_j$. As shown later, these two methods produce values with an excellent agreement.

3.7 Implementation

Lynx is implemented in the Java programming language. Historically, Java was not a viable option for scientific computing due to a reduced performance compared to programming languages such as Fortran or C/C++. However, improvements in compiler technology, and automatic optimization of the byte code for the particular architecture has allowed Java to match, and in some cases even exceed, the performance of these more classical languages. Java is a very attractive programming language for code development since it includes a mature implementation of the object-oriented design. In addition, Java includes a large native library providing support for features that are not inherently available in C++ and require the use of various external libraries (such as Boost++). Examples of such features are multi-threading and a graphical user interface.

In order to analyze a typical 2D fuid model, a number of magnetic field lines corresponding to the resolution of the mesh used by the electron solver must be analyzed. Figure 3.12 shows the typical simulation inputs to Lynx. This mesh,

corresponding to the magnetic topology for the CHT, consists of 15 magnetic field lines. Plasma density and electron temperature is shown here, with additional data including the magnetic field strength, neutral density, and the normal electric field. A unique Lynx simulation is performed for each magnetic field line. Modern computer architectures offer multi-cored processors, allowing multiple computational threads to execute in parallel. This is the approach taken in Lynx. Instead of parallelizing the actual 1D computation, speed up is obtained by running multiple simulations concurrently. For instance, the results presented in Chapter 5 were obtained on a Windows desktop workstation containing dual quad-core Xeon processors. This allowed for up to 8 magnetic field lines to be studied in parallel. Since this approach does not require any data packing and transfer that is necessary in coarse-grained MPI applications, the code runs effectively at a 100% parallel efficiency.

Adding multi-threading support to a Java application is trivial. The actual simulation main loop is implemented in a `Simulation` class. The required steps were: (a) inheriting from a parent `Thread` class and (b) defining a public void method `run()` that is executed automatically once the thread is created. In comparison, implementing multi-threading in a C++ code would require either downloading and configuring the Boost++ libraries, or relying on the OpenMP pragmas to vectorize loop operations. The second approach would result only in a limited multi-threading support. In a multi-threaded program, care must be taken to avoid a locking situation, where a thread suspends an operation waiting on another thread to finish some operation. Such a situation arises when asynchronous methods are used. One such a method is the native implementation of `Math.random`. Since only one call can be made to this function at any given time, calls from multiple threads will be evaluated in sequence, completely destroying any parallel speedup. To alleviate this issue, each simulation thread simply needs to implement its own generator by instantiating a copy of the base `Random` object.

Additionally, since typically the number of magnetic field lines is greater than the number of available CPUs, a simple scheduler was implemented to launch threads as CPU resources become available. The simulation cases are stored in an `ArrayList` of `Simulation` objects. The scheduler determines the number of running threads once per second by looping through the list and counting number of cases with a `running` flag set to true. This number is compared to the number of available CPUs. If a CPU is available, the next thread in sequence is launched. During the initial pass the scheduler also checks for any threads that have finished. The result from that particular field line is interpolated onto the input 2D mesh (i.e., one vertical line in Figure 3.12 is populated and the corresponding `Simulation` object is destroyed to free up memory space.

The 2D mesh containing inputs from the global discharge is read in the Tecplot format using a class called `Reader`. Another class `Settings` processes the user input file. This file, contains parameters controlling the entire simulation sequence that are independent of the magnetic field line. An example of the settings file is shown below.

```
#version info
VER 2.1

# domain: input file, 1D per cell, solver mode, wall type
DOMAIN mesh2d.dat 2 CYLINDRICAL DIELECTRIC

#time steps, steps per orbit, steady state min, averaging delta
TIME_STEP 30000 75 50 2000

#frequency of trace, system, field saves
OUTPUT 20 100 1000

#count (not for ions), mass (AMU), charge (el.unit)
ELECTRONS 100000 5.48e-4 -1
IONS 131.3 1

#wall model, roughness, SEE multiplier, wall temp
WALLS SYD 0.5 1 1000
```

```
# collisions: true/false/[const sigma]
COLLISIONS true

# wave: a b w p: a*cos(wt+p)+b
WAVE 0 1 4e5 0
```

Chapter 4

Code Validation and Mobility Studies

4.1 Introduction

This chapter introduces results from studies undertaken to validate the kinetic code. Several unit tests were developed and integrated within the development environment. NetBeans, the development environment used in this work, has a built-in support for unit testing. The tests included tests of particle motion, energy conservation, wall interactions, particle loading, collisions, potential solver, and transport calculations. The code was subsequently used to model a simple academic configuration consisting of a vertical field line in a planar topology. This study was undertaken to investigate the response of electrons to a variation in various simulation parameters, such as the wall model, magnetic mirror, and electron temperature.

4.2 Unit Tests

4.2.1 Single Particle Motion

The single particle motion test is the most fundamental test, since it investigates the ability of the integrator to correctly resolve the closed cyclotron orbit. In this configuration collisions were ignored and the particle did not move in the parallel direction. Objective of the test was to verify energy conservation, as well as the

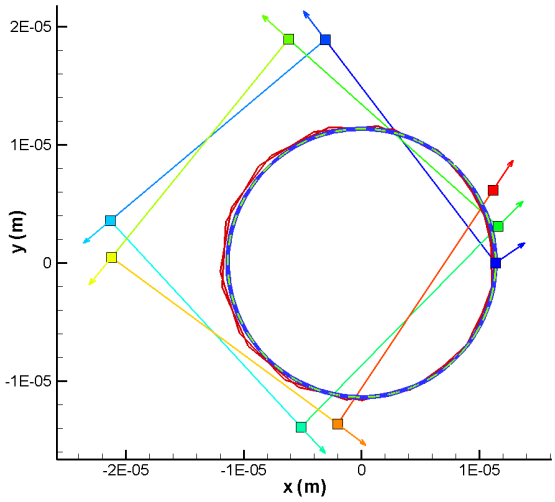


Figure 4.1: Comparison of particle traces after two orbits for several different value of it_{orbit}: 4 (multicolor), 14 (red), 140 (blue) and 1400 (dashed green)

correct placement on $x_{gc} = 0$ with the correct Larmor radius, r_L. A single particle was loaded into the simulation with $E_{\|} = 0$, $v_{\|} = 0$, $B = 0.1$T, and $v_{\perp} = 10^5$ m/s. Under these conditions, a magnetized particle will complete a closed orbit about the magnetic field line. The radius of the orbit is given by the Larmor radius, $r_L = \dfrac{mv_{\perp}}{|q|B}$. In the unit test, the particle position was integrated for 20 orbits. The particle traces after 2 orbits are shown in Figure 4.1. The four curves correspond to results obtained with four different values of Δt. The multi-colored trace corresponds to a particularly large value in which an orbit is completed in only 4 time steps. At this large value we can see that both the orbit radius and the origin are incorrect. A marked improvement is seen by increasing the number of steps taken to complete the orbit to just 14. This case is shown in red. The two remaining traces shown in blue and dashed green show the solution computed with 140 and 1,400 steps, respectively. These two traces are effectively identical. In addition, all cases result in a closed orbit, confirming energy conservation. A non-conserving integration method, such us the simple forward method, would result in the particle following a spiraling path.

The solutions were characterized numerically by considering the centroid $x_o = 0.5(x_{min} + x_{max})$ and the radius $r = 0.5(x_{max} - x_{min})$ and comparing them to the

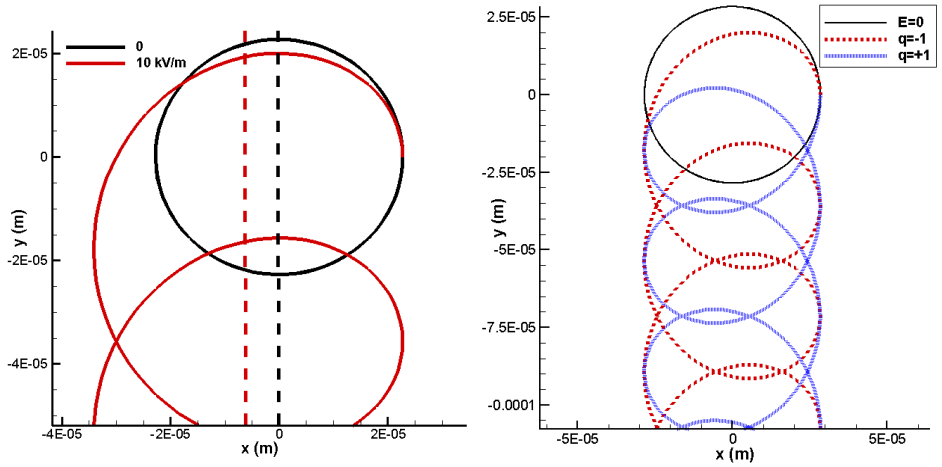
Table 4.1: Summary of errors for a single particle motion

it_{orbit}	E_o (%)	E_r (%)
4	35.5	44.9
14	0.54	2.88
75	0.0155	0.0879
140	0.00647	0.0255
1400	0.00387	0.000253

analytical values. The error in the origin was computed relative to the particle orbit as $E_o = |x_o/r|$. The error in radius was computed relative to the analytical r_L as $E_r = |(r - r_L)/r_L|$. Errors after 20 orbits are summarized in Table 4.1. This table also includes the case $it_{orbit} = 75$ corresponding to the typical setting used in subsequent simulations. As can be seen from this table, at $it_{orbit} = 75$, both errors are less than 1%. This time step generally offers a good compromise between accuracy and computational performance.

4.2.1.1 $E \times B$ drift

The $E \times B$ test is similar to the Larmor radius test in Section 4.2.1 with the addition of a fixed $E_\perp$ term. The objective of this test was to verify the correct loading of particles in the presence of a finite $E_\perp$. Figure 4.2 shows the computed particle traces. The black circle is a reference case with $E_\perp = 0$. In this setup, $E_\perp = 10,000$ V/m, $B = 0.01$ T (pointing out of the page) and $v_\perp = 5 \times 10^5$ m/s. The first picture shows the impact of a simplistic particle loading. If particles are injected by simply rotating $v_\perp$ by a random angle without considering the velocity shift that occurs in the presence of an applied electric field, E/B, the particle will describe a circle with an incorrect Larmor radius and a finite shift in the guiding center. This particle orbit is shown by the red trace and the dashed line indicates the shift in the guiding center. This numerical drift would then induce a finite amount of non-physical transport. The second figure shows the correct loading scheme in which the drift velocity is taken



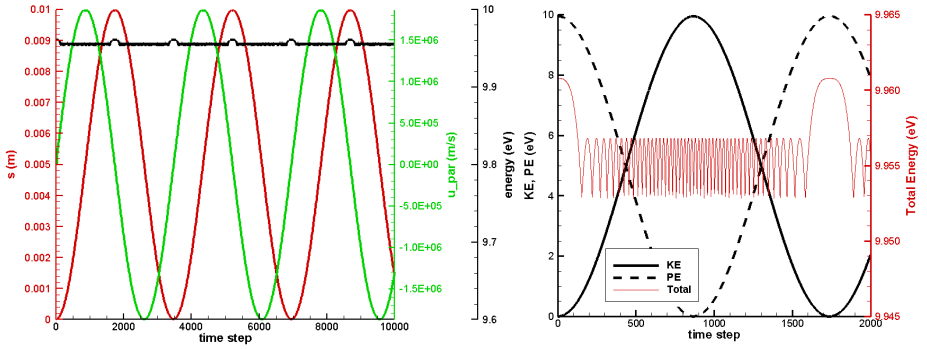
(a) Incorrect Loading (b) Correct Loading

Figure 4.2: Particle trace with simple and improved loading in the presence of $E_\perp$

into account. This trace corresponds to the scheme described in Section 3.4.2. Again, the black circle shows the reference orbit, and the two colored traces correspond to an electron and a matching positron. Although positively charged particles are not used in the simulation, this test verifies the correct implementation of the integrator. Both particles are seen to drift in the same direction, agreeing with the theoretical observation that the $\vec{E} \times \vec{B}$ drift is independent of particle charge. The pass criterion for this test was identical to the previous case, E_o and $E_r < 1\%$

4.2.1.2 Particle in a Potential Well

Energy conservation in the direction parallel to the field line was tested by loading a single electron into a potential well. The well was modeled by the quadratic function $\phi = \phi_M \left[1 - 4\left(s/L - 0.5\right)^2\right]$ with $\phi_M = 10$ V. The electron was loaded near the inner wall at $s = 0.001L$ and let to oscillate for 100,000 time steps. Results for the first 10,000 steps are shown in Figure 4.3. These results are typical of the entire sequence. The particle is seen to follow an oscillatory path with no change in the minimum and maximum position. This is the predicted behavior for an energy conserving system. In the unit test, energy conservation was checked at each time step by computing



(a) Particle trace (b) Close up of energy trace

Figure 4.3: Plot of a particle trace for a particle in a potential well. Figure (b) shows a close up of the total energy.

the kinetic energy $W = 0.5mv_{\|}^2$ and the potential energy $U = |q|\,(\phi_m - \phi)$. The test pass criterion was set as $E = |(E - E_0)/E_0| < 0.01$, where $E = W + U$ is the total energy. This total energy is also plotted in Figure 4.3. As can be seen from this plot, there is a small periodic deviation from a constant value around the inflection points, $v_{\|} = 0$. Upon a closer inspection, it can be seen that these "dimples" correspond to the correct system energy, and the flat profile shows a slight difference from the expected value. This deviation at positions where $v_{\|} \neq 0$ seems to be an artifact of the characterization method in the unit test and not an actual error in the integrator. In the leapfrog method, velocity and position are known at times offset by $0.5\Delta t$. However, in order to compare the energies, values at the same temporal point need to be known. In the unit test, a simple method is used to accomplish this. The unit test algorithm follows

```
xm = part.x
IntegrateVelocity(part)
MoveParticle(part)
xp = part.x
```

The potential energy was evaluated at $x = 0.5(xp + xm)$. Due to a non-uniform velocity $v = v(x)$ dependence in the quadratic field, this approach results in a discrepancy

in computing the value of $x^{k-1/2}$. The close up of the energy trace is shown in Figure 4.3b. It should be noted this error disappears when a linear potential profile is used.

4.2.1.3 Magnetic mirror

The ability of the integrator to resolve a magnetic mirror was tested by loading a particle into a linearly varying magnetic field. The magnetic field profile was given by $B = B_0(1 + (s/L)(R_m - 1)$. The linear profile results in a constant acceleration that can be easily integrated to obtain an analytical relationship for particle motion. From theory, the force acting on a particle is $F_\parallel = -\mu \partial B/\partial s$. Given a constant gradient, the velocity and position can be evaluated,

$$dv = -\frac{\mu_M}{m}\frac{dB}{ds}dt \tag{4.1}$$

$$w = -\frac{\mu_M}{m}\frac{dB}{ds}t + w_0 \tag{4.2}$$

$$s = -\frac{1}{2}\frac{\mu_M}{m}\frac{dB}{ds}t^2 + v_0 t + s_0 \tag{4.3}$$

where $w = v_\parallel$ and s is the distance along the field line. Taking the initial particle position $s_0 = 0$, and placing the wall at $s = s_w$, the maximum velocity for a confined particle can be determined from $1/2(\mu_M/m)\frac{dB}{ds}t^2 - w_0 t + s_w = 0$. The solution of this equation is given by the quadratic equation with the determinant

$$D = w_0^2 - 4\frac{1}{2}\frac{\mu_M}{m}\frac{dB}{ds}s_w \tag{4.4}$$

No real solution exists for particles impacting the wall prior to reflecting, and hence the threshold velocity is

$$w_{0,max} = \sqrt{2\frac{\mu_M}{m}\frac{dB}{ds}s_w} \tag{4.5}$$

For $s_w = 0.01$ m and $R_m = 30$, $w_{0,max} = 538,516$ m/s. Figure 4.4 shows the trace of a particle with initial velocity just slightly below this threshold, $w_0 = 538,000$ m/s.

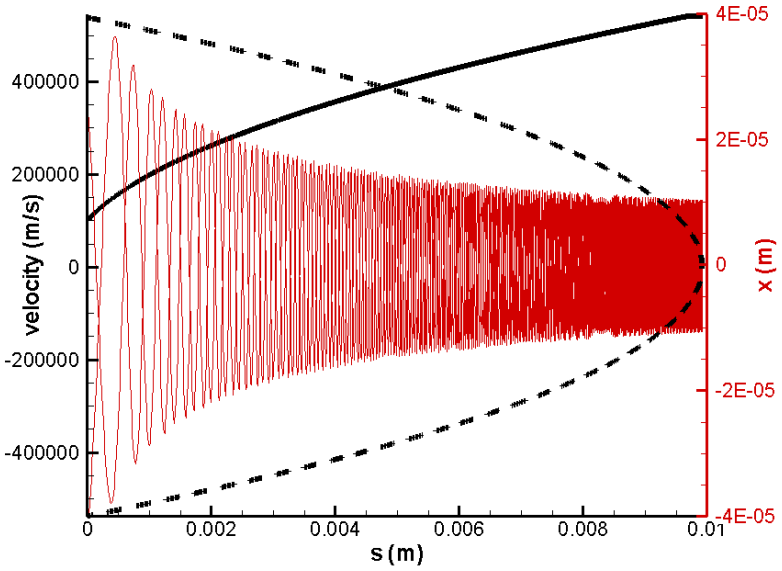
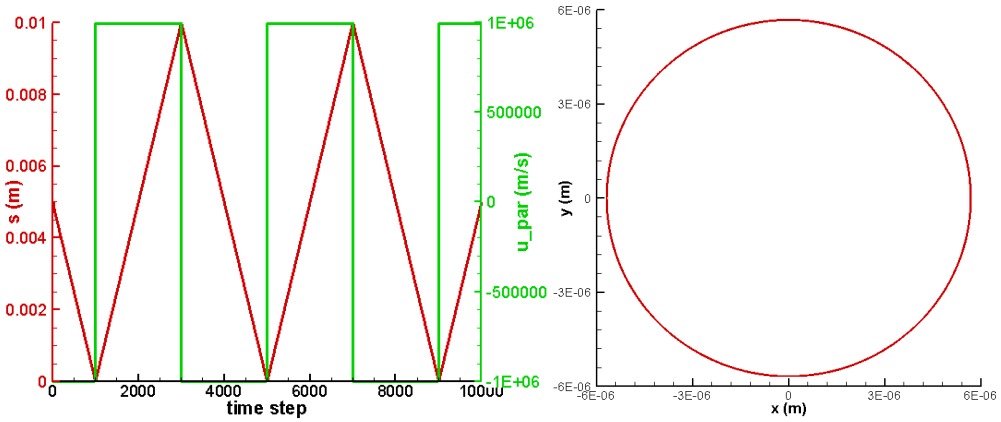


Figure 4.4: Particle trace for an electron reflecting in a linearly strengthening magnetic field. The solid and dashed line correspond to the $v_\perp$ and $v_\parallel$ component of velocity.

The potential solver was not used in this unit test, hence $E_\parallel = 0$. The particle can be seen to approach close to the wall prior to reflecting in the magnetic mirror. The two black curves show the particle velocity components. The perpendicular component is seen to increase in response to the decrease in $v_\parallel$, keeping the total kinetic energy constant. It should also be pointed out that due to the decrease in velocity, the particle spends an increased amount of time in the mirror where $v_\parallel \sim 0$. This effect could be of importance to electron transport. Electron density in the mirror region may not be necessarily reduced due to the decrease in electron velocity. Instead, the mirror appears to primarily be responsible for converting the parallel velocity component in the perpendicular, resulting in a local strongly anisotropic electron population, $kT_{e,\parallel} \ll kT_{e,\perp}$.

4.2.2 Wall Interactions

Wall interactions are an important contributor to electron transport in Hall thruster. However, an incorrect treatment of electron reflection at the wall can lead to a



(a) Position and Velocity (b) Particle Trace in x-y plane

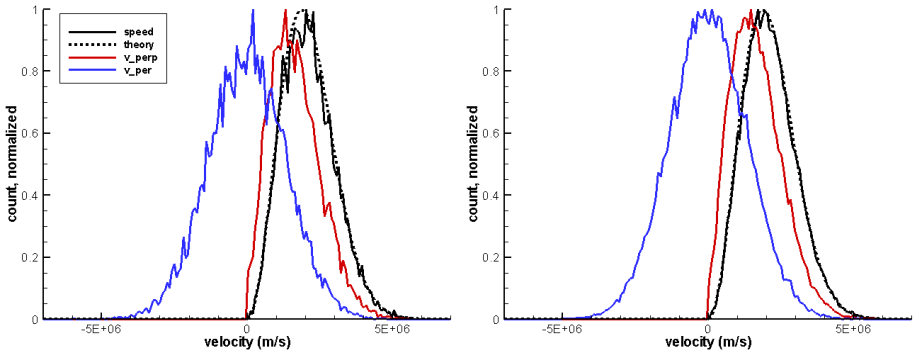
Figure 4.5: Trace of a particle bouncing between two completely elastic walls

numerically-driven diffusion. To verify the proper wall treatment, a single particle was loaded into a domain bounded by specular elastic walls. No electric field was present and collisions were ignored. In such a configuration, the electron is expected to complete close cyclotron orbits with no deviation in the guiding center origin. Since the wall is also elastic, the magnitude of $v_{\|}$ is expected to remain constant. The velocity components used in the test were $v_{\|} = 10^6$ m/s and $v_{\perp} = 10^5$ m/s. The time step $\Delta t = 5 \times 10^{-12}$ s and the spacing between the walls is 0.01 m. This time step is comparable to the time step used in the simulation with $it_{orbit} = 75$. For this particular configuration, the electron is expected to collide with the wall every 2000 time steps. Results from this unit test are shown in Figure 4.5. As can be seen from the figure, the electron impacts the wall every 2000 time steps as predicted, and the velocity magnitude remains invariant. In addition, Figure 4.5b shows a "top-down" view at the orbital plane. The particle is seen to trace out a closed circle with no shift in x_{gc} or r. Non-specular reflection would be demonstrated by a departure from this circular trace.

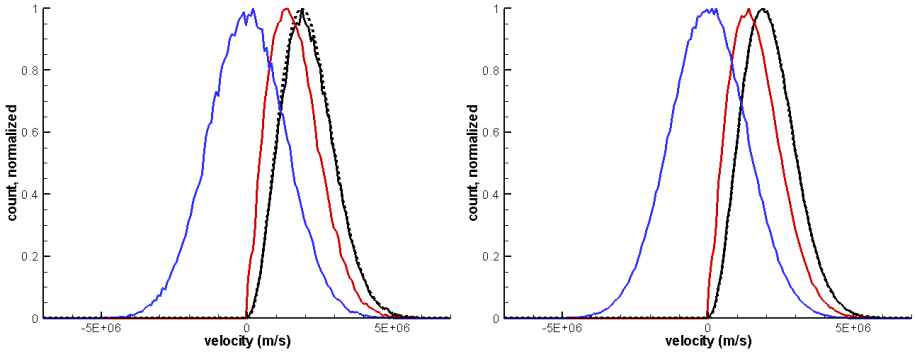
4.2.3 Particle Loading

The next set of tests investigated effects related to the entire electron population. The first of these was a test of the particle sampling algorithm. This test consisted of two parts, the ability of the sampling routing to capture the prescribed ion density profile and the ability to resolve the prescribed velocity distribution function. The result from the first test was presented earlier and can be seen in Figure 3.3. The ability of the particle creation algorithm to capture the prescribed velocity distribution function was tested by loading a predefined number of computational particles and comparing the numerically obtained velocity distribution function with theory. These results are summarized in Figure 4.6. This figure illustrates the effect the number of particles plays on the accuracy with which the VDF is resolved. Since in the PIC method, each particle corresponds to a volume in the 6-dimension velocity space of the Boltzmann equation, increasing the particle number improves the accuracy with which the velocity distribution is captured. This is clearly seen in these results. A good agreement is obtained using 100,000 particles, while 500,000 particles produce distribution almost indistinguishable from theory on the scale of the graph.

Figure 4.7 focuses on the high-energy tail of the distribution function. The NWC theory supposes that additional electron transport is driven by wall fluxes of high energy particles penetrating the sheath. Hence, it is imperative that the loading scheme captures this high-energy region. Unfortunately, as can be seen from this plot, increasing the number of particles has only a marginal effect on the tail. With 10,000 particles, the highest energy particles have velocity of approximately 5×10^6 m/s. In order to increase the maximum velocity just 40% to 7×10^7 m/s, the number of particles must be increased 500 times to 5 million. Such a high number of particles in numerically ineffective. Instead, a variable weight model should be used, in which a lower weight is assigned to high energy particles. One such a model was used in the work of Fox [56]. Implementation of this model is left for future work.



(a) 10,000 (b) 50,000



(c) 100,000 (d) 500,000

Figure 4.6: Velocity distribution functions obtained with different number of computational particles

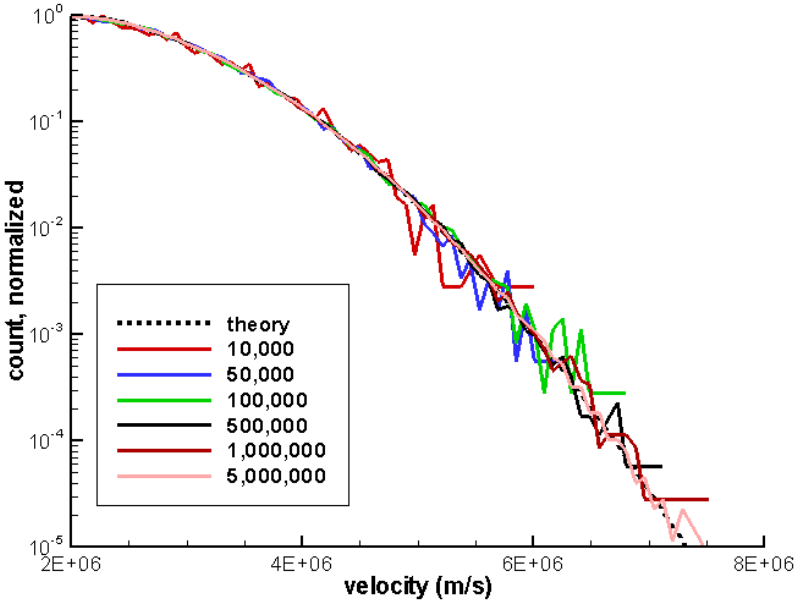


Figure 4.7: Close up of the high energy tail

4.2.4 Potential Solver

The field solver was tested by prescribing a defined variation in charge density and comparing it with the corresponding potential profile. The profile used was the sinusoidal variation, $\phi = \phi_M \sin(2\pi s/L)$ with $\phi_M = 10$ V. For a completely radial magnetic field line, $s = r$. Then using $\nabla_r^2\phi = (1/r)\partial/\partial r\,(r\partial\phi/\partial r) = -\rho/\epsilon_0$ we obtain

$$\rho_r = \epsilon_0\phi_M \left[\frac{4\pi^2}{L^2}\sin\left(\frac{r}{L}2\pi\right) - \frac{2\pi}{L}\frac{1}{R}\cos\left(\frac{r}{L}2\pi\right)\right] \tag{4.6}$$

Similarly, for the axial case in which $s = z$, we have $\nabla_z^2\phi = \partial^2\phi/\partial z^2$ and

$$\rho_z = \epsilon_0\phi_M \left[\frac{4\pi^2}{L^2}\sin\left(\frac{r}{L}2\pi\right)\right] \tag{4.7}$$

The boundary condition for both cases is identical $\phi_0 = \phi_L = 0$. These charge densities are plotted in Figure 4.8 along with the potential profile. The potential profiles for the two cases and the analytical variation overlap each other. The test was automated by computing the residue and comparing it to the tolerance value of 10^{-3}. Since the solution is obtained using a direct method, there is no inherent tolerance value built into the solver that could be used in this comparison.

4.2.5 Collisions

The first objective of the collisions unit test was to verify the probability selection algorithm. For the test, three collision processes with constant sigmas of 10^{-16}, 10^{-18} and 10^{-20} m^2 were specified. 20,000 electrons with $kT = 10$ eV were loaded and let to collide with a fixed background $n_n = 10^{19}$ m^{-3} for 10,000 time steps. The time step was set to 5×10^{-12} s which closely corresponds to the value computed with $it_{orbit} = 75$. No potential solution was performed and specular walls were used. After the simulation completed, the numerically-obtained collision frequency for each process

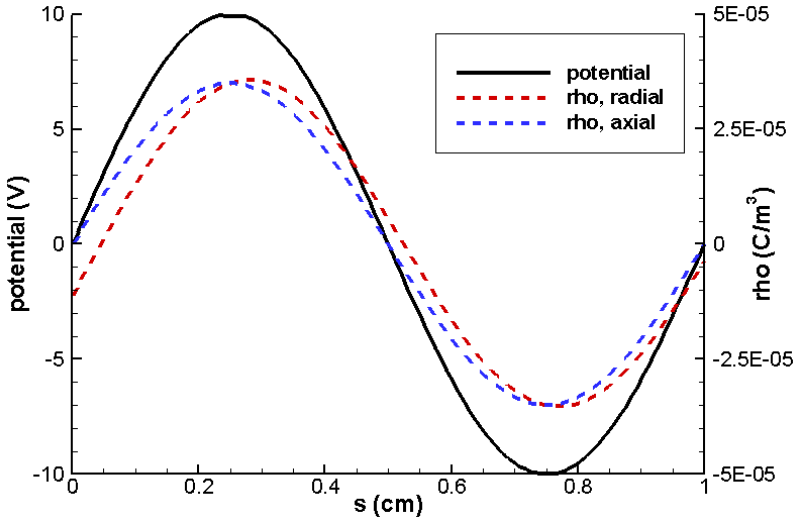


Figure 4.8: Prescribed charge densities and the computed potential for solver test

was compared to the analytical value, $\nu_i = n_n \sigma_i \bar{v}$, where $\bar{v}$ is the average electron velocity. Error for each process was computed from $E = (\nu_{num} - \nu_{theory})/(\nu_{theory})$. The total collisions cross section, $\nu_o = \sum_{i=1}^{3} \nu_i$ was also computed. The test passed if the total cross section agreed within 50% of the expected value. The test results are summarized in Table 4.2. This table also compares the results at various collision call frequencies. The column E_1 corresponds to results in which collisions were performed at each time step. Similarly, E_4 shows results from a case where collisions were calculated once every 4 time steps, and $\Delta t_{col} = 4\Delta t_{sim}$ was used. As can be seen from this table, the code is insensitive to the frequency of collision calculations. The velocity phase space of the electron population after 10,000 time steps is seen in Figure 4.9. The population is seen to retain the Maxwellian distribution. Figure 4.10 plots the temporal evolution in total kinetic energies for a system of colliding particles. The kinetic energy is seen to remain constant when $E_\perp = 0$, however begins to increase for a nonzero $E_\perp$. This increase corresponds to electrons diffusing across the magnetic field and receiving an energy gain corresponding to $\Delta U = e r_L E_\perp$.

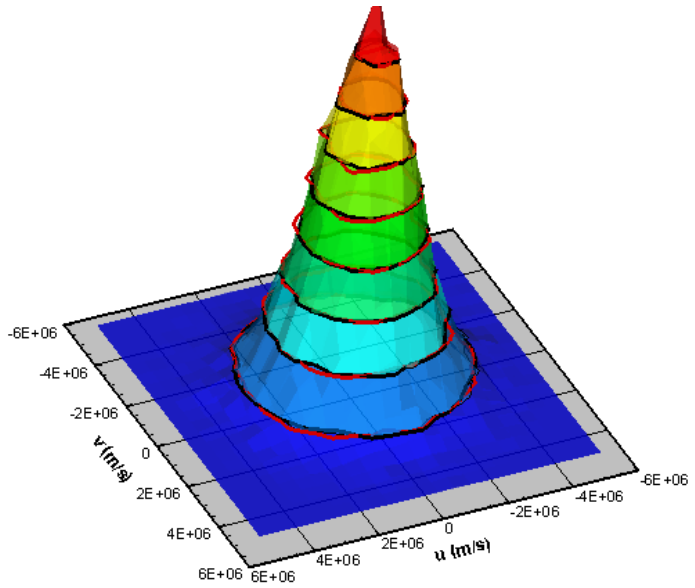


Figure 4.9: Comparison of initial (black and flood) to final (red) velocity distribution function after 10,000 time steps for a system containing 20,000 particles and colliding with constant $\sigma = 5 \times 10^{-19}$ m^2.

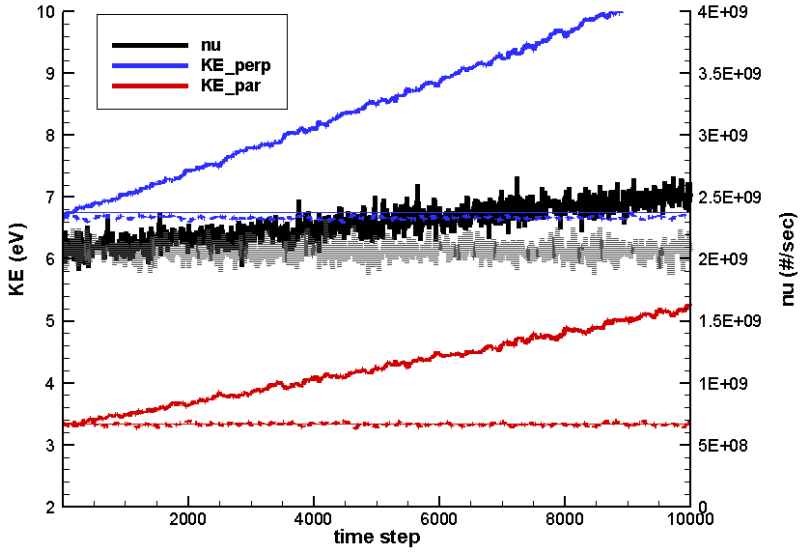


Figure 4.10: Comparison of system kinetic energies and collision rates for several cases. The thick solid line corresponds to the case with $E_\perp \neq 0$ and $\nu \neq 0$. The thick dashed lines are results for $\nu \neq 0$ but $E_\perp = 0$. Finally the thin solid line is the result obtained with $E_\perp \neq 0$ and $\nu = 0$.

Table 4.2: Errors in collision pair selection algorithm

σ (m^2)	E,1 (%)	E,4 (%)	E,15 (%)	E,75 (%)
10^{-16}	-0.65	-2.41	-8.63	-34.2
10^{-18}	+0.08	-0.76	+0.41	-0.41
10^{-20}	-3.41	+7.43	-11.1	-10.1
Total	-0.65	-2.39	-8.54	-33.8
Time (s)	63.7	38.7	32.2	30.27

4.2.6 Random Number Generator

Statistical methods such as particle-in-cell depend heavily on random numbers, and as such it is imperative to select a generator with a sufficiently large period that is also computationally efficient. Majority of random number generators are in reality pseudo-generators that return consecutive integers from an analytical sequence. Period of a random number generator is the number of successive samples that can be obtained before the sequence starts to repeat. Sampling of random numbers has been an ongoing issue in programs written in the C programming language. Some compilers implemented a generator with a particularly small period of only 32,768. The smallest difference between random numbers is $1/P$, where P is the period. For this case $\Delta R = 3.05 \times 10^{-5}$. As an example, for a case where a position is sampled from $x = R * L$, the smallest distance between sampled points is $\Delta x = 3.05 \times 10^{-6}$ m for $L = 10cm$. This number is independent of the number of samples, and loading more than $\sim 32,000$ points will not improve the spatial coverage.

The standard approach in C codes was to implement an alternate RNG with a large period, with Mersenne Twister and Numerical Recipes (NR) providing several viable alternatives. To determine whether such an approach was necessary, a Java implementation of the standard NR generator was compared with the generator implemented natively in Java. Additionally, a particularly bad generator with period of only 20 was also developed for comparison. The generators were tested by sampling 20 million numbers and binning them into 1 million bins. A "good" generator will

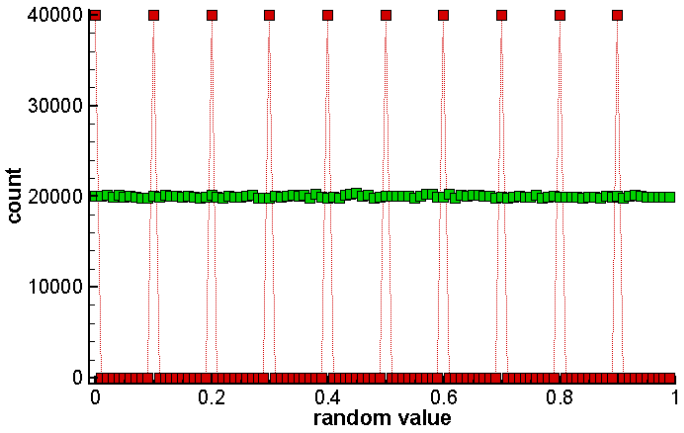


Figure 4.11: Histogram comparing a good random number generator (green) with a bad one (red).

Table 4.3: Sigmas and timing results for 3 random number generators

method	σ	time (s)
Bad	100.0	5.42
Java	0.0707	16.18
NR	0.0706	17.82

result in all bins receiving an equal coverage, while a bad generator will result in most values occupying a small number of bins, and the remaining bins receiving no coverage. This is illustrated in Figure 4.11. The generators were characterized numerically by computing the standard deviation $\sigma = \sqrt{(1/N)\sum_{i=1}^{N}(c_i - \bar{c})^2}$, where $\bar{c}$ is the average bin count. The passing criterion was set to $\sigma < 1$. The resulting sigmas along with timing data are listed in Table 4.3. As can be seen from this chart, the built-in RNG is effectively equivalent to the NR implementation. Since it executed at slightly higher speeds, the built-in generator was used for the subsequent simulations.

4.2.7 Transport Calculation

The final unit test verified the algorithm used to determine μ, the mobility coefficient. The tests were used to verify that (a) no transport is measured in the absence of transport inducing mechanisms (i.e. no numerical artifacts), (b) the correct value of mobility is obtained for classical cases, and (c) the results are independent of the

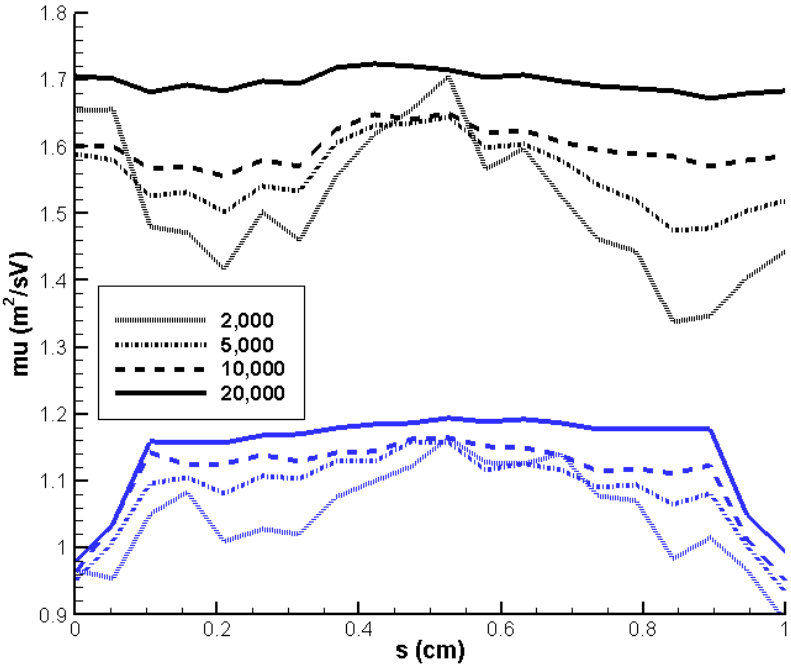


Figure 4.12: Comparison of the algorithm used to determine transport. The black lines correspond to calculations in which all particles are checked once per orbit. The blue curve corresponds to the case where only particles having completed a full orbit since a collision or a wall impact are included.

number of time steps. In these tests, the particle selection algorithm was also compared. As indicated in Figure 3.11, the true location of a particle guiding center can be determined only for particles not undergoing any scattering processes in the last cyclotron orbit. Two sets of cases were run, one in which all particles were characterized, and one in which only particles not having scattered in the last cyclotron orbit were considered ("orbit check"). Specular walls were used in this analysis, however, wall impacts were marked as scattering events, as were collisions. The code-determined mobility was compared to the analytical value obtained with the actual simulation collision frequency ν. Electric field $E_\perp = 20,000$ V/m was used. Tables 4.5 and 4.5 compare the mobilities obtained after a different number of time steps. The spatial variation is shown in Figure 4.12.

These plots show a good agreement in results obtained after a different number of time steps. The orbit check method is seen to reduce mobility by about 30%.

Table 4.4: Mobility without orbit check

it_{max}	μ	μ_{theory}	v_d	$v_{d,theory}$	E_{mu} %
1,000	1.4	1.17	1.4×10^4	1.2×10^4	+20.28
2,000	1.6	1.19	1.6×10^4	1.2×10^4	+31.31
5,000	1.6	1.21	1.6×10^4	1.2×10^4	+28.46
10,000	1.6	1.28	1.7×10^4	1.3×10^4	+24.60
20,000	1.7	1.40	1.9×10^4	1.4×10^4	+21.95

Table 4.5: Mobility with orbit check

it_{max}	μ	μ_{theory}	v_d	$v_{d,theory}$	E_{mu} %
$\mu = 0$	0.00077	0.00	5.8	0.0	+0.07
1,000	1.0	1.17	1.0×10^4	1.2×10^4	-12.46
2,000	1.1	1.18	1.1×10^4	1.2×10^4	-9.56
5,000	1.1	1.22	1.1×10^4	1.2×10^4	-8.77
10,000	1.1	1.29	1.2×10^4	1.3×10^4	-13.97
20,000	1.1	1.41	1.2×10^4	1.4×10^4	-18.59

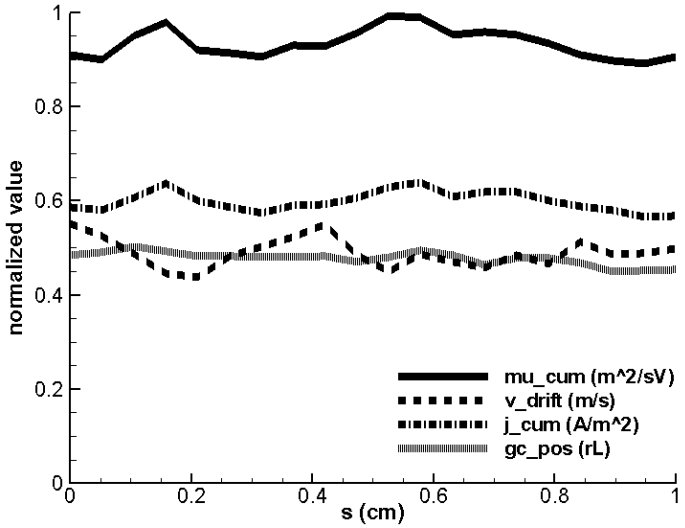


Figure 4.13: Field line transport properties for collisions, specular walls, and no potential solve with uniform neutral density. The normalization factors are 1, 37500, 500, and 10 for μ, v_d, j, and x_{gc}, respectively.

In addition, it results in a dip in the computed value near the walls. This dip is non-physical and is due to the omission of wall-impacting electrons from the transport calculation. For this reason, the orbit check method is not used in the actual simulations. Figure 4.13 shows several code-determined transport properties for a case similar to Figure 4.12. These include cumulative mobility, guiding center drift velocity, cumulative value of current density, and the guiding center position. All values are seen to be relatively constant. The level of oscillations in these plots is an indicator of the numerical noise in the method. For this simple setup, the variation in mobility is seen to be ± 0.1 T^{-1}. Mobilities in Hall thruster approach or even cross below this value. This finding clearly indicates that an important future work item is finding a method for reducing numerical noise without increasing the computational overhead.

The ability of the code to capture variation in μ was tested by varying the collision cross-section for the constant σ collision model. Specular walls were used and potential solver was not used. The background neutral density was 10^{19} m^{-3} and $B = 0.01$ T was used. Electrons were sampled from a population at 10 eV. Results obtained after 5000 time steps are shown in Table 4.6. Each entry corresponds to results obtained by averaging three independent simulations, however, the individual runs showed very little variation, with difference in values between runs limited to few percent. A generally good agreement is seen in all cases, with the code capturing the correct order of magnitude of classical mobility in all cases, and in the range corresponding to the expected drift speeds in the Hall thruster, the error magnitude remaining below 20%.

Finally, the dependence on the perpendicular electric field $E_\perp$ was also examined. This test is analogous to the test described above, except this time $\sigma = 10^{-17}$ m^2 was held constant, and the electric field was allowed to vary. Results from this test are summarized in Table 4.7. These results are seen to follow the same trends observed

Table 4.6: Mobility variation with σ, without orbit check, 5000 time steps, and $E_\perp = 20$ kV/m.

σ (m^2)	μ	μ_{theory}	v_d	$v_{d,theory}$	E_{mu} %
10^{-20}	0.023	0.016	4.82×10^2	3.27×10^2	+42.2
10^{-19}	0.224	0.155	4.47×10^3	3.09×10^3	+45.1
10^{-18}	1.71	1.48	3.00×10^4	2.95×10^4	+15.6
10^{-17}	14.4	39.9	2.89×10^5	2.79×10^5	+2.98
10^{-16}	33.5	47.7	6.78×10^5	9.54×10^5	-29.7

Table 4.7: Mobility variation with $E_\perp$ and $\sigma = 10^{-17}$ m^2.

$E_\perp$ (V/m)	μ	μ_{theory}	v_d	$v_{d,theory}$	E_{mu} %
10,000	10.2	7.53	1.00×10^5	7.53×10^4	+35.0
30,000	17.3	19.2	5.17×10^5	5.76×10^5	+10.0
50,000	22.1	27.4	1.10×10^6	1.37×10^6	-19.5

previously in Table 4.6. The difference in $E_\perp$ results in a variation in the in-plane component of particle velocity $v_\perp$, and hence a different collision frequency. Hence at lower values of $E_\perp$, the theoretical value of mobility is reduced which is analogous to a decrease in σ. Again, the code is seen to over-predict mobility at lower values of theoretical $\mu_\perp$ and underpredict it at at higher values. However, the agreement remains within 50%.

4.3 Parametric Studies

The code was next applied to an "academic" configuration to study the impact of various parameters on electron dynamics and transport. The geometry assumed in the the following runs is sketched in Figure 4.14. The magnetic field line consists solely of the radial component. However, the field line was assumed to follow the centerline of a magnetic mirror, allowing for $\partial B/\partial s \neq 0$. Neutral density was set to $n_a = 10^{19}$ m^{-3}. Ion density was $n_i = 5 \times 10^{16}$ m^{-3}, and the normal electric field $E_\perp = 20,000$V/m was used. Initial kT_e is isotropic with $kT_e = 1$, 10, or 35 eV. The magnetic field strength $B = 0.01$ T was used. The magnetic mirror was not included

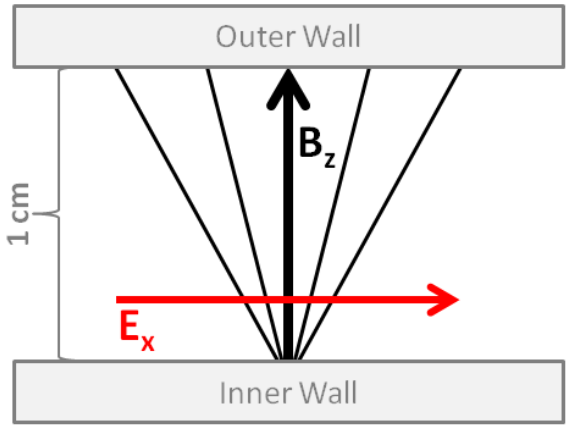


Figure 4.14: Simulation setup for mobility studies. Planar geometry is used.

in the first set of simulations, and is described in more detail later. For simplicity, the planar $x - y$ configuration was assumed, and conductive walls, $\phi = 0$ on both walls, were used. These parameters were chosen to reduce the number of factors influencing the results.

Figure 4.15 shows the typical solution. Here the black curve plots the plasma potential, the blue curve is the charge density ρ and the red curve is the electric field along the field line. As expected, a generally zero field is achieved in the bulk region, and a strong electric field is seen in the near-wall sheath. Plasma also deviates from quasineutrality here as can be seen from the non-zero potential. The times required to run these cases were on the order of minutes. 100,000 particles were used in each simulation, and each simulation ran for 20,000 time steps.

4.3.1 Velocity Distribution Function

The first run was concerned with investigation of the modification of the electron EDF by secondary electrons. Such an effect was reported previously by Sydorenko [53]. For this analysis the normal component of electric field $E_\perp$ was set to zero and a simplified wall model was used. The walls acted as a source of secondary electrons, with the SEE coefficient given by k_{see}. For $k_{see} = 1$ one SEE was created

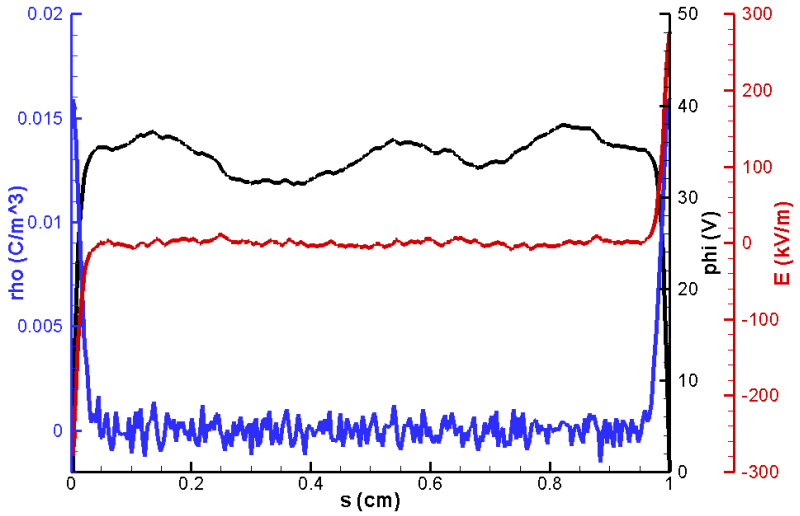
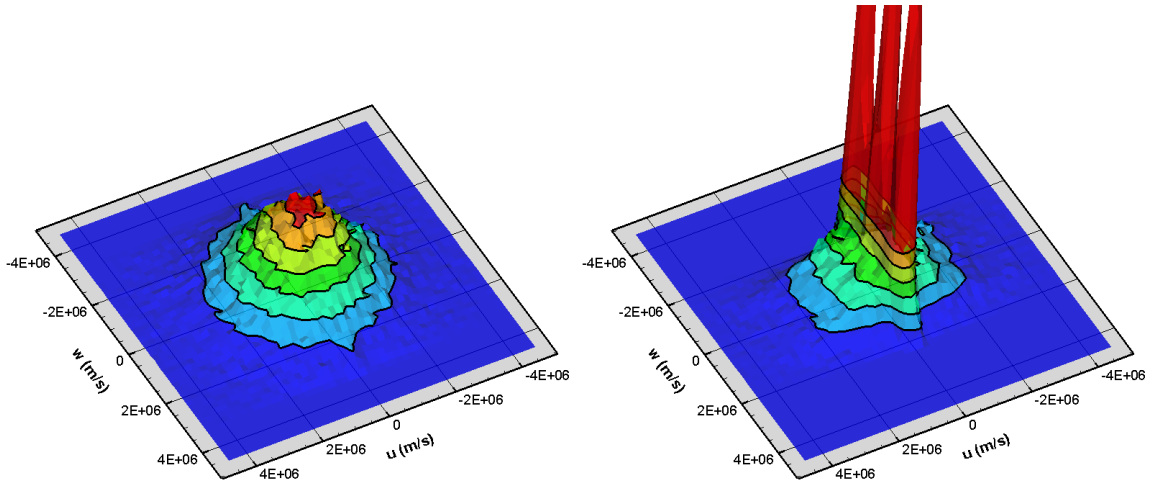


Figure 4.15: Typical plasma parameters at steady state

for each incident electrons and for $k_{see} = 1.2$ on average 1.2 secondary electrons were emitted. Velocity distribution function after 10,000 time steps is shown in Figure 4.16. The first plot shows the result obtained with $k_{see} = 1$. This population shows only a minimal hint of a SEE population, given by the kinks at $u = 0$. The bulk population remains Maxwellian. However, increasing the emission coefficient to 1.2 produces a very different result. The electron population becomes saturated by the secondary electrons and the distribution becomes anisotropic. Furthermore, a beam-like population emerges, corresponding to SEE streaming between the two walls.

4.3.2 Secondary Electrons Under a Magnetic Mirror

Figure 4.17 shows a possible transport mechanism in the presence of a particularly strong magnetic field gradient. Under this configuration, electrons can be expected to be accelerated to the wall, and cause an emission of secondary electrons. These electrons then generally diffuse in the $-E_{\perp}$ direction. If the mirror is strong enough, the SEE will be accelerated back to the wall where it will induce another secondary electrons.Electrons thus continue to travel towards the anode. It should be noted that this setup assumes the magnetic field lines are continuously diverging which



(a) $k_{SEE} = 1.0$ (b) $k_{SEE} = 1.2$

Figure 4.16: Modification of velocity distribution function by secondary electrons

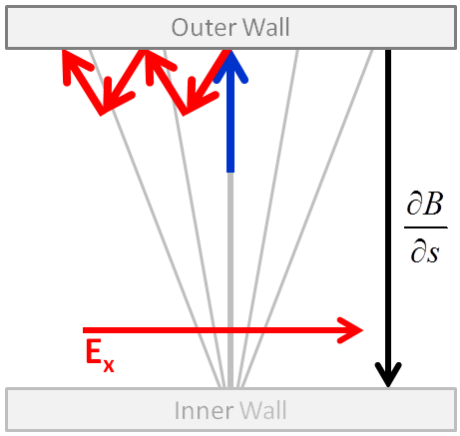
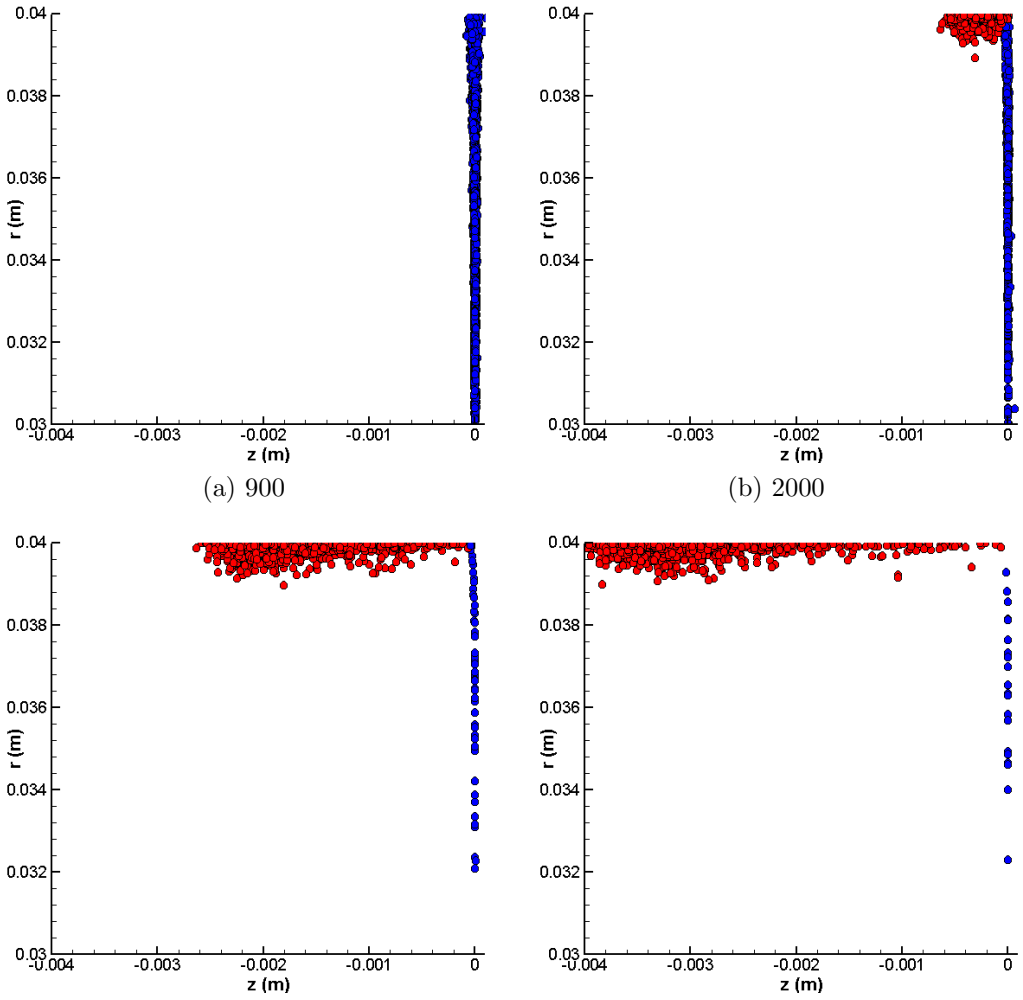


Figure 4.17: Near wall transport due a magnetic pressure

may not be a valid assumption in a general thruster devices. However, this setup also approximates the situation that arises in SEE-saturated sheath, in which the potential profile becomes non-monotonic, and sheath inflection occurs. In that case, secondary electrons born at the wall will also lack the sufficient energy to escape the wall region, and will cascade in a similar fashion towards the anode. This is the basis of the near-wall conductivity model of electron transport.

A time sequence from the simulation results is presented in Figure 4.18. This figure shows positions of random 2000 particles sampled at different simulation times. The



(a) 900 (b) 2000

(c) 6000 (d) 9000

Figure 4.18: Near wall transport of secondary electrons under a strong mirror

primary population is shown in blue and the secondary electrons are shown in red. The SEE emitting walls from the previous analysis were used in this test case. The walls were set to start emitting SEE at time step 1000. Electrons impacting the walls prior to this step were absorbed. In addition, electrons lost to the wall were not reinjected, resulting in an eventual depletion of the primary population. A clear SEE population is seen at time step 2000. This population is seen to diffuse in the $-E_{\perp}$ direction, per diagram above. The distinct tail is due to the reduced density in the depleted primary beam.

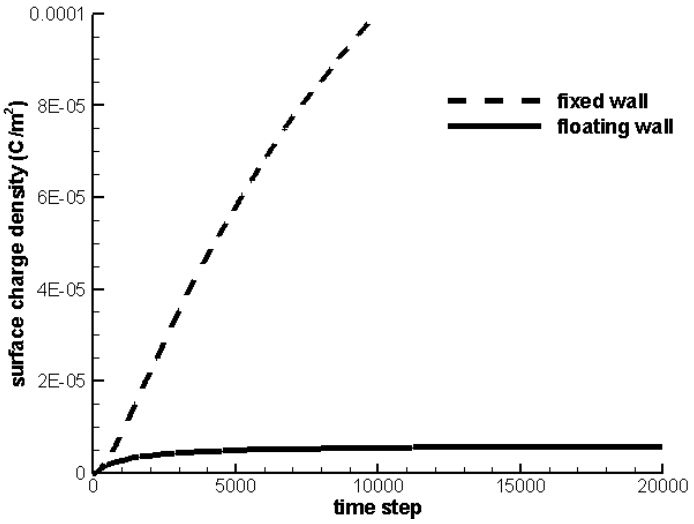
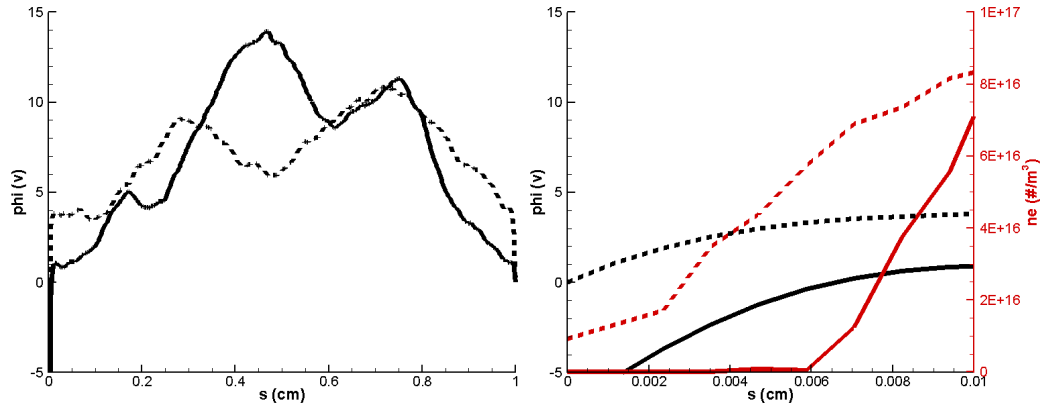


Figure 4.19: Time evolution in collected wall charge for a fixed and a floating wall

4.3.3 Dielectric Walls

The effect of the dielectric walls was also investigated. For this run, an electron population was loaded between Sydorenko walls. Figure 4.19 show the time evolution in collected wall charge for a fixed (conductor) and a floating (dielectric) wall. Clearly, a true conducting wall will result in zero surface charge, however, considering a Dirichlet "dielectric" wall is a useful testing construct. The collected charge on this wall continues to grow, since no feedback exists between the wall and the plasma. A different situation arises with a true dielectric wall. While the surface charge initially increases in a manner similar to the fixed wall, the wall charge quickly asymptotes. This in effect indicates a reduction in wall flux, which is also represented in a reduction in the secondary electron population. In this case, the number of secondary electrons decreased from 150 to 40. Figure 4.20 shows a representative potential profile. Similar profiles develop in the bulk population. Figure (a) shows a close up of the near-wall sheath region which is not clearly visible in Figure (b) due to the small sheath size in this particular setup. The potential on the floating wall is seen to have decreased, resulting in a decrease in electron density.



(a) Potential Profile (b) Sheath

Figure 4.20: Plasma potential and electron density for a floating dielectric (solid) and a fixed Dirichlet (dashed) wall

4.3.4 Azimuthal Oscillations

Electric field in the azimuthal direction, E_θ will naturally induce axial transport due to the crossed $\vec{E} \times \vec{B}$ drift. The idealistic axisymmetric configuration of a Hall thruster dictates that $E_\theta = 0$. However, azimuthal waves are known to exist and have been reported and analyzed elsewhere. Choueiri [24] reports two types of azimuthal waves: a low frequency "spoke" mode in the 5-25 kHz range, and a high frequency azimuthal model with frequencies in th 20-60 kHz range. Since the typical range of cyclotron frequencies is in the gigahertz range, such oscillations become very low frequency when viewed on the time scale of an electron. The result is an almost constant E_θ component, resulting in a uniform x drift. That this is indeed the case can be studied by modulating E_y according to $E_y = E_w \cos(\omega_w t + p) + E_0$. The parameters used were $E_w = -1,000$ V/m, $p = 0$ and $E_0 = 0$. Two values of ω_w were investigate, 15kHz and 40kHz, corresponding to the middle of the reported range. As expected, at this frequency range, the electron exhibits a uniform x drift. This finding confirms the importance in azimuthal waves, and suggests that effects such as the spoke mode need to be captured in Hall thruster simulation codes. In determining the importance of such oscillations on the overall electron transport, the

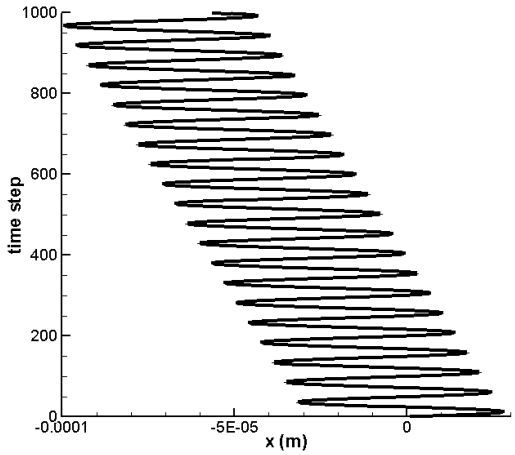


Figure 4.21: Guiding center drift to an azimuthal electric field

actual frequencies of these ionization events need to be taken into account.

4.3.5 Synergistic Wall Effects

A common assumption in Hall thruster codes is that

$$\mu = \mu_{classical} + \mu_{walls} + \mu_{Bohm} \tag{4.8}$$

The classical term corresponds to the transport due to particle collisions. Ignoring the Bohm term, the validity of the $\mu = \mu_{classical} + \mu_{walls}$ relationship can be tested by running three sets of simulations, one with collisions and specular walls, one with wall effects but no collisions, and one with both. The field solver needs to be included to take into account any possible sheath effects. To identify any possible dependence on temperature, three sets of electron temperatures were also considered, with $kT_e =$ 1, 10, and 35 eV. A simplified academic approach was used to represent collisions. Instead of utilizing a realistic combination of momentum transfer, Coulomb, and excitation and ionization collisions, only the momentum transfer interaction was used and a constant cross-section $\sigma = 1 \times 10^{-18}$ m^2 was used. Walls were either specular for the collisions only case, or were represented by the Sydorenko model. The plasma density in all cases was 1×10^{16} m^{-3} and neutral density was 10^{19} m^{-3}. In all

cases $E_\perp = 20,000$ V/m and $B = 0.01$ T were used. The time step was set from $it_{orbit} = 150$.

Results from these three runs are shown in Figure 4.22. The results are grouped by the particle energy. The first column shows the current density variation with the distance along the field line, while the second column shows the corresponding mobility. The second set of pictures also includes a qualitative plot of electron density (in red) and electric potential (in blue). These two curves are included to illustrate the extent of the bulk and the sheath regions. Since the sheath scales with the Debye length, which in turn scales with electron temperature, the sheath thickness is seen to increase from top to bottom. Although values of current density and mobility are included, these are merely illustrative since this particular setup does not correspond to a real thruster. We can see that for $kT_e = 1$ eV, electron transport is completely dominated by collisions. There is very little difference between the results computed with collisions only, and by including both collisions and wall effects. The wall-induced transport is negligible in this case.

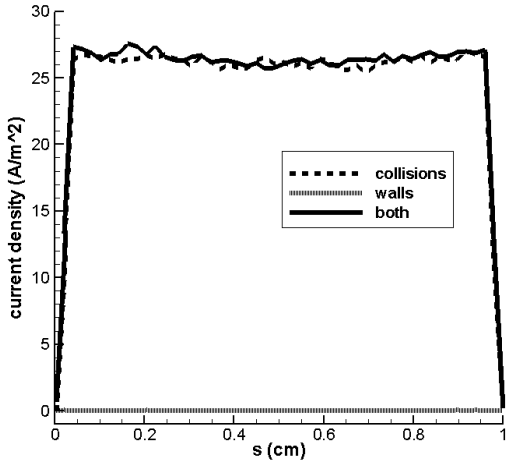
The situation changes somewhat as the electron temperature is increased. At $kT_e = 10$ eV, wall interactions can be seen to play a role in mobility, with mobility demonstrating a clear near wall increase. This increase in mobility does not correspond to a linear increase in current density, due to the decrease in electron density in the sheath region. However, of more interest is the observation that $j_{wall+collisions} > j_{wall} + j_{collisions}$. This additional "anomalous" current can be seen in both the bulk region and also in the near wall sheath, where it is especially prominent. This increase in total current is even more pronounced when one considers the $kT_e = 35$ eV case. This case exhibits a finite cross-field current due to wall effects, j_{walls}, with increased magnitude near the walls. On the other hand, the current density due to collisions only shows a flat profile in the bulk region, and rapid decay in the sheath. However, when the two independent processes are combined, the

resulting current profile shows a strong deviation from the two basis. A significant increase in near-wall conductivity can be seen when both collisions and wall effects are considered. The bulk population also demonstrates an increase in current. The representative values for $j_{collisions}$, j_{walls}, and j_{both} are 71.5, 1.8 and 83 A/m^2, respectively. Hence, in the bulk population, the combination of the two processes results in a 14% increase in trans-field current density.

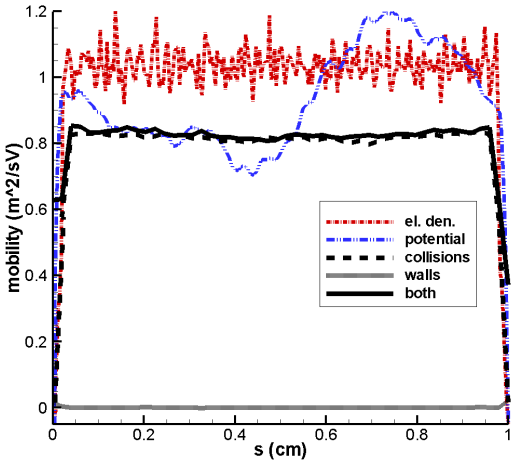
These results demonstrate that

$$\mu_{classical+walls} > \mu_{classical} + \mu_{walls} \tag{4.9}$$

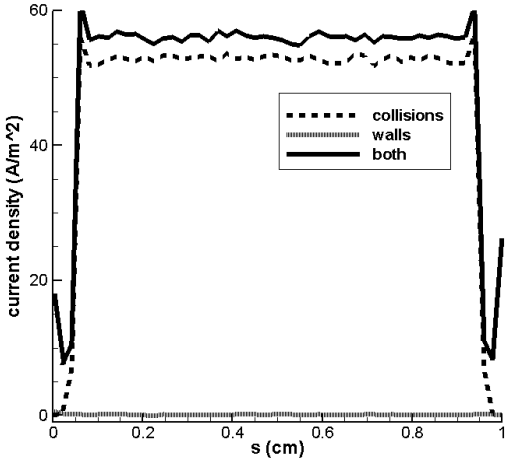
The actual quantitative results obtained here only correspond to the academic setup considered here and may not be representative of a physical Hall thruster. Yet, the observation of a synergistic response is of a significant importance to Hall thruster modeling efforts. This finding indicates that simple analytical models taking into account each mobility term individually may not account for the interaction between the processes. Such synergistic responses are however not easily included in analytical models, suggesting the need for a kinetic treatment of electron transport. The explanation for this effect is simple. In a steady discharge, a balance develops between the wall potential drop and the parallel component of electron energy. The size of the potential well adjusts self-consistently to trap electrons away from walls. As such, in this classical mode, wall flux is not a significant contributor to conductivity due to the low flux of primary electrons (a different situation arises in the case of a saturated sheath or a magnetic mirror). Collisions scatter the particle velocity components. Even a small angle event, such as Coulomb collision, may impart a sufficient $\Delta v_{\parallel}$ to allow the electron to reach the wall. This additional scattering is demonstrated in an increased number of secondary electrons. As an example, the case with wall collisions only resulted in a 0.05, 0.80, and 6.81 percent of total population occupied by



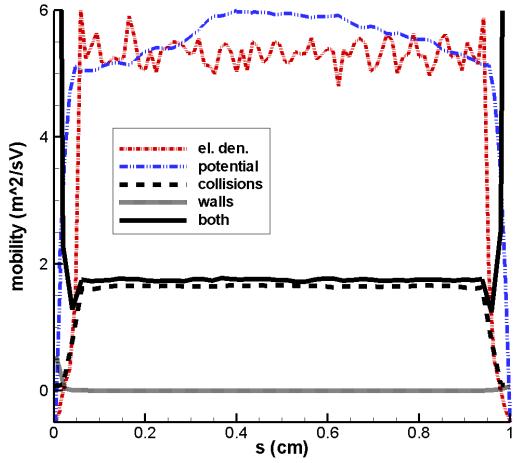
(a) Current density, $kT_e = 1$ eV



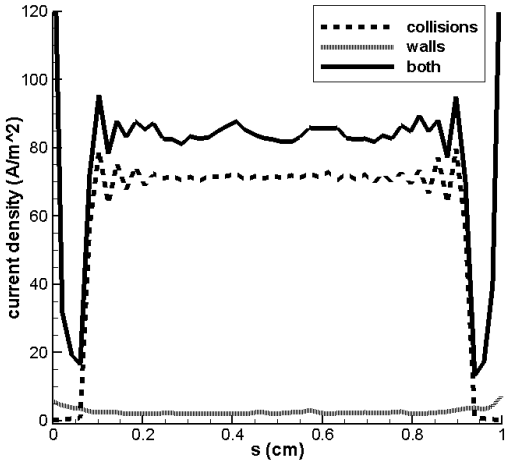
(b) Mobility, $kT_e = 1$ eV



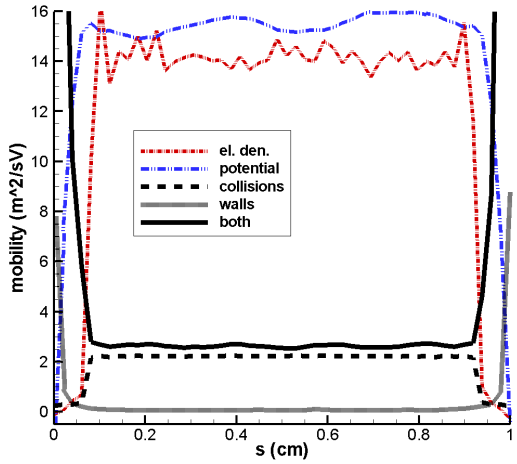
(c) Current density, $kT_e = 10$ eV



(d) Mobility, $kT_e = 10$ eV

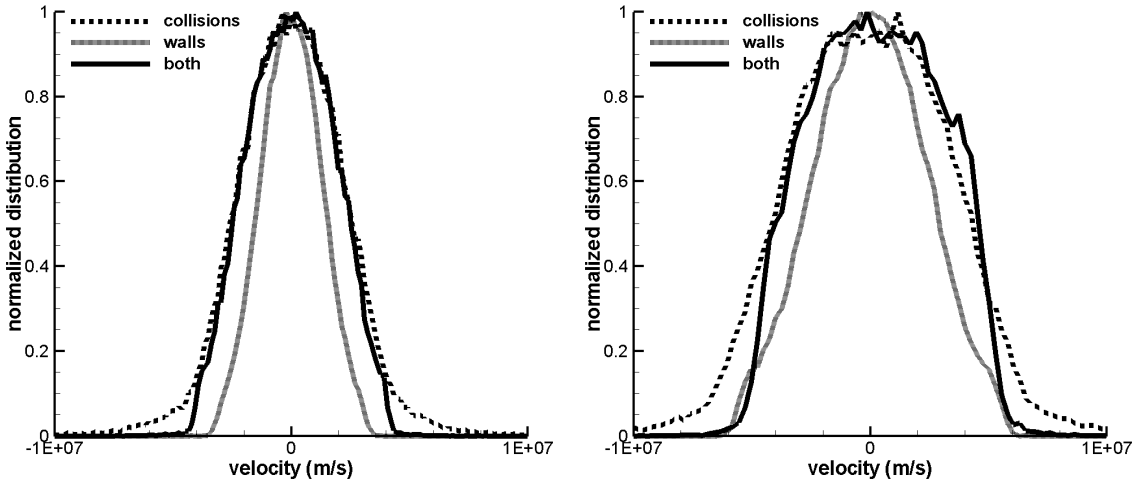


(e) Current density, $kT_e = 35$ eV



(f) Mobility, $kT_e = 35$ eV

Figure 4.22: Mobilities due to collisions and/or wall effects. Qualitative potentials and electron densities shown for the combined (walls+collisions) case.



(a) 10 eV (b) 35 eV

Figure 4.23: Comparison of velocity distribution functions for the three considered cases for initial electron temperatures 10 and 35 eV.

secondary electrons, where the three values correspond 1, 10, and 35 eV, respectively. Once collisions are included, these percentages increase to 0.06, 1.32, and 10.23%.

The effect of collisions on near wall transport can also be seen by plotting the velocity distribution functions along the magnetic field line. These plots are shown in Figure 4.23. The dashed and dotted lines show the velocity distribution functions for the two individual cases. The wall cases demonstrates a cooler population which is due to both the loss of high energy electrons to the wall and the injection of cold secondaries. The inclusion of collisions is seen to increase the spread of the VDF, resulting in an increase in $kT_{e,\parallel}$. This increase is due to redistribution of redistribution of $kT_{e,\perp}$ by the scattering events, which can be visualized as events rotating the particle velocity space. It should be noted that this additional thermalization is expected to become more pronounced as $kT_{e,\perp} > kT_{e,\parallel}$. This additional energy imparted into the parallel $v_{\parallel}$ component thus results in an increased number of electrons able to penetrate the sheath and interact with the wall.

Chapter 5

Studies of the Princeton Cylindrical Hall Thruster

The second spatial scale considered by the multiscale approach is the scale of the thruster. On this scale the global discharge parameters, such as potential drop and heavy particle densities, are determined. As such, this scale forms the pillar on which the remaining analysis rests on. The 2D hybrid code provides the background and cross-field properties utilized by the kinetic code described in Chapters 3 and 4, and also provides inputs to a subsequent near-wall erosion analysis and to plume modeling. In this work, the previously described code HPHall was used. HPHall was developed at MIT in late 1990s by Fife and has since been improved by the original author [60] as well as other researchers [61, 62]. In this work an older version of the code was used which did not contain some of the modifications described in these references. The analysis was performed using the 2.6 cm Princeton Cylindrical Hall thruster (CHT). This thruster was selected due to both its smaller size and interesting physics associated with the nonstandard geometry. More specifically, of interest was the magnetic mirror near the inner pole. Unfortunately, during the course of this work, it was realized that HPHall, due to peculiarities of its implementation, is not suitable for modeling Hall thrusters with converging field lines, as is the case in the CHT. This in turn limited the flexibility in controlling the boundary conditions and limited the kinetic computation mainly to the annular region. This issue is described

below. As such, the results presented in this chapter serve primarily to illustrate the multiscale approach and should not be thought of as an accurate representation of the thruster internal dynamics.

5.1 Overview of the Thruster

The model is deployed to the 2.6 cm Cylindrical Hall thruster (CHT) from Princeton Plasma Physics Laboratory. This thruster is described in greater detail in [3]. Here only the parameters important to the analysis are summarized. The most important characteristic of this device is its non-standard geometry. While typical Hall thrusters consist of an annular channel, the CHT contains an annular upstream zone and a cylindrical acceleration zone. The lack of the inner wall in the acceleration region is expected to lead to an increased thruster lifetime and improved performance due to reduced losses of ions to the walls. From the academic standpoint, this configuration also introduces interesting new physics. The magnetic field lines converge near the centerpole, resulting in a region of increased magnetic pressure. Electrons are then expected to be preferentially scattered to the outer wall, possibly resulting in a non-symmetric sheath. This finding was touched upon in in a prior work [63].

Experimental measurements of this thruster have been presented in [64] and [65]. The CHT can be operated in two modes based on the current applied to the magnetic coils. Our work correlates to the "direct" mode. In this configuration, magnetic field lines cross the channel between the outer wall and the centerpole without forming a cusp near the outer wall. Electron temperature in the thruster was found to peak at 25 eV just outside the exit plane. Potential decays slowly through majority of the channel. Only 50V of potential drop were measured in the first 1.5 cm from the anode. Most of the potential drop occurs near the exit plane, in the acceleration zone. Additional 50V potential drop was measured to occur outside the thruster. Anode current was approximately 0.3 A, and plasma density was $\sim 6 \times 10^{17}$ m^{-3}.

5.2 HPHall Modeling

The schematic of the simulation setup is shown in Figure 5.1. HPHall uses a structured but a non-uniform mesh, boundary of which is shown in the figure. The nodes are computed by connecting the control points along the boundary and relaxing them with a Laplace solver. Such a mesh simplifies capturing the geometry of non-rectangular devices, as well as the downstream near-plume region. It is also presumed that such a mesh was selected to increase the alignment of radial grid lines with the magnetic field lines. However, the particle mesh is not used for solving the electron equations. Instead, internally HPHall creates another virtual mesh on which the electron equations are solved. Plasma parameters determined from particles are interpolated onto this secondary mesh. This mesh is also shown in Figure 5.1. The vertical lines correspond to approximately equidistant magnetic stream function λ lines. The two thick lines at the upstream and the downstream end form the boundaries. The upstream λ line is the anode line. Fixed potential is applied along this line and particle grid nodes upstream are set to this constant value. The downstream line is the cathode line. Reference ϕ_0 and $kT_{e,0}$ is set on this line. Fife also initially set the reference plasma density here, however, the incarnation of the code allowed for this value to be set automatically to match the ion data. The electron equations are solved only between these two lines. Downstream of the cathode, HPHall determines electron temperature by assuming a linear decay.

It should be noted that the graphics of the magnetic mesh does not completely correlate to the implementation in HPHall. The data structure used by HPHall uses a variable number of radial segments, which leads to an interesting visualization problem. The mesh shown here, and in the subsequent analysis, uses a uniform number of radial segments corresponding to the average number used by HPHall. HPHall injects kinetic neutrals at the upstream boundary. The neutrals are injected between a selected inner and outer radial node. Plot of the initial neutral plume,

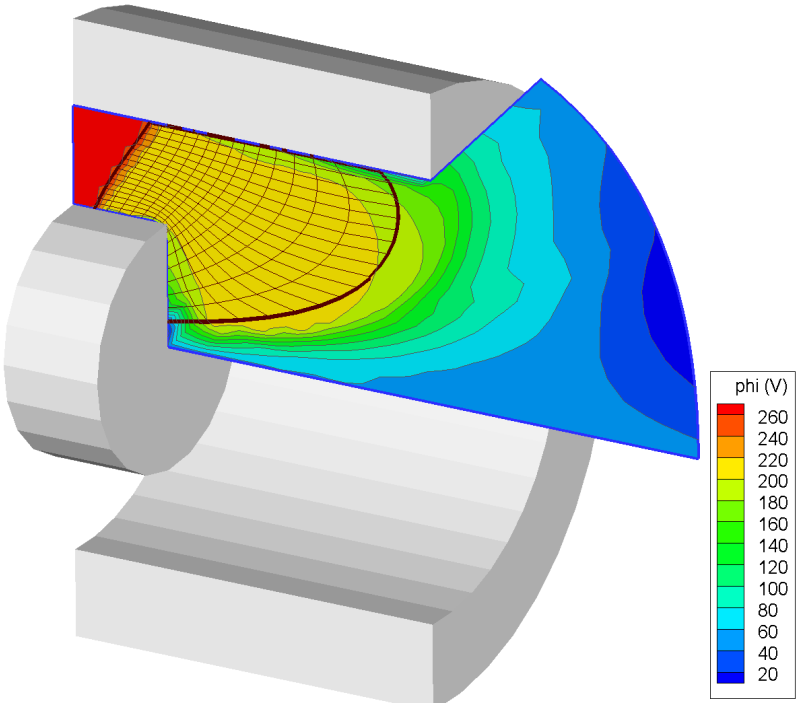


Figure 5.1: Model of the cylindrical Hall thruster. Slice shows the HPHall computational domain. The mesh corresponds to the region in which the electron conservation equations are solved.

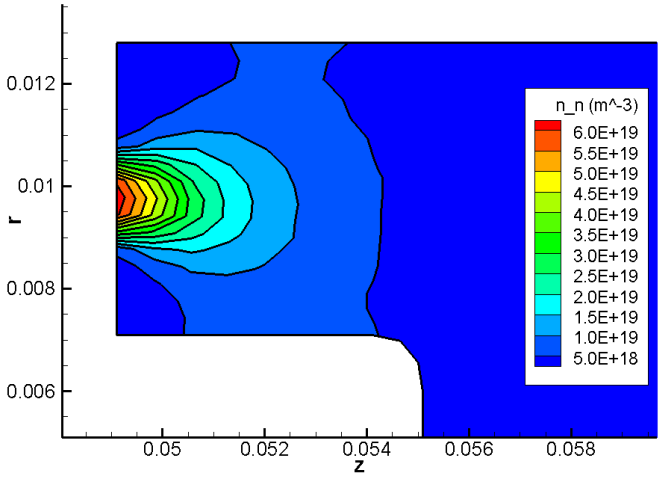


Figure 5.2: Neutral injection at the anode, showing neutral density after 50 time steps. Ionization was not considered.

computed by turning ionization off and running the simulation for only a small number of time steps, is shown in Figure 5.2. The typical HPHall simulation starts by first prefilling the chamber with neutrals by running the code in a "neutrals-only" model. The simulation is then restarted in a normal mode in which ions are created and electron equations are solved. Ionization is modeled by calculating the ionization rate in each cell according to the local electron temperature and neutral density. A corresponding number of ions is then created and the specific weight of random neutrals in the cell is reduced by a corresponding amount. HPHall uses a variable weight model, simplifying this implementation. It should be noted that such an approach is inherently different from the statistical approach taken by MCC or DSMC methods, and is similar to the implementation used previously to model the formation of CEX ions in an ion thruster plume [66]. Ion positions are integrated using the particle-in-cell method. The code assumes charge neutrality, and hence $n_e = n_i + 2n_{i2}$, summation of singly and doubly charged ion populations. In order for quasineutrality to hold, electron temperatures must adjust accordingly to accommodate diffusion. As indicated in Chapter 2, HPHall solves the energy equation

$$\frac{\partial}{\partial t}\left(\frac{3}{2}n_e k T_e\right) + \nabla \cdot \left(\frac{3}{2}n_e k T_e \vec{u}_e + \vec{q}_e\right) + \nabla \cdot \left(n_e k T_e \vec{u}_e\right) = S_h - S_i \tag{5.1}$$

on the previously described lambda mesh. Constant electron temperature at each field line is assumed. This then allows each field line to be treaded as single volume element. Two-dimensional properties appearing in the energy equation are area averaged over the field line. The conservation equations are solved only in the direction normal to the field lines, but 2D radial contribution is used to compute the coefficients at each point. Electron velocity, given by the momentum balance,

$$\vec{u}_e = \mu_e \left(\frac{d\phi^*}{d\hat{n}} + \frac{k}{e}\left(\ln(n_e) - 1\right)\frac{dT_e}{d\hat{n}}\right) + u_i \tag{5.2}$$

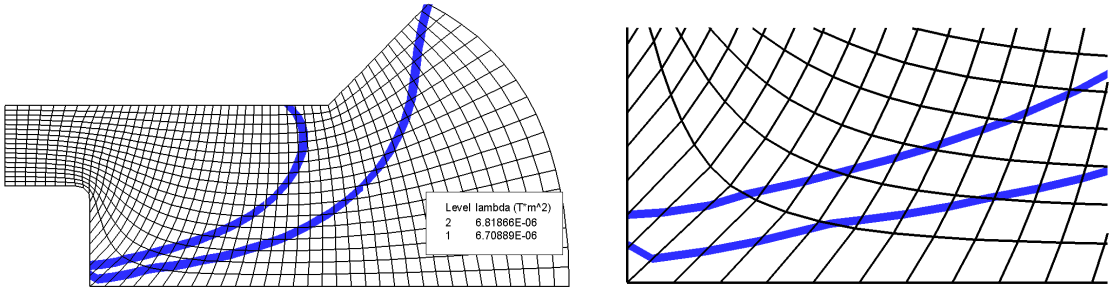
is incorporated into the temperature solver. Here μ_e is the mobility term. HPHall determines mobility from the classical relationship, and also by considering anomalous Bohm term.

$$\mu_\perp = \frac{|q|}{m\nu}\left(\frac{1}{1+\omega_c^2\tau^2}\right) + \mu_{walls} + k_B\frac{1}{16B} \tag{5.3}$$

where $\omega_c^2 = (|q|B/m)^2 \sim 1/B^2$ scales with $1/B^2$. This distinguishes the classical mobility from the "anomalous" Bohm term which demonstrates a linear dependence. The Bohm coefficient is one of the several user specified parameters. Once temperature has been determined, thermalized potential ϕ^* is computed at each lambda line from current conservation. The radial variation in potential is then recovered from the thermalized potential relationship, $\phi^* = \phi - kT_e/e\ln(n_e)$.

5.2.1 Inability of HPHall to resolve the CHT Magnetic Field Topology

HPHall is strongly dependent on the simulation boundary conditions as noted by Nakles [44]. Unfortunately, the unstructured mesh topology used by HPHall makes it difficult to model the CHT. For yet to be understood computational reasons, HPHall shifts the desired cathode λ line upstream towards the anode such that the lambda line intersects a grid node at the bottom mesh boundary. While this shift likely has a negligible impact in the standard annular topologies for which HPHall was developed, it introduces a severe limitation when a CHT-like configuration is used. In the CHT, the magnetic field lines converge near the centerpole. In addition, due to the body-contouring grid, the region near the center pole is not well resolved and consists of large cells. This shift then results in the code utilizing a vastly different magnetic field line than desired. This behavior is plotted in Figure 5.3. This figure shows the particle mesh and two different magnetic field lines. The desired boundary line, corresponding to a particular set of experimental measurements, is shown on the right. The second line, shown to the left, is the actual cathode line boundary utilized by the code. The second image shows a close up the region where the field lines converge. It needs to



(a) (b) Close up of the inner wall region

Figure 5.3: The desired (right) and the used (left) cathode λ line

be noted that the actual shape of the magnetic field lines is determined internally by HPHall by differentiating the providing magnetic field profile. These two lambda lines can be seen to correspond to vastly different regions in the thruster, indicating that the specified boundary condition no longer correlates to the spatial position at which it's being applied. Even more important is the fact that HPHall does not solve the electron equations downstream of the cathode λ line. This region contains much of the interesting physics, since it is here where the potential drop responsible for accelerating the ions is established. Unfortunately, attempts taken to mitigate this issue were not successful. These attempt looked at increasing the mesh resolution or attempting to utilize a different magnetic field lines. Due to this issue, predominantly only the annular segment of the thruster is considered by the electron solver, and as such, any mobility correction obtained from Lynx is not expected to have a significant impact on the overall solution. Hence for this reason, the results presented here need to be taken with a grain of salt, and understood to serve mainly an illustrative role of the multiscale model. As noted in the last chapter, the primary future work task is development of an alternative to HPHall that will be able to capture non-standard geometries.

Table 5.1: Summary of HPHall simulation inputs

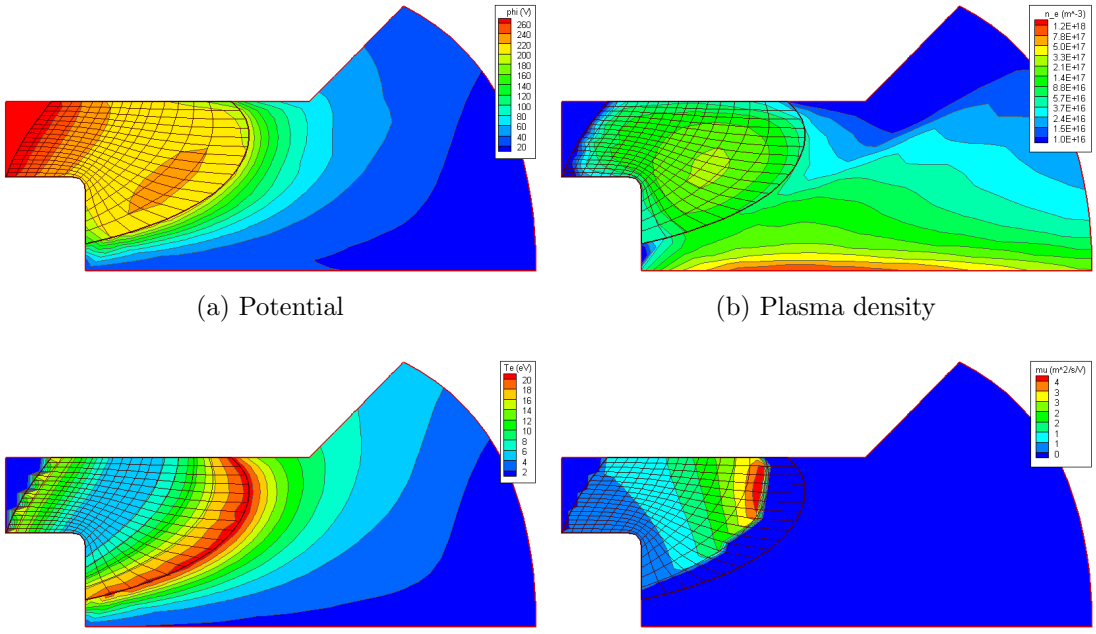
Parameter	Value
mass flow rate	4×10^{-7} (kg/s)
discharge voltage	275 V
anode line position	(0.0520, 0.010) m
cathode line position	(0.0670, 0.010) m
cathode temperature	26 eV
cathode potential	210.6 V
cathode density	$6{\times}10^{17}$ m^{-3}
cathode emitter potential	-20 V
Bohm coefficient	1.0

5.2.2 Initial Results with Classical Mobility

The initial results obtained with HPHall are plotted in Figure 5.4. The relevant input settings are summarized in Table 5.1. This figure shows the plasma potential, plasma density, electron temperatures and the mobility used by the code. Electron temperature is constant along each field line, as given by the solver model. Potential is seen to be approximately constant along each field line, with deviations arising due to plasma density variation. Of importance is a peak arising near the cathode line. In addition, the potential profile shows no significant variation outside the near-anode region. Both of these results seem to be non-physical. The flat profile results in low values of electric field and hence low acceleration of ions created in the annular ionization zone. The potential hill retards the ion motion even further. In addition, ions born in this region will have a tendency to be accelerated towards the anode. It is believed that this result is directly related to the reduced computational domain used by the electron solver due to lambda positioning issue identified in Section 5.2.1 and does not correspond to a physical feature.

5.2.3 Kinetic Code Inputs

Despite this issue, the HPHall simulation was used as a proof of concept study for coupling the kinetic code with an axial hybrid thruster simulation. The kinetic



(a) Potential

(b) Plasma density

(c) Electron temperature

(d) Mobility with NWC

Figure 5.4: HPHall simulation results obtained using the analytical mobility model. The overlaid mesh corresponds to the subdomain in which electron equations were solved.

code requires as inputs information related to the global state of the discharge. The λ mesh shown in this section is used only internally in HPHall, and is not exported in a format convenient for the kinetic analysis. Hence, a simple code was developed to contour HPHall results. These results are saved in a Tecplot file `2d\_ave\_tp.dat`. The code reads the Tecplot file into a two-dimensional array. Only results of interest, such as plasma density, temperature, and electric field components are imported. The code performs contouring the magnetic field lines using the marching squares technique. Contouring is performed for the λ values used internally in HPHall; these values are exported from HPHall simulation. Contouring starts by searching for the corresponding value along the bottom edge of the computational domain. Edge cuts are then determined by linear interpolation of node values, and the cuts are connected to form a spline. Properties of interest are then interpolated onto the spline control points. Collection of several splines results in the magnetic mesh shown in Figures

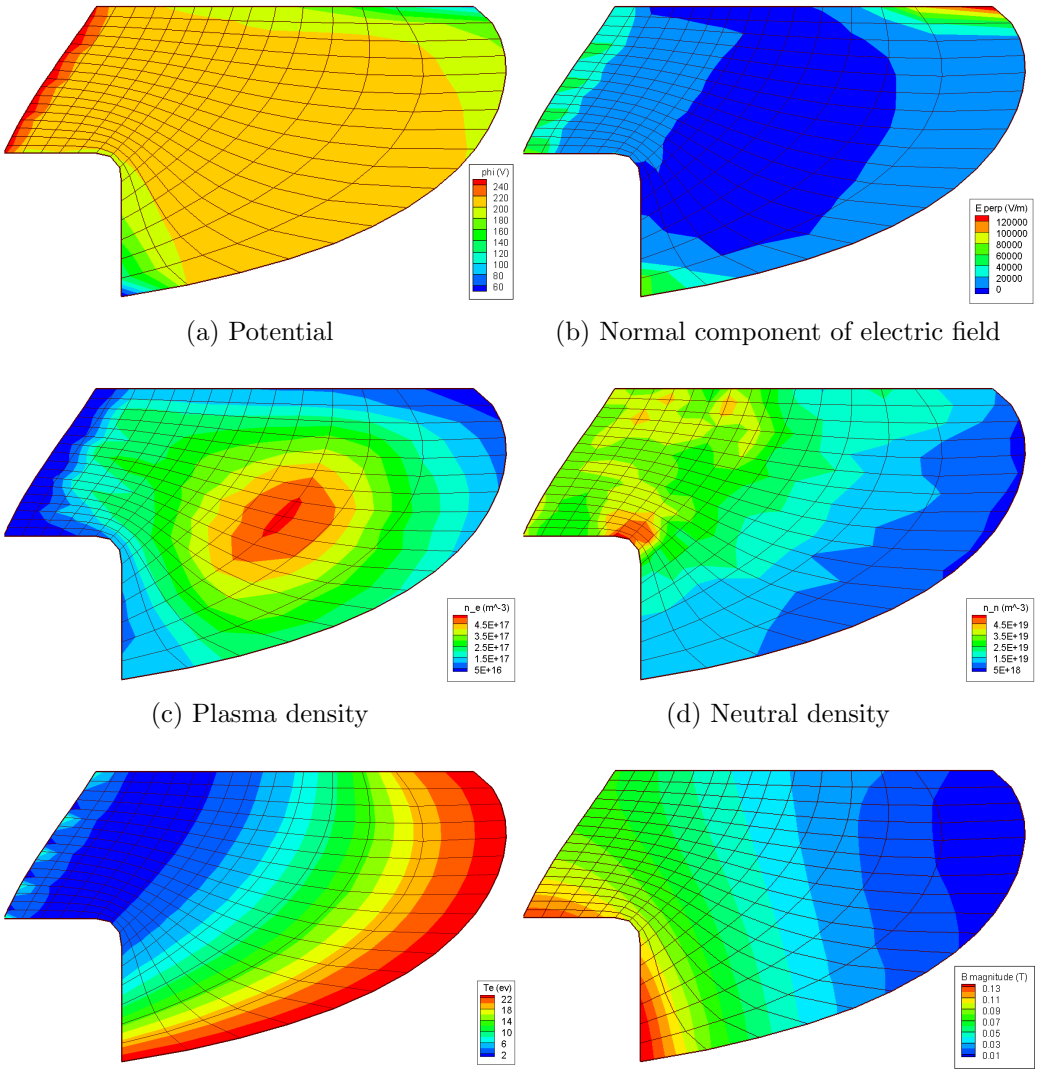
5.5 and 5.6 The first figure shows the inputs utilized by the kinetic code. The second figure shows properties of interest, such as mobility and ion normal velocity. As predicted from the inflected potential profile, ions are seen to travel back towards the anode in a region beyond the annular zone.

5.3 Kinetic Transport Calculation

5.3.1 Transport Analysis

The Lynx simulation was performed with 100,000 particles for total of 50,000 time steps. This large number of steps was used to investigate temporal evolution of computed properties. The simulation of the entire system took approximately one hour on an eight-core Windows workstation. Steady state was achieved in the first 1000 time steps, hence majority of effort was used to average the cumulative data. Results obtained by the simulation, interpolated onto the HPHall mesh are shown in Figure 5.7. In Lynx, potential is grounded at the outer wall with $\phi_{wall} = 0$. To aid in correlation with the HPHall data, potential was shifted according to $\phi_H = (\phi - \phi_{max}) + 200V$, where ϕ_{max} is the maximum potential along a field line. This result shows an interesting behavior. Despite the individual field lines being computed independently of each other, the results show a spatial coherence. This is particularly apparent in the potential valley occurring between field lines L2 and L6 (the field line indexing starts with 0 at the anode). Electron density also retains its prescribed shape, which can be seen by comparing Figure 5.7b with 5.5c. The two plots of mobility compare the mobility computed with the averaged drift velocity to the value obtained by averaging mobility computed with $E_\perp$ evaluated at the particle position. The two methods produce practically identical values.

Detailed variation in transport parameters along the field lines is shown in Figure 5.9. These plots show the variation along the four selected lines shown in Figure

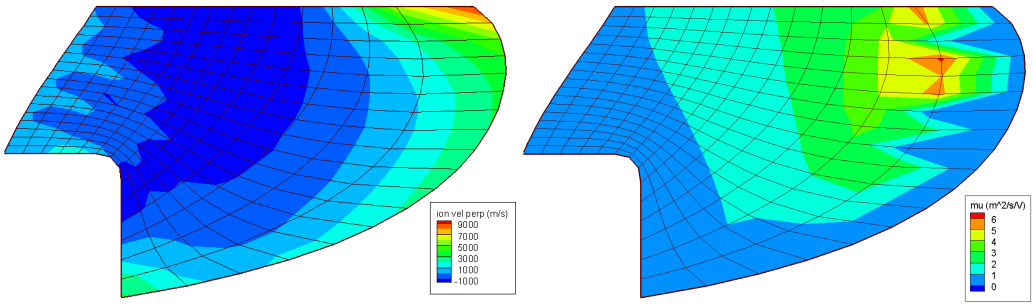


(a) Potential (b) Normal component of electric field

(c) Plasma density (d) Neutral density

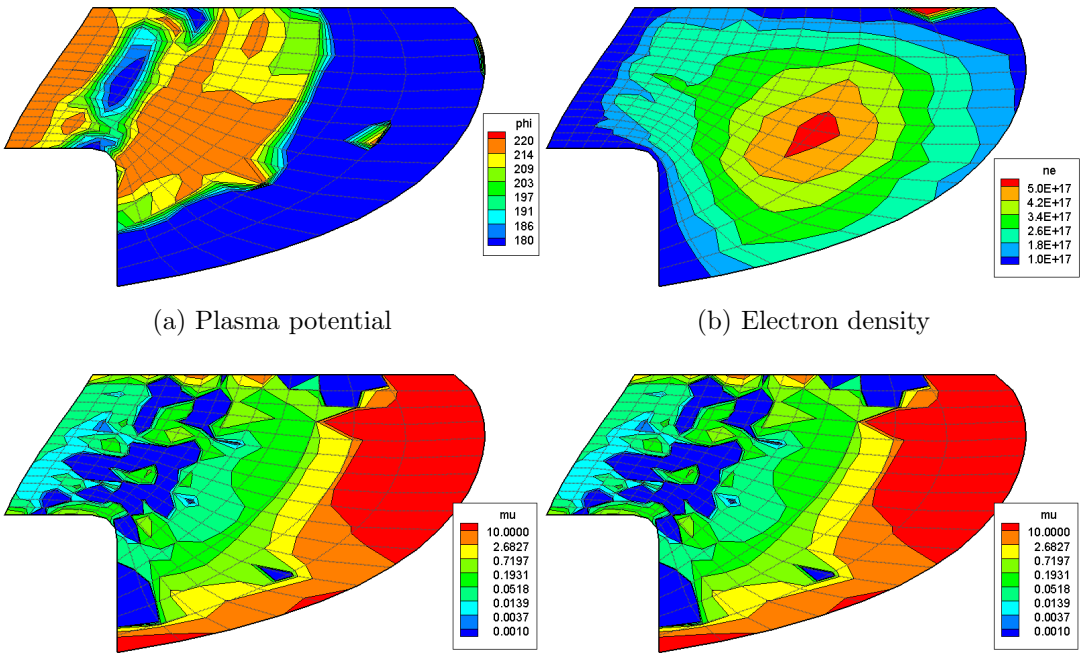
(e) Electron temperature (f) Magnetic field

Figure 5.5: Plasma parameters serving as inputs to the kinetic code. Kinetic simulation is performed for each field line (vertical grid lines). Additional input, which is not shown here, is the magnetic field strength.



(a) Perpendicular ion velocity (b) Mobility

Figure 5.6: Additional parameters interpolated onto the magnetic field line mesh. These parameters are not used by the kinetic code.



(a) Plasma potential (b) Electron density

(c) Mobility from $\overline{v_d}$ (d) Mobility from $\overline{\mu}$

Figure 5.7: Simulation results from kinetic Lynx simulations, interpolated onto the HPHall mesh

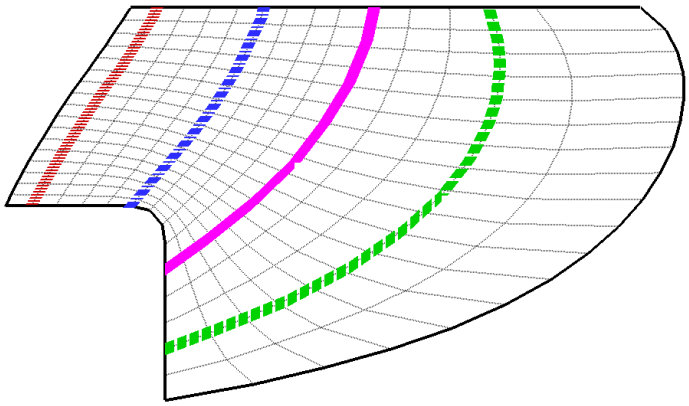
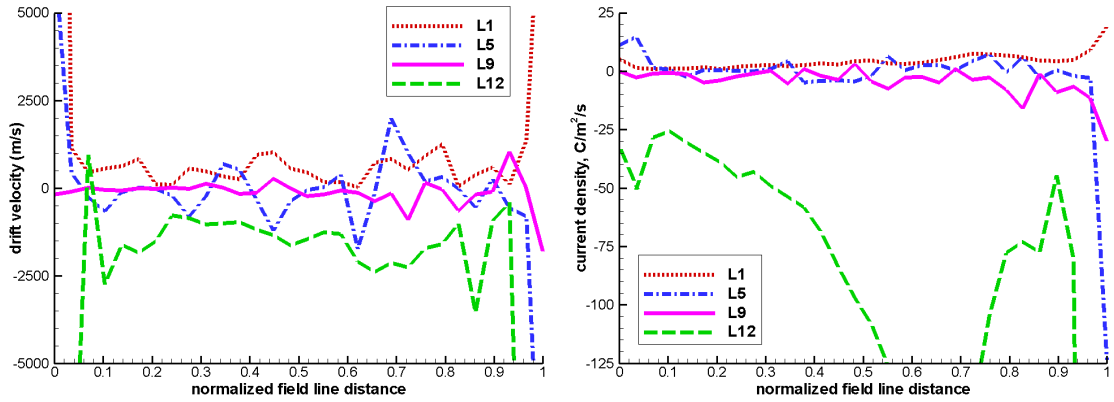


Figure 5.8: Magnetic field lines plotted in Figure 5.9

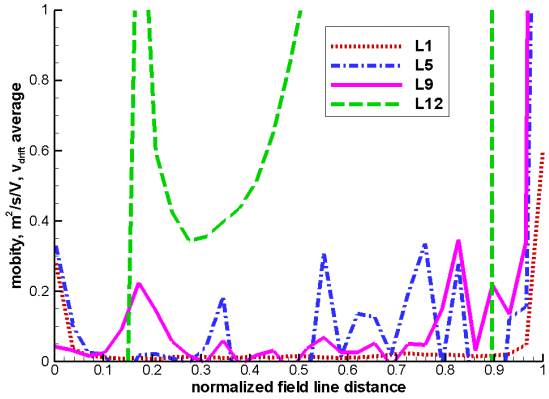
5.8. From left to right, these lines are L1, L5, L9, and L12. The drift velocities in general follow the trends expected by considering the prescribed electric field. Line L1 exists in a region with a a potential rise towards the anode. Line L5 and L9 both exist in a region in which HPHall predicted very little change in potential, and hence, these results indicate no significant drift. The L12 line crosses through the region containing the potential hill and an inverted potential profile. This case also corresponds to the largest value of current density due to the peak of electron density in this region. The second row compares mobilities computed using the two averaging methods, showing that they are in agreement. Finally, the last row plots the average guiding center positions and collision rates. Along the lines of the computed drift velocity, the guiding center is seen to have shifted towards the anode for L1 and in the anti-anode position for the L12 line. The other two lines existing in a region with low $E_\perp$ do not display a significant guiding center shift, indicating low transport and low drift velocity.

The temporal evolution in the cumulative current density is shown in Figure 5.10. This comparison was made to determine the effect of the number of simulation time steps on the obtained results. Even after 20,000 time steps, the code is seen to closely resolve the distribution shown at 50,000 steps. This finding indicates that in typical iterative studies foreseen once a full coupling of the kinetic and 2D code is achieved, a

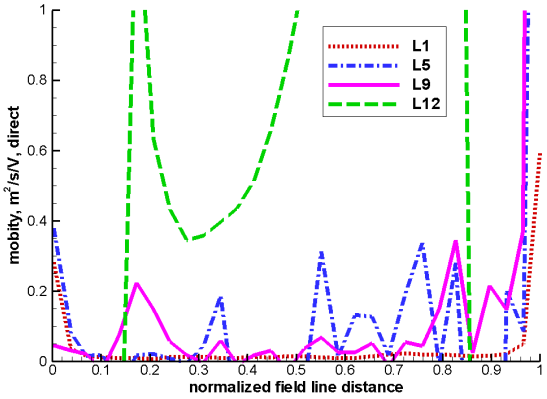


(a) Drift velocity

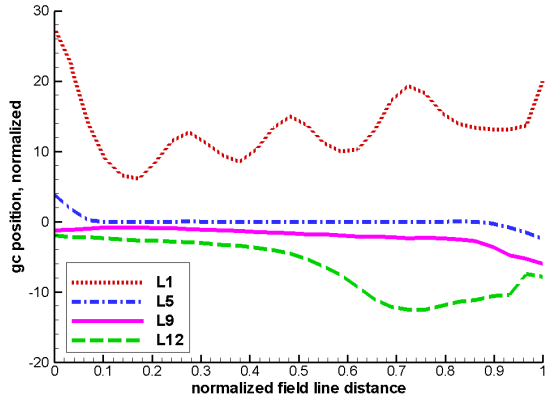
(b) Current density



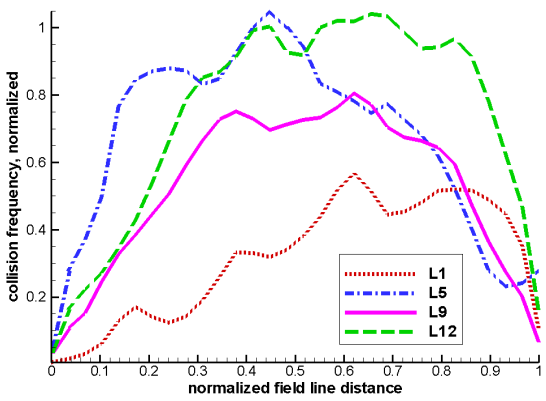
(c) Mobility from $\overline{v_d}$



(d) Mobility from $\overline{\mu}$

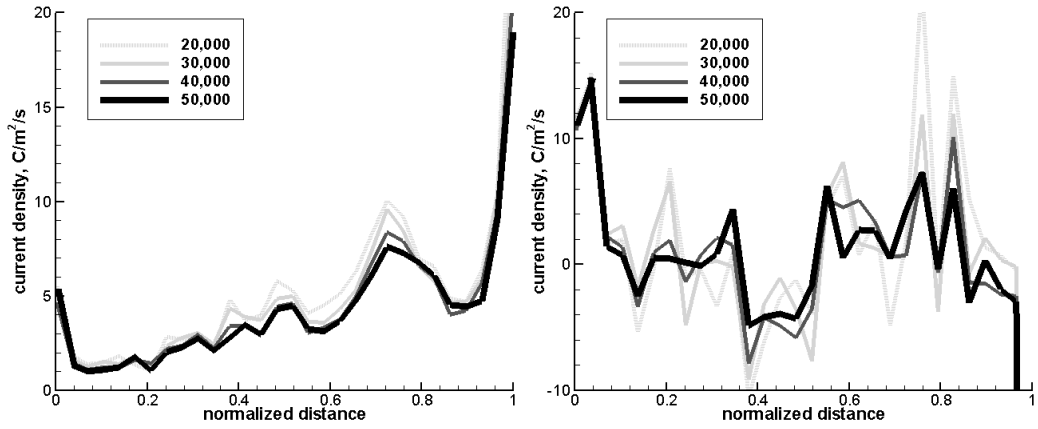


(e) Guiding center position



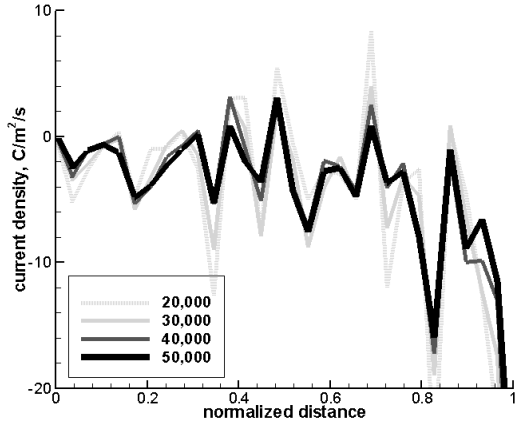
(f) Collisions rate

Figure 5.9: Simulation results along selected magnetic field lines. Guiding center is scaled by the local Larmor radius. Collisions frequency plot shows the total number of collisions, normalized by the maximum value.

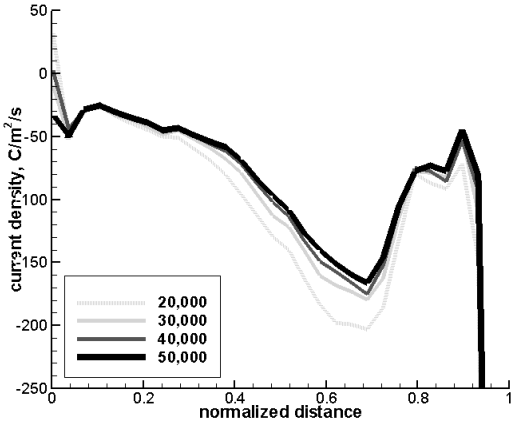


(a) L1

(b) L5



(c) L9



(d) L12

Figure 5.10: Temporal evolution in average current density for the four selected field lines

small number of steps may be sufficient. It's interesting to note that current density shows an oscillatory behavior. Such a behavior has been hypothesized to arrive from near wall conductivity and has been analyzed previously [67].

5.3.2 Results with Self-Consistent Mobility

In this work, an iterative two-way coupling between Lynx and HPHall was not attempted, instead only a single coupling back into HPHall was analyzed. This coupling was performed by loading the mobility map obtained in Lynx into HPHall. To accomplish this, HPHall was modified to allow loading of a fixed mobility map

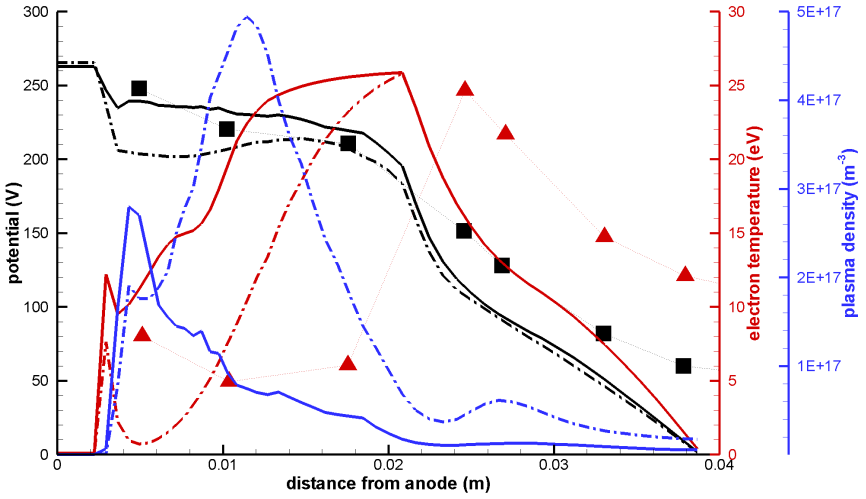


Figure 5.11: Plot of several plasma parameters along the thruster centerline. The markers correspond to experimental data from [64]. The results computed using the analytical mobility model are shown using the dashed lines. The results obtained with the kinetic mobility are shown with the solid lines.

from a file. HPHall was subsequently restarted and run for additional 5,000 time steps. Comparison of centerline plasma properties can be seen in Figure 5.11. Solid markers correspond to experimental data. The dashed lines are the results obtained with the classical model, and the solid lines are the ones obtained by loading the self-consistently determined mobility. A slight improvement in plasma potential is seen near the anode, with the potential dip disappearing. As expected no change is seen in the cylindrical region. Temperature results are less conclusive, partly due to the poor agreement of the initial results with the data. The peak of plasma density is seen to decrease and move towards the anode.

5.4 Particle Sampling

The HPHall simulation was also used to obtain inputs to the plume code, similar to what is described in [68]. The open design and the self-induced electric fields that arise in Hall thrusters make it difficult to analytically describe the velocity and flux distribution function of ions exiting the thrusters. Sampling kinetic particles as

they cross a prescribed plane is a simple and effective way to describe the velocity space. By sampling velocities along with the corresponding spatial position for a sufficiently large number of particles, a discretized velocity distribution function can be obtained. This approach can also be used to directly estimate the thrust. The sampling boundary used in this work is shown in Figure 5.12 by the dashed line. As can be seen here, this boundary is located downstream of the exit plane. A peculiar feature of Hall thrusters is that ions continue to be accelerated outside the device. In the case of the CTH, the external acceleration accounts for approximately 50 eV of ion energy. Since plume codes do not resolve the near-thruster region, it is necessary to capture the entire acceleration profile in the particle data. As such, it is necessary to sample the particles a sufficient distance way from the exit plane, in a region where potential drop becomes negligible and velocity reaches its peak value. In HPHall, the sampling is performed by saving particle information to a file in the Tecplot file format. An example of the sampled data is shown below. As can be seen from this data, and also from the contours in Figure 5.12, this particular simulation resulted in ion beam velocities ~ 12 km/s. This value underpredicts the expected value, $v_i = \sqrt{2e\Delta U/m_{Xe+}} = 19$ km/s with $\Delta U = 250$ V and corresponds to only approximately 100 V of potential drop. This low value is due to the previously described difficulties in utilizing HPHall with this thruster geometry. In order to obtain a more realistic plume profile for the analysis presented in Chapter 7, an artificial $\delta w = 5$ km/s was applied to the sampled data. Such an adjustment is not necessary with correctly modeled thruster discharges. As an example, a similar particle sampling approach applied to a standard annular thruster geometry resulted in particle exit velocities following the predicted velocity, and a significant improvement in plume comparison with experimental data [52].

```
VARIABLES = z r vz vr vt t
ZONE T=HPHALL_XE+
```

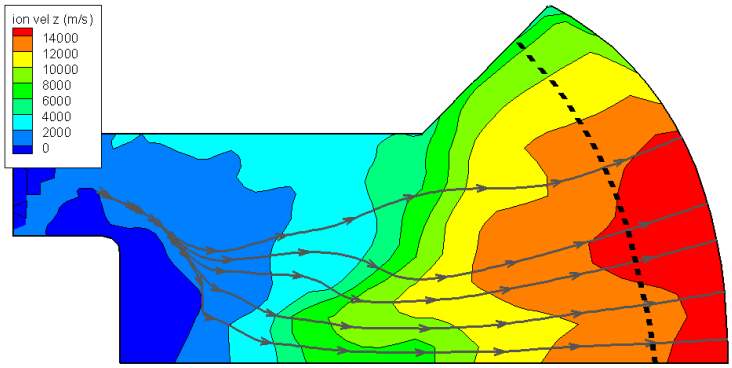


Figure 5.12: Ions crossing the indicate mesh line are sampled to obtain a discretized source model.

```
0.0560 0.00285  9953.1  1260.5  858.9 4.00e-008

0.0552 0.00726  5727.3  4361.5   12.3 4.20e-007

0.0560 0.00306 12252.2 -1955.4 2030.7 5.45e-007

0.0559 0.00349 12874.8  4233.8 1375.6 5.55e-007

0.0555 0.00603 15138.7 -2407.2  846.8 6.00e-007

0.0556 0.00546  8375.1  1064.4  597.9 6.50e-007

0.0550 0.00847  5096.3  2531.0  -29.0 6.65e-007

0.0562 0.00114 14118.7 -1332.2 2391.0 7.05e-007
```

Chapter 6

Plasma Wall Transition in the Presence of an Inclined Magnetic Field

6.1 Introduction

Besides obtaining inputs for a plume simulation, the axisymmetric model can also be used to study the interaction of ions with thruster walls. This interaction influences both the efficiency of the thruster as well as the lifetime. The next aspect of the multiscale approach deals with the near wall region, especially as applicable to novel thruster geometries. In the classical Hall thruster, the magnetic field consists primarily of the radial component. Such a configuration appears ideal at first since it produces an electric field directed along the thruster axis. However, the presence of walls modifies the near-wall potential structure and results in a local component accelerating ions into the walls. Ion wall flux contributes to a loss of thruster efficiency and to a limited thruster lifetime due to channel erosion. In order to mitigate these losses, some novel Hall thrusters [3, 69] have begun experimenting with magnetic fields with convex geometry. Near the walls, this so-called magnetic lens induces an electric field with a radial component directed towards the channel centerline [70]. An interesting aspect of the lens configuration is that in the vicinity of the wall, the resulting magnetic field lines can approach the wall with a highly inclined

incidence angle θ, as measured from the wall normal. Such a configuration leads to an electric field with a strong radial term that, in the case of a sufficiently large θ, dominates the component due to the sheath potential drop [4]. This can be seen from a simple example. Consider a typical 300V Hall thruster with a 200V potential drop occurring across a 1 cm wide acceleration zone. The magnitude of the electric field $E_\perp$ is then 2×10^4 V/m. Next consider the potential drop due to the wall sheath. The electric field along the magnetic field line in the vicinity of the wall can be estimated from $E_\parallel = T_e \partial \ln n / \partial r \sim T_e / \Delta r \sim 20$ eV / 0.1cm $\sim 2 \times 10^5$ V/m [71]. Here Δr is the sheath thickness, which is taken to be 10 Debye lengths. The angle at which the radial component of the electric field becomes negative is given by $E_r = E_\parallel \cos\theta - E_\perp \sin\theta \sim \tan\theta = E_\parallel / E_\perp$ or $\theta \sim 85°$.

Ions are then accelerated away from the wall and a complete sheath collapse is expected. Although plasma-wall transition has been subject of much past research, such a research typically considered only the generalized radial case [53]. This chapter investigates the sheath formation and collapse in the presence of a two dimensional magnetic field. This analysis is performed using a simple axisymmetric electrostatic particle-in-cell (ES-PIC) code. The code is used to determine the structure of the plasma sheath for several magnetic field configurations. For simplicity, the analysis presented in this chapter is performed with an analytical potential drop and magnetic field line topology. In an actual coupled analysis, values obtained by the thruster simulation in the near-wall bulk region would be used to define the problem.

6.2 Computational Model

6.2.1 Simulation Domain

The code is based on the hybrid approach in which ions are treated as particles, but electrons are represented by a fluid model. The computational domain is limited

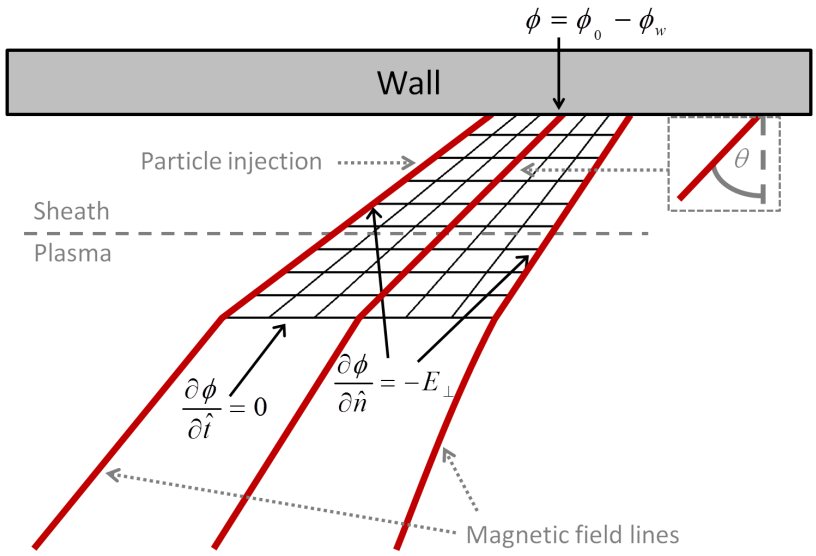


Figure 6.1: Schematic of the computational domain. Ion particles are injected from the left. The inset shows a cylindrical Hall thruster and highlights the region analyzed by our code.

to a small region near the outer wall, as illustrated in Figure 6.1. The small size of the computational domain allows the code to resolve the Debye length and thus directly compute the electric potential in a reasonable amount of time (each simulation takes approximately 30 minutes). The domain captures the acceleration region characterized by the presence of strong applied magnetic field. In this formulation, the anode and the primary ionization zone are located to the left. The upper boundary represents the wall, while the bottom boundary extends into the quasineutral bulk plasma region. Ions are injected into the simulation along the left boundary and leave through the open right and bottom face or by recombining with the upper wall. The inset in Figure 6.1 is an example of a hybrid annular/cylindrical Hall thruster such as the CHT in which the magnetic field geometries of interest can be found. The highlighted box illustrates the location of the simulation domain. It should be noted that the size of the region of interest is artificially increased in this drawing for clarity.

To simplify the subsequent computation, a simulation mesh is selected such that radial gridlines are aligned with the magnetic field. Such a formulation allows us

to specify the necessary reference values as a function of the axial grid coordinate only. In constructing the mesh, attention was paid to two requirements. First, the mesh had to be capable of capturing the magnetic topology of interest: varying angle of magnetic field, and also the convergence of magnetic field lines in a magnetic mirror. Secondly, the mesh had to be suitable from the computational perspective. Particle methods require scattering of particles to the grid nodes, and conversely gathering forces by collecting values from the grid onto particle locations. Topologically structured meshes are preferred here, since physical coordinates can be mapped to the computational space via evaluation of analytical functions. The mesh shown in Figure 6.1 satisfies both of these requirements. The mesh coordinates are given by

$$r = r_0 + j * \Delta r \tag{6.1}$$

$$z = i * \Delta z_j - (nr - 1 - j) * \Delta r * \tan(\theta) - 0.5 * (nz - 1) * (\Delta nz_j - \Delta z_w) \tag{6.2}$$

where Δz_j is the local cell spacing. The cell spacing varies linearly between the top and bottom boundary. These mesh coordinates can be easily inverted. The j component is obtained first from the radial r coordinate. The i coordinate is then recovered from the axial position z using the second equation.

6.2.2 Particle Injection

Xenon ions are injected into the simulation domain along the left boundary with initial velocity $u_z = u_0 + u_{th}$. Here u_0 is the drift component and uth is a random thermal velocity obtained by sampling the Maxwellian distribution function at 1 eV. The magnitude of the drift component was set to 6 km/s, corresponding to approximately 25 eV of upstream acceleration. Initial radial velocity is also obtained by sampling the random thermal component. The number of computational particles injected per time step is obtained from $p = \dot{m}\Delta t/w = n_i \bar{u} A m \Delta t/w$ where $n_i =$

5×10^{16} m^{-3} is the injection ion density, and w is the macroparticle weight. The weight was selected such that cells in the bulk region contained approximately 200 computational particles at steady state. Particles were loaded with a zero azimuthal component. We assume that no forces act in the azimuthal direction and hence the cylindrical equations of motion reduce to the Cartesian form. Ion positions are updated at each time step according to the Leapfrog algorithm by integrating the Lorentz force, $\vec{F} = -e\nabla\phi$. The magnetic term is omitted, since in a Hall thruster, ions are not magnetized. Ions impacting the upper wall or leaving the computational domain were removed from the simulation. Collisions were not included as they generally play only a minor role in the sheath.

6.2.3 Potential Solver

Potential was computed by solving the Poisson's equation, $\epsilon_0\nabla^2\phi = -e(n_i - n_e - n_s)$, with the three densities on the right side corresponding to ions, primary electrons, and secondary electrons, respectively. The ion density n_i was obtained by scattering positions of kinetic ions to the computational grid. The electron density is computed from the Boltzmann equation following the approach in [47] and [72]. In the frame of reference of ions, electrons respond instantaneously to a disturbance. The time-dependent and convective terms then vanish from the momentum equation. Also, since in Hall thrusters the sheaths is generally collisionless, we can disregard the collision operator. We thus arrive at the force balance,

$$\frac{\partial}{\partial x}(n_e k T_{e\|}) + \frac{n_e(kT_{e,\perp} - kT_{e\|})}{B}\frac{\partial B}{\partial s} + n_e e E_\| = 0 \tag{6.3}$$

These terms correspond to the gas pressure, magnetic mirror, and electric field effects, respectively. Utilizing $E_\| = -\partial\phi/\partial s$, the above equation can be integrated to

obtain an expression for bulk electron density,

$$n_e = n_0 \exp\left[\frac{e}{kT_{e\|}}(\phi - \phi_0) - \frac{kT_{e\perp} - kT_{e\|}}{kT_{e\|}} \ln\left(\frac{B}{B_0}\right)\right] \tag{6.4}$$

This is the well known Boltzmann relationship modified by the magnetic field strength term. This term is seen to reduce the electron density in regions of an increasing magnetic field this is the magnetic mirror effect. The standard Boltzmann relationship is recovered if the magnetic field magnitude remains constant along the field lines. The magnetic mirror term also drops out if plasma is isothermal. However, as outlined in [53], Hall thruster plasma is not isothermal. Following the results from the kinetic analysis in [53], $kT_{\|} = 10$ eV and $kT_{\perp} = 2kT_{e\|} = 20$ eV is used. For this particular set of input parameters, Equation 6.4 simplifies to $n_e = n_0(B_0/B)\exp[e(\phi - \phi_0)/kT_{e\|}]$.

It should be noted that Equation 6.4 holds independently for each magnetic field line. The three constants with the 0 subscript are the reference density, potential, and magnetic field strength. These values are unique and independent along each line. We also assume that electron temperature remains constant in the parallel direction, $\partial T_e/\partial s = 0$, and that there is no variation in magnetic field strength in the axial direction, $\partial B/\partial z = 0$. The reference density is obtained self-consistently from the computed ion density along the bottom edge of the simulation domain where $n_i = n_e = n_0$. A linear decay in potential is applied for majority of cases, with $\phi_0 = \phi_L - E_{\perp}(z_w - z_{w,0})$, where $E_{\perp} = 20$ kV/m. Since in Hall thrusters the potential profile adjusts self-consistently based on the local discharge parameters, an alternate cusp configuration is also investigated in which the potential profile exhibits a deep valley. This setup is discussed in more detail in Section 6.3.4. The strength of the magnetic field is computed from the conservation of magnetic flux, $\phi_m = \int_S \vec{B} \cdot \vec{ds}$ or $Br\Delta z = C$, a constant value. Here Δz is the cell spacing at the corresponding r

value. As indicated by Equation 6.4, terms relating to the magnetic strength appear only as a ratio allowing us to select an arbitrary value for the reference field.

The secondary electron density n_s is obtained from $\nabla \cdot (n\vec{u}) = 0$. Density of secondary electrons at the wall is given by $n_{s,w} = sn_{e,w}$ where $s(T_e, \theta)$ is the SEE yield [71]. The electrons are assumed to be emitted with an isotropic angular distribution and energy dependence based on the linear relationship given by Dunaevsky [73],

$$s(T_e, \theta) \approx \sigma_0 + (1 - \sigma_0)\frac{E_p}{E_1} \tag{6.5}$$

For Boron Nitride, the typical wall material in conventional Hall thrusters, the coefficients σ_0 and E_1 are 0.54 and 40, respectively. E_p is the energy of the incoming particle, measured in eV. Initial velocity of the secondary electrons is taken to be $u_{s,w} = (2kT_w/\pi m_e)^{1/2}$. Energy conservation dictates $u = (2q\Delta\phi/m)^{1/2}$, leading to

$$n_s = sn_{e,w}\left(\frac{kT_w}{e\pi}\frac{1}{\phi - \phi_w}\right) \tag{6.6}$$

6.2.4 Boundary Conditions

Potential along the top wall is fixed as $\phi_w = \phi_0 - \Delta\phi_w$, where the wall potential drop is given by [71] as

$$\Delta\phi_w = T_e \ln\left\{\frac{1 - s(T_e, \theta)}{v_0\left(\frac{2\pi m}{T_e}\right)}\right\} \tag{6.7}$$

where v_0 is the ion velocity at the sheath edge, which in this formulation is set to the Bohm speed. The problem is closed by prescribing the normal electric field $E_\perp$ along the left and right boundaries, and zero tangential electric field $E_\perp = 0$ on the bottom boundary. The electric field along the left and right boundaries is non-uniform for cases with a diverging magnetic field line topology. This can be seen from a simple observation of the increasing distance between field lines as one moves away from the wall. The magnitude is obtained numerically by computing the normal distance d to

the next magnetic field line (grid line) at each node. The electric field is then set from $E_\perp = -\Delta\phi_0/d$. Potential is solved using the finite volume method.

6.2.5 Quasineutral Solver

In addition to the Poisson solver, an alternate method for obtaining potential was also implemented. This approach was developed in order to approximate the solution from Hall thruster codes such as HPHall. As described previously, HPHall does not solve the Poisson's equation, and instead computes radial potential by assuming quasineutrality, $n_e = n_i$ (with ion density obtained from kinetic ions) in conjunction with the thermalized potential model, $\phi = \phi^* + kT_e/e \ln(n/n_0)$. This approach is analogous to the formulation used to derive the relationship for bulk electron density, Equation 6.4. This expression can be inverted to obtain

$$\phi = \phi_0 + \frac{kT_{e\parallel}}{e} \ln\left(\frac{n}{n_0}\right) + \frac{kT_{e\perp} - kT_{e\parallel}}{e} \ln\left(\frac{B}{B_0}\right) \tag{6.8}$$

where $n_e = n_i = n$. Our expression extends the quasineutral formulation by taking into account the magnetic mirror term.

6.2.6 Implementation

Simulations were performed on a domain with 50 cells in the axial and 30 cells in the radial direction. Cell spacing was set to $\sim \lambda_D = 10^(-4)$ m. The simulation time step was adjusted automatically by the code from its initial value of 1.5×10^{-9} s such that ion particles traveled no more than 0.33 cell lengths per time step. The thruster diameter was assumed to be 6 cm. The simulation started by pre-filling the domain by injecting and propagating ion particles under the initial electric field. This fast pre-fill was found to improve the subsequent solver convergence rate. The simulation then continued in the normal mode in which the electric field was

updated at each time step. The simulation continued until steady state characterized by approximately zero net change in particle counts between successive iterations. The simulation then continued for additional 2000 time steps during which results were averaged. The typical number of computational particles at steady state was 700,000. Simulation results, including potential, number densities, particle velocities, and wall fluxes were then exported. A marching squares algorithm was implemented to automatically contour the resulting velocity map to obtain the sheath boundary. In this work, we defined the sheath boundary as the contour where the radial component of velocity $v = v_B$, the Bohm velocity. The code was implemented in the Java programming language and was run on a Dell Precision workstation with eight CPU cores. Each simulation was launched as an independent thread, and a simple scheduler was implemented to allow concurrent execution of the simulation cases.

6.3 Results

6.3.1 Potential Distribution at Uniform Density

Often it is possible to obtain useful insight into the solution by considering a simplified case that can be evaluated in a reduced computation time. In this case, it is possible to investigate the potential distribution that forms in the presence of a completely uniform plasma. These results are illustrated in Figure 6.2. In all cases, plasma density of 5×10^{16} m^{-3} was used. The contours correspond to the lines of constant potential and the streamlines visualize the electric field. The classical Hall thruster with a solely radial magnetic field is shown in Figure 6.2a. As indicated previously, this configuration results in a primarily axial electric field. However, near the wall, the sheath potential drop modifies the electric field structure such that the electric field becomes oriented towards the wall. Ions located in this near wall region are then expected to be lost to the wall recombination. Figure 6.2b illustrates what

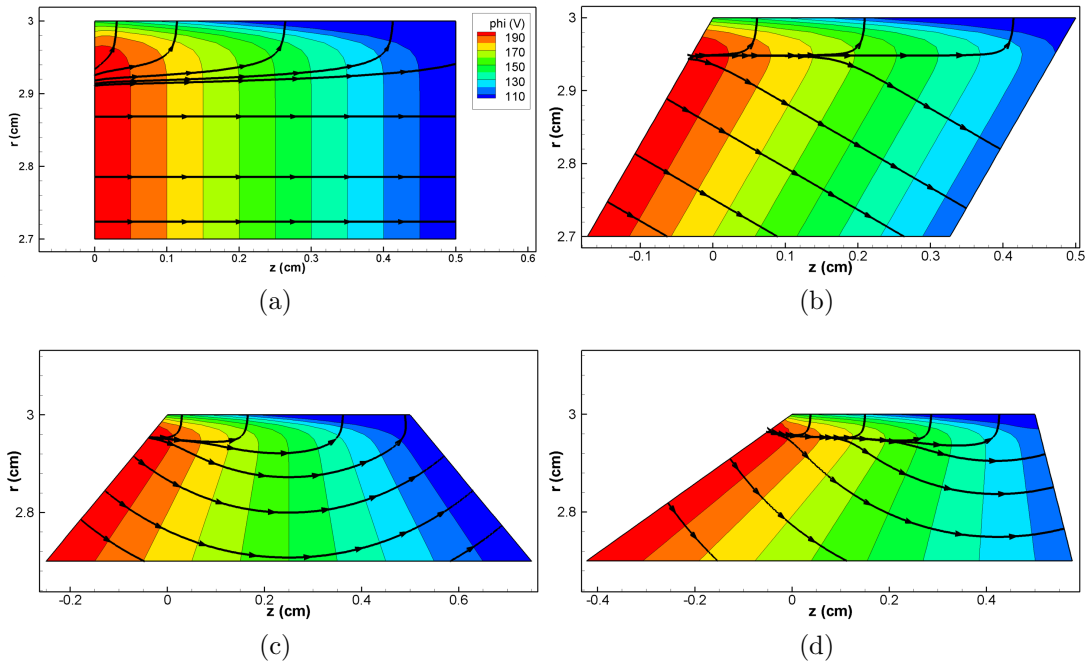


Figure 6.2: Potential with uniform plasma density. a) $\theta = 0°$, $B_w/B_0 = 1$ b) $\theta = 30°$, $B_w/B_0 = 1$ c) $\theta = 0°$, $B_w/B_0 = 2$ d) $\theta = 30°$, $B_w/B_0 = 2$

happens when the magnetic field angle is increased to 30°. Increase of the magnetic field incidence angle results in a compression of the region containing the radial electric field. Analogously, the critical streamline delineating the near wall region from the bulk acceleration zone moves closer to the wall.

Cases c) and d) show the effect of the magnetic lens. The net angle of the magnetic field is zero in case c), however, the magnetic field strength decreases away from the walls. The solution is approximately antisymmetric. The electric field is seen to first accelerate the ions away from the wall. The field subsequently acts to direct the ions back towards the wall. However, since ion axial velocity is increased by the electric field, the net radial deflection will be smaller in this diverging section. The predicted result is a net acceleration of ions away from the wall. Case (d) extends the magnetic mirror effect in (c) by including the 30° field inclination from (b). By comparing the electric field stream traces between cases (c) and (d), we can see that one of the effects of this combined configuration is to increase the acceleration of ions away from

the wall. This effect is due to the change in the magnetic field geometry.

6.3.2 Sheath variation with magnetic field angle

These initial observations can be expanded by performing series of plasma simulations. First, the effect of an increasing magnetic field angle in the absence of a magnetic mirror is investigated. Secondary electron emission was not included in this set, $s = 0$ in Equation 6.6. The magnetic field angle θ increased from 0° (magnetic field normal to the wall) to 85° (highly inclined configuration). Results for 0°, 40°, 60° and 70° are plotted in Figure 6.3. Ion densities are shown using the contour plot. Velocity streamlines are also plotted, as well as the sheath boundary. The sheath edge is plotted by the solid red line and corresponds to the contour where the radial velocity component (i.e. the component normal to the wall) reaches the Bohm velocity $v_B = \sqrt{kT_e/m_i}$. We can see that in the case of a zero magnetic field angle, the solution obtained by the code is similar to the well-known boundary layer problem. The sheath forms a short distance from the injection plane and continues to grow as more ions are accelerated from the bulk plasma towards the wall. Plasma density decrease is also influenced by the net increase in ion velocity due to the axial electric field. The result along the left boundary is somewhat non-physical, since in a real device, the sheath thickness will be finite at the entrance to the acceleration zone. Since the simulation resolves only a small subset of the Hall thruster channel, it is unable to capture the sheath that forms upstream of the domain. To investigate the role the initial sheath profile has on results, a modified particle loading algorithm was tested in which the injection density decayed exponentially towards the wall. The differences between the two solutions was found to be limited to several cells near the injection boundary. This finding can be explained by realizing that any ions injected into the sheath will be rapidly lost to the wall. Hence, this initial region is ignored and the sheath is characterized by its maximum thickness.

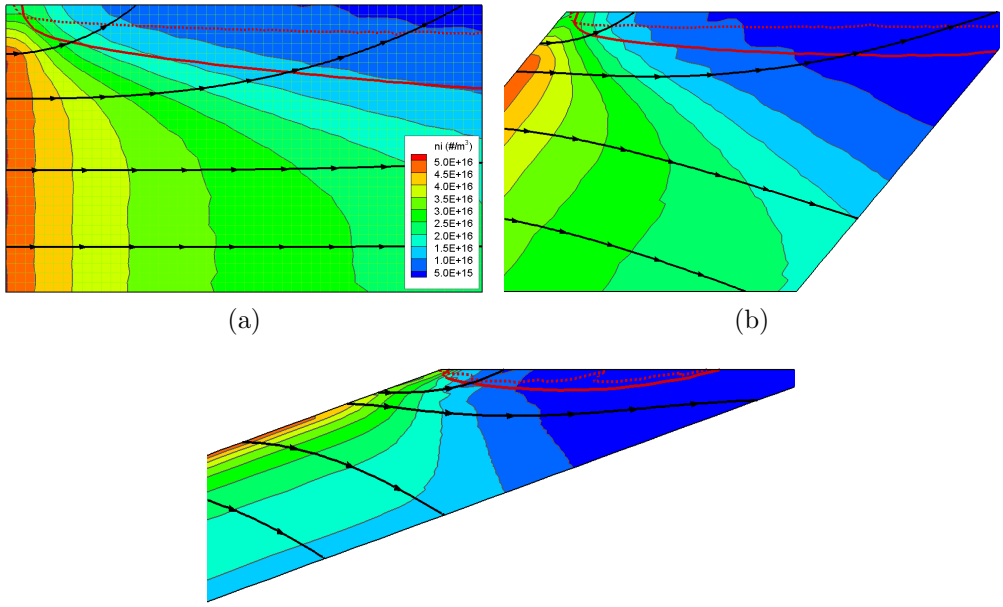


Figure 6.3: Simulation results showing the ion density profile for three different magnetic field line angles, 0°, 40°, and 70°, respectively. Streamtraces show ion trajectories. The red lines correspond to the sheath edge as computed with the Poisson solver (solid) and the QN model (dashed line).

From this simulation we see that the effect of an increasing magnetic field angle is a reduction in the sheath thickness, as stipulated in the previous section. At 40°, the sheath thickness asymptotes to a constant value. Furthermore, ion density at the sheath edge has been reduced due to the acceleration of ions away from the wall by the electric field. At 70° we start seeing the first evidence of a sheath collapse. The sheath has reduced in axial size and extends only over a small section of the wall. The remaining section of the wall demonstrates a collapsed sheath. In this region, ion velocity streamlines become parallel to the wall. Instead of being accelerated into the wall, ions are seen to travel on a trajectory along the wall. This effect is not seen in configurations with a smaller magnetic field inclination angle θ.

This study also looked at the effect the potential solver plays on the sheath solution. The sheath profile indicated by the dashed red line corresponds to the solution obtained using the simplified quasineutral approach. We can see that the sheath

thickness is artificially compressed, and extends to approximately three computational cells. For illustration, the computational mesh is displayed in the top figure. Since the QN approach does not actually resolve the sheath, the potential drop is driven by local deviation of ion density from the reference value. Due to the numerical approach, the applied wall potential drop extends exactly across a distance equal to one cell length. The QN solution is non-physical, since the magnitude of the near-wall electric field is directly related to cell spacing. This is in contrast with the solution from the Poisson solver, which predicts the sheath potential drop to extend over the distance of several Debye lengths and is independent of the mesh (assuming sufficient resolution). In the QN approach, the wall effect is communicated into several additional cells by ion dynamics. Ions located in the cell attached to the wall will be accelerated into the wall, resulting in a net decrease in ion density on the $j_{wall} - 1$ node. This decrease then results in a corresponding decrease in potential, Equation 6.8. Since this node is shared with the subsequent cell, ions located in that cell will also be affected by the wall. However, this ion-driven response has only a limited capability to communicate the wall potential drop to the bulk plasma. In general, it is found that this numerically-driven transport is effective only over the distance of several cells. As such, the QN approach under-predicts the ion-attractive capability of the wall. This effect results in the relative reduction of the sheath thickness.

6.3.3 Influence of Secondary Electron Emission

The next investigation looks at the effect of secondary electron emission (SEE). As note previously, SEE may be an important driver in the so-called anomalous electron transport across magnetic field lines. Here only the role of SEE on the sheath profile is considered. Same set of cases presented in the previous paragraph was run with the wall potential modified by the presence of SEE. We include secondary electrons in our code by computing the SEE emission coefficient using Equation 6.5. From

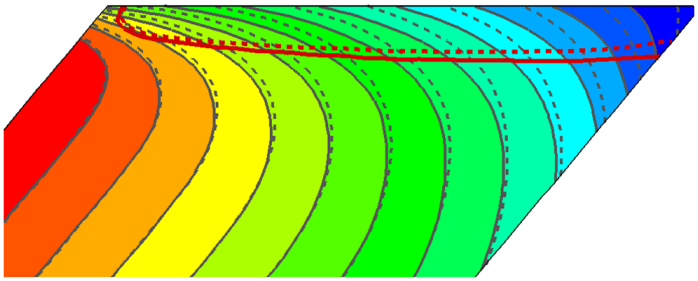
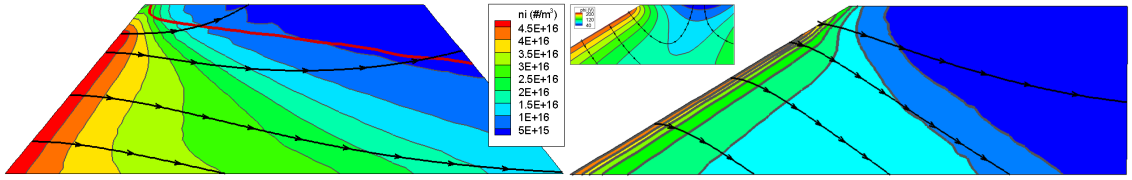


Figure 6.4: Potential profile for 40° magnetic field line inclination. Dashed lines indicate solution with secondary electron emission.

the wall potential relationship, Equation 6.7, we can see that the presence of SEE acts to decrease the sheath potential and hence the sheath thickness. This prediction is confirmed by the result illustrated in Figure 6.4. This figure shows the potential contours for the 40° magnetic field inclination. The dashed lines correspond to the case with secondary electron emission. Presence of SEE is seen to reduce the sheath thickness.

6.3.4 Magnetic Mirror Effect

The next set of simulations was run to investigate the effect of magnetic mirror. The mirror ratio $R_m = B_w/B_0 = 2$ was used, and cases without and with a 40°magnetic field inclination were compared. The role of the prescribed wall potential profile was also investigated. Figure 6.5 indicates that the mirror plays a role similar to that of the inclined field. Convergence of field lines near the wall results in an inclined electric field analogous to the pure field rotation, Figure 6.3b. We can however see some evidence of the defocusing effect in the diverging section. While under the pure rotation, the velocity streamlines in the bulk plasma are directed towards the channel centerline, in this configuration we see the ion trajectories to follow path parallel to the thruster axis. This is the preferred behavior, since the radial beam component does not contribute to thrust. Large plume divergence also leads to possible spacecraft contamination risks. Figure 6.5b plots the solution obtained by including a 40°



(a) (b)

Figure 6.5: Ion density contours in the presence of magnetic mirror. Magnetic mirror strength of 2 is used in both cases. Case (b) includes a 40o magnetic field inclination and a potential well. The inset shows the potential, with the contours ranging from 200V to 40V.

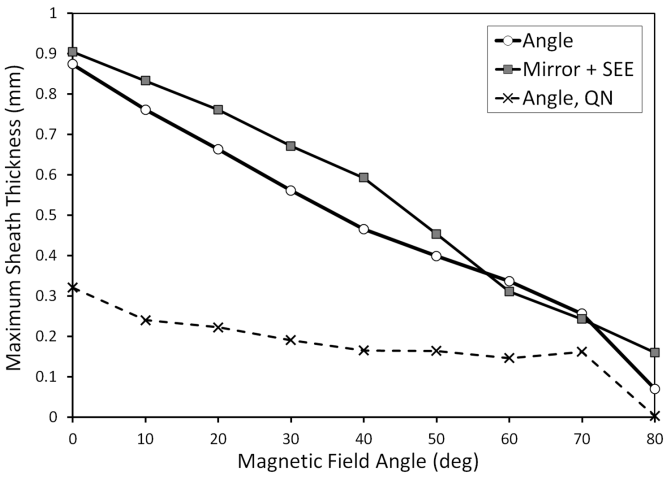
magnetic field inclination and a potential "valley" that is known to occur in devices utilizing cusped magnetic profiles. In the CHT, such configurations arise from the difference in the physical location of the inner and outer magnets and the details of the magnetic circuitry. The potential valley was modeled by superimposing a parabolic potential drop over the linear decay, $\phi_0 = \phi_L - E_\perp(z_w - z_{w,0} + \Delta\phi_C\,[4(\hat{z} - 0.5)^2 - 1]$, where $\hat{z} = (z_w - z_{w,0})/L_z$ is the normalized distance and $\Delta\phi_C = E_\perp L_z$ is the potential drop in the cusp selected to equal the potential drop in the linear region. The potential profile at steady state is shown in the inset in Figure 6.5b. It can be seen that although an electric field forms directing ions into the potential well, this profile has only a negligible effect on the fast moving bulk ions. The primary effect of the potential well is to further increase the electric field accelerating the ions and ions are accelerated towards the channel centerline. It should be noted that this particular example does not take into account the dynamic nature of Hall thruster discharges. Fluctuations in the structure of the internal discharge could lead to the presence of slowly moving ions in the vicinity of the well, and these ions would subsequently be accelerated into the wall.

6.4 Discussion

6.4.1 Sheath Collapse

Figure 6.6 plots the variation in maximum sheath thickness with the incidence magnetic field angle. Three cases are considered: inclined magnetic field, inclined magnetic field with a magnetic mirror and SEE, and an inclined field computed using the quasineutral approach. Linear potential drop was used in all cases. It can be seen that in all cases the maximum sheath thickness decreases as the magnetic field incidence angle is increased. Figure 6.3 shows that at 70° the sheath surrounds only a small portion of the wall. From Figure 6.5 it can also be seen that the sheath has reduced in maximum thickness by 70% from the value obtained at 0°. A sharp decrease in thickness at $\theta > 70°$ indicates the onset of a sheath collapse. The sheath thickness obtained in the presence of magnetic mirror generally follows the trend of pure field inclination. The slight increase in the thickness with the mirror may be purely a simulation artifact due to the larger domain dimension in the presence of the mirror. Of greater interest is the solution obtained using the QN approach. This resulted is plotted using the dashed line. The QN method is seen to under-predicts the sheath thickness by a factor of 3. In addition, the QN approach also exhibits a reduced dependence on the applied magnetic field angle. The sheath thickness remains approximately uniform and equal to $2\Delta r$ until $\theta > 70°$, when the sheath fully collapses. This finding coincides with the discussion in 6.3.2. This finding confirms that the quasineutral formulation used in typical Hall thruster codes is not a good candidate for determining near-wall ion dynamics. Since due to numerical reasons it is impractical to resolve the Debye length and solve the Poisson's equation directly, additional steps need to be taken in hybrid Hall thruster codes to correct the ion wall flux [43].

The sheath is seen to start collapsing at angles greater than 70°, confirming the



(a)

Figure 6.6: Variation in the maximum sheath thickness with the angle of magnetic field. Dashed line corresponds to the solution obtained assuming quasineutrality.

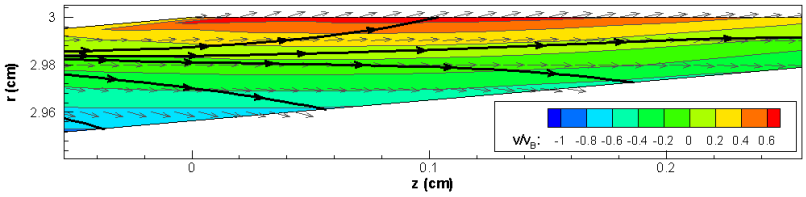


Figure 6.7: Plots of normalized radial velocity and ion velocity streamlines at $\theta = 85°$.

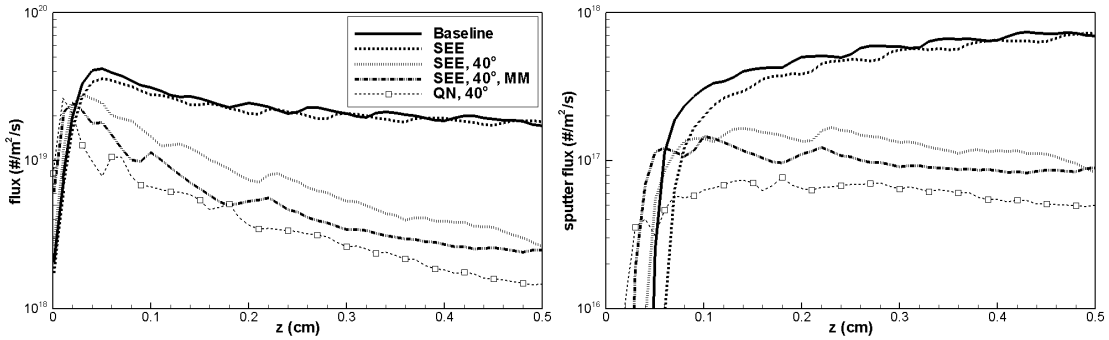
simple analysis presented in the introduction. To better illustrate the dynamics at this highly inclined magnetic field geometry, the ion velocity contours and velocity streamlines at the 85° incidence angle are plotted in Figure 6.7. The contour plot corresponds to the radial velocity component normalized by the Bohm speed. At this high incidence angle, the value of the normalized velocity remains below unity, indicating that the Bohm speed is never reached. In addition, ions are moving towards the wall only along a small region near the left boundary. This result is likely a direct byproduct of our loading scheme since it affects only the ions injected into the sheath. Ions originating in the bulk plasma are accelerated away from the wall. Ions located just a small distance from the wall are seen to follow trajectory first parallel to the wall, and subsequently turning away from it. Ions are thus seen to be repelled by the wall, indicating a sheath collapse.

6.4.2 Erosion and Lifetime

These numerical results confirm that the presence of highly inclined magnetic fields results in a decreased sheath thickness. This observation is next correlated to the wall flux. From mass conservation, $\Gamma_w = n_{i,s}u_B$, where the terms on the right hand side correspond to the ion density at the sheath edge and the Bohm velocity. The computed wall flux is shown in Figure 6.8a. We can see that although the presence of SEE tends to reduce the sheath thickness, it has only a negligible effect on ion wall flux. The flux remains approximately constant for these cases along the wall length, with the slight decrease due to the reduction in bulk ion density due to ion acceleration. The initial spike is an artifact of the loading scheme, as noted previously. The inclined magnetic field is seen to reduce the wall flux considerably, which can be attributed to the net acceleration of ions away from the wall and hence a reduced sheath ion density $n_{i,s}u_B$. This observation has a profound effect on both the ionization efficiency and the thruster lifetime, since ion losses to the walls are a major contributor to both of these inefficiencies. Here we consider only the impact on wall erosion. Material sputtering yield scales with both the impact angle and the energy of the incoming ions. Several models exist for computing sputter yields for Boron Nitride, the material typically used in SPT-type Hall thrusters. In this work we utilize the logarithmic fit suggested by Garnier [74],

$$Y_0(E) = 0.0156 \ln E - 0.0638 \tag{6.9}$$

This fit is valid from the energy threshold of 60 eV up to the keV range. In our analysis we neglect low energy sputtering. For angular dependence of yield, quadratic



(a) (b)

Figure 6.8: Comparison of wall flux and emitted sputtered yield for 5 selected configurations.

polynomial fit is recommended by Yim [31]

$$Y(E,\theta) = Y_0\left[-4.45\times10^{-7}\theta^4 + 4.91\times10^{-5}\theta^3 - 9.72\times10^{-4}\theta^2 + 3.44\times10^{-3} + 1\right] \tag{6.10}$$

where θ is in degrees and is measured from the wall normal.

Figure 6.8b shows the calculated sputter yield. The profiles for the baseline and SEE configurations are shown to exhibit an increase in sputter yield in the axial direction even though the flux decreases. This response is due to the angular dependence. Figure 6.3a indicates that the ion incidence angle increases with the distance along the wall. Similar response is seen in the remaining cases. Presence of a 40° magnetic field inclination results in flux reduction by approximately 60%, leading to a correspondingly similar reduction in erosion rate. In addition, we see that the magnetic mirror has only a marginal effect on wall flux and erosion rates in the inclined field configuration. This can be explained by realizing that both the magnetic field incidence angle and the magnetic mirror generate analogous electric field profiles.

The dashed line with markers corresponds to a 40° incidence angle and the quasineutral field solver. We can see that although the QN solver reproduces the general trend seen in the reference case, the computed fluxes and yields differ consid-

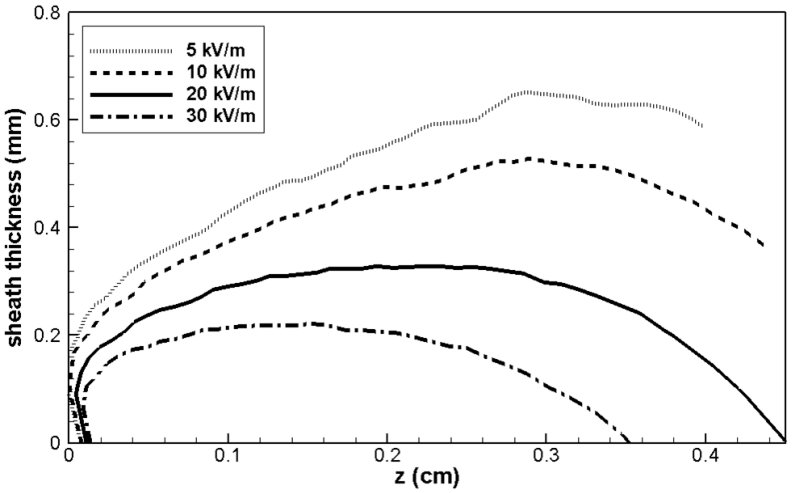


Figure 6.9: Sheath profile for $\theta = 60°$ as a function of normal electric field.

erably from the baseline. Average QN flux is 2×10^{18} #/m^2/s compared to 2×10^{19} #/m^2/s obtained using the Poisson solver. Comparably, the average sputter yield decreases from 5×10^{17} #/m^2/s to 5×10^{16} #/m^2/s with the QN solver. This difference corresponds to an order of magnitude reduction in wall and sputter flux due to numerical effects, again indicating the necessity for additional treatment of ion wall fluxes in quasineutral simulations.

6.4.3 Sheath Stability

The analysis presented in the previous paragraphs was performed using the prescribed normal component of electric field $E_\perp = 20$ kV/m. Although the actual profile of the potential drop in the Hall thruster adjusts in response to the internal plasma dynamics, the total potential drop is a design parameter arising from the applied potential drop between the anode and the cathode. To investigate the effect the field strength has on the sheath profile, we ran the code for several values of $E_\perp$ with $\theta = 60°$, no SEE, and no magnetic mirror. These results are plotted in Figure 6.9. The solid line at 20 kV/m corresponds the case studied previously. Reducing the applied potential drop leads to a thicker sheath, as expected. We can see that for this particular field angle, full sheath collapse will occur approximately at $E_\perp = 50$

kV/m. It should be pointed out that this model predicts that the inclined magnetic field leads to sheath formation only if axial electric field is small. According to this model it is predicted that the potential drop inside the Hall thruster channel decreases leading to shifting the potential drop outside. Such effect has significant implications on the plume formation and the thruster contamination aspects. It is interesting to point out that such trend is also observed experimentally [23, 64, 75].

Chapter 7

Plume Modeling

7.1 Introduction[1]

The final component of our multiscale approach addresses the expansion of the plasma plume. Outside the thruster, the magnetic field strength decays rapidly and the plume dynamics is dominated by the electrostatic Lorentz force, $\vec{F} = q\vec{E}$. However, an important factor in the plume is collisional interaction between particles. Of particular importance is the so-called charge exchange reaction (CEX). The resonant charge exchange occurs when an ion and neutral of the same parent species come into a close contact to exchange an electron without a significant momentum or energy transfer. Since the velocity of neutrals is much smaller ($\sim$ 1 km/s thermal velocity) than the velocity of the ions ($\sim$ 17 km/s), this interaction

$$Xe^{\circ}_{slow} + Xe^{+}_{fast} \rightarrow Xe^{+}_{slow} + Xe^{\circ}_{fast} \tag{7.1}$$

results in the creation of slow ions and fast neutrals.

The thruster plume always contains a radial $\Delta\phi$ component due to the radial decay in density. Furthermore, due to the geometric expansion of the plume, the $\vec{E} = -\nabla\phi$ field points slightly back towards the thruster. The strength of this

[1]This chapter was reviewed by AFRL public release office and was approved for public release as Distribution A, distribution unlimited.

field can be estimated by considering typical potential drops and beam radii, E_r = 20V/20cm = 100V/m. Considering the primary beam ions, we can see that this beam merely acts to increase the beam divergence. Assuming the ions were initially moving in axial-only direction and that a uniform field acts over a distance of 0.5 m, the induced radial velocity component is $v_r = \sqrt{2 * 50eV/m_{Xe+}} = 8572$ m/s, giving $\theta = \tan^{-1}(v_r/v_z) = 27°$. A much stronger effect is seen on the CEX ions. Since these ions can be approximated as initially stationary, they will be accelerated by potential gradients in the radial direction. Assuming the initial energy is $\sim 0eV$, the CEX ions exit the plume with $W = e\Delta\phi$ kinetic energy. This effect has been demonstrated numerous times by experimental measurements with sweeps of energy analyzer probes showing a clear population of low energy ions at high angles from the beam centerline [49, 76].

Outside the thruster plume, the charge exchange ions expand into a donut-like structure commonly known as charge exchange wings. These wings have been studied extensively by previous researchers, including Roy [66], Wang [77], and Boyd [49]. These expansion of these ions is based on local electrostatic fields and can result in an ion backflow into regions with no line of sight to the thruster. This introduces several issues. First, the collection of plasma current by spacecraft components could lead to a dielectric charging and potentially disastrous arcing. Fortunately, this effect is generally not pronounced, in fact, since ions will be attracted to negatively charged surfaces, electric propulsion plumes can in fact reduce the spacecraft potential. However, directly related to this effect is an impact on instruments. Many space weather satellites utilize charged particle sensors to study space plasmas and phenomena such such solar storms and magnetic reconnection. Operation of EP devices used for station keeping can modify the local plasma environment around the sensor and lead to a spurious current collection. Furthermore, CEX plume expansion can also result in a direct contamination. The Xenon propellant used in typical EP thrusters is not re-

acting, however, impacts of CEX ions could potentially lead to low-energy sputtering of condensible material. Similarly, condensible materials sputtered from the thruster can be ionized in the plume and subsequently expanded radially in a manner similar to the standard CEX expansion. These molecular contaminants can subsequently polymerize to the surface and form monolayers that reduce transmission of optical sensors and modify emissive properties of thermal control surfaces [78].

7.2 Draco Plume Simulation Code

Unfortunately, it is difficult if not outright impossible to predict the dynamics of the backflowing ion plume without performing numerical simulations. Such simulations allow the designer to perform trade studies, and find the optimal location for sensitive instruments. Quite a large number of numerical codes have been developed precisely to tackle modeling of EP thruster plumes [77, 79–82].In this work we utilize an electrostatic particle-in-cell code Draco. Draco was developed in 2005 at Virginia Tech and was subsequently integrated into the AFRL Coliseum framework [83]. The author of this dissertation was the primary developer of Draco. An early version of the code is described in the author's master's thesis [84] and also in [48]. The code has gone through a large number of changes since then, with several improvements of interest implemented in the course of this dissertation. These new features primarily tackle numerical issues necessary to make the code more efficient and robust. These features are described below.

7.2.1 Mesh Splitting

Draco was initially developed as a serial code operating on a strictly Cartesian mesh $x = x_0 + i\Delta x$. However, the need to model larger simulation domains dictated the necessity to incorporate support for parallel processing [85] and non-Cartesian rectilinear stretched meshes [68]. Mesh stretching allows the user to capture a larger

physical domain without increasing the number of mesh nodes (unknowns) that need to be solved for and also stored in memory. The latter item also resulted in an addition of *zones* to describe the topology in each axial direction. This was necessary since it would be difficult to describe general simulation set ups using a single stretched mesh. Often we need to use a fine mesh in a high density region around the thruster, and an expanding mesh in the low density far plume.

Unfortunately, while the implementation of the multi-zone approach decreased the number of computational nodes needed to capture a certain region of interest, it also resulted in an additional overhead on the particle push. The primary strength of Draco had always been its speed. Particle codes such as Draco utilize a large number of computational particles to describe the velocity distribution function of the simulated fluid. At every time step, two sets of interpolations are necessary per particles. First the particle positions are scattered to the grid to compute the electric field. The electric field is then interpolated back onto the particles to update their velocity. The strength of the Cartesian mesh approach is that is that the interpolation becomes trivial - the cell in which the particle is located is easily inverted from position, $i = (x - x_0)/\Delta x$. The cell logical coordinates are also easily obtained from $l_i = i - \text{int}(i)$. In the initial implementation of the multi-zone approach, the zones existed as virtual boundaries overlaid over a single computational mesh. The particles moved in the parent mesh, and in order to compute the logical coordinates, the zone containing the particle had to be first determined. This was done by looping through the zones and comparing particle positions to the zone boundaries. This was a highly inefficient operation as indicated by profiling studies, such as the one shown in Figure 7.1. This plot shows the effect of increasing the number of particles from 200k to 500k (2.5 times) and also of increasing the mesh size 8 fold (2x2x2). Blue bars are for the small mesh size while the red ones are for the larger one. Similarly solid bars indicate the runs with 200k particles, while the dashed one are for runs with 500k

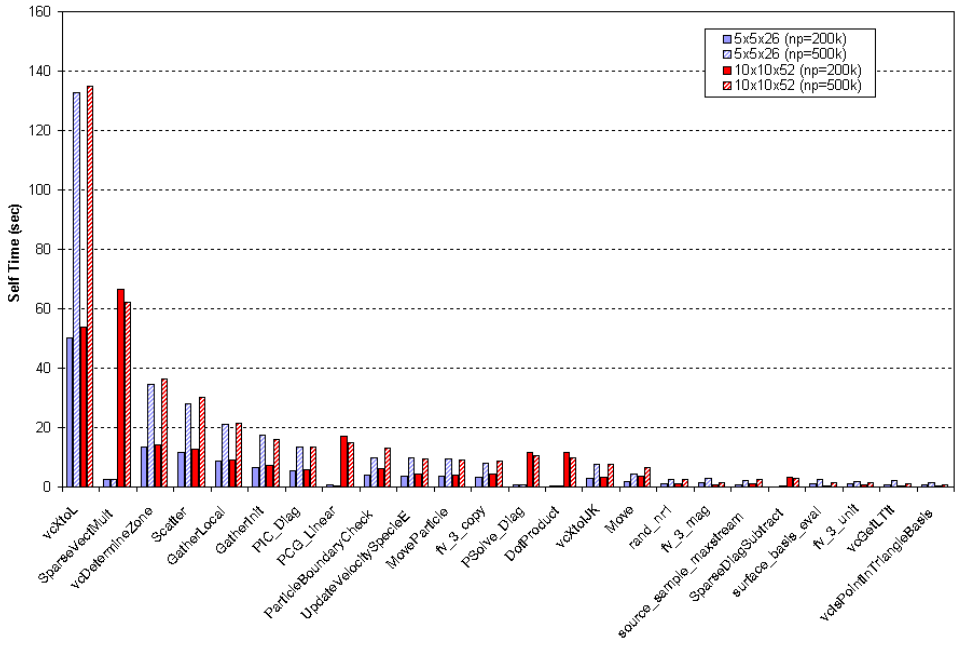


Figure 7.1: Profiling studies from previous version of Draco

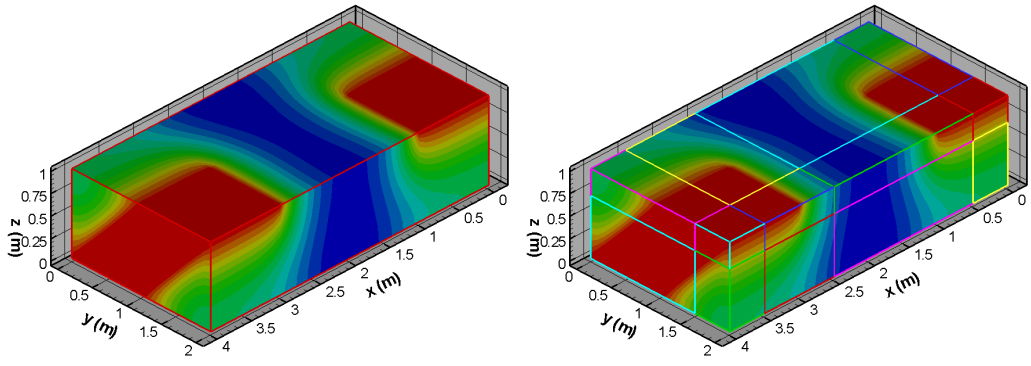
particles. In this study, the PCG solver was used to obtain the plasma potential.

Reading from left to right, we can seen that functions utilizing the largest amount of computational time are `vcXtoL`, `SparseVectMult`, `vcDetermineZone`, `Scatter`, `GatherLocal`, and `GatherInit`. Of these, only one, `SparseVectMult` is performing an actual computation. All other functions are responsible for interpolating particle positions to and from the mesh! The function taking up the disproportionately largest chunk of computational resources is `vcXtoL`, which is the function that translates the physical coordinates into logical ones. This is the step that is supposed to be trivial for a Cartesian mesh, yet here we see that it is in fact dominating the code performance. In comparison, the actual particle move implemented in `Move` is trivial. As expected, the time is directly proportional to the number of simulation particles (the matrix-vector multiplication in `SparseVectMult` on the other hand demonstrates dependence on mesh size, as expected). The `vcDetermineZone` function contains the code searching for the particle zone which was added once multiple zones were implemented. Increasing the mesh size is mainly demonstrated as a performance hit

in the PCG potential solver. Since most Draco simulations do not use a Poisson solver, this is a non-issue.

Hence, part of the development effort went into optimizing the code and improving the code performance. This was achieved by rewriting the internal mesh representation. Instead of treating the zones as virtual boundaries superimposed over a single contiguous mesh, the mesh was divided into separate chunks. Each chunk took the ownership of particles located in it. Since each zone contains only a single mesh definition, this rewrite completely eliminated the need to search for the containing zone. It also resulted in an improved memory utilization. It is hard to guarantee a large contiguous memory block on modern multi-process operating systems. Consider a computer system with 2 Gb of RAM. Now assume that the bottom 300Mb is used by the operating system. Next, assume the user loads a data analysis program which allocates the next 600 Mb of RAM. Next, the user opens a text editor that uses 10 Mb of data. Finally, the user closes the data analysis program, freeing the 600Mb of RAM, and starts a simulation. Assume the simulation uses a mesh with $250 \times 250 \times 250$ nodes and that the size of data stored on each node is 20 doubles (20*8 bytes). The total memory required to allocate this mesh is 1.19 Gb. Despite the operating system reporting 1.7Gb of free memory, this allocation request will fail. The reason is simple - the system lack a single contiguous memory block of this size. The memory structure in our simple example consists of 300Mb used by O/S, a 600 Mb empty space, 10 Mb block used by the text editor, and finally another empty 1138 MB (1.11 Gb) block. The behavior described in this example is actually fairly typical and previous experience showed that Draco simulations were typically limited to about $200 \times 200 \times 200$ nodes, despite the amount of available RAM. By dividing the mesh into a number of smaller blocks, it is more likely the operating system will be able to accommodate the allocation requests.

The downside of the zone split is that the some performance gain is lost to trans-



(a) (b)

Figure 7.2: Potential solution obtained for a single and multiple (16) zones

ferring particles across zone boundaries. However, since such a packing / unpacking operation is also required in the distributed parallel mode, the distributed zone handling approach in fact simplified development and debugging of the parallel code. Besides changing the way the particles move, the split into individual zones also affected the field solver and mesh properties. A boundary buffer was implemented to obtain data from neighboring zones. In the case of a processor boundary, the neighbor data is obtained via MPI send/receive commands. The neighbor data is used to add up properties such as particle densities along zone boundaries, and to compute potential and electric fields. Figure 7.2 shows potential around an arbitrary test geometry computed on a mesh segmented into 16 zones, and on a corresponding mesh consisting of only a single zone. The multi-zone case was in addition run in parallel with 4 zones per processor. As can be seen from this plot, the two solutions are identical.

The division of mesh into multiple zones had another benefit for parallel computations. Previously, due to difference in which zones and processor boundaries were handled, multi-zone meshes were not supported in parallel. In a parallel run, each zone was automatically assigned to a new processor. This imposed a severe limitation on the possible domain decomposition. Consider for instance a computer cluster

containing 32 CPUs (CPU counts often come in powers of 2). In order to utilize all 32 CPUs, a $4 \times 4 \times 2$ domain decomposition is required. Such a zone definition is not always ideal. The new implementation completely divorces the number of zones and the number of available CPUs. Zone assignment is now performed in a round-robin case. In the case of a serial run, all zones are given to the first (and only) processor. If multiple computers are present, they are assigned zones in a sequential order: zone 0 goes to CPU 0, zone 1 to CPU 1, zone 2 to CPU 2, zone 3 to CPU 0, zone 4 to CPU 1, etc... This example assumes there are 3 CPUs in the simulation space. Such an assignment results in optimal work balancing for majority of cases. The only limitation is that for obvious reasons the domain contains at least as many zones as CPUs.

Finally, it should be noted that parallel communication across an arbitrary number of processors and zones is a non-trivial task. Care is needed to avoid deadlock, a situation in which processors keep waiting on each other. This is avoided in the new Draco by a assigning each face a unique faceID. Communication is performed by looping over the faceIDs consecutively. Each processor checks if it contains that particular faceID, and if it does, communication is performed. Additional care is also needed in the summation of shared data. It is imperative that the summation is performed one dimension at a time, otherwise, inconsistent values will be obtained along nodes shared by more than two zone.

7.2.2 Mesh Intersection

Second important change is related Draco's helper mesh-generation module *Volcar*. While rectilinear meshes speed up particle push and interpolation, they make it more difficult to capture realistic geometries. In finite-element analysis, the mesh shape contours to the surface, but in a rectilinear mesh a cut cell approach must be taken to capture the surface geometry. Draco uses a limited cut-cell approach in

which the surface mesh is used to generate boundaries for particles, but a staircase representation is used by the potential solver. Volcar is responsible for classifying nodes as "internal" (fixed potential, assuming conductive objects) or "external" (free space), and also for collecting surface elements in interface volumes to be used for particle-surface impact checks. Volcar first generates the rectilinear zones according to the user domain definition. It then intersects the volume mesh with a surface mesh describing the geometry. Node location is based on the orientation of the surface normal vectors, with the vectors pointing into the free space.

The original Virginia Tech version of Draco utilized a hybrid structured/tetrahedral mesh. Each simulation cell (a brick) was subdivided into 5 tetrahedra (see, for instance, Figure 2 in [48]). The mesh intersection was performed not with the parent block but with each tet. The motivation for this implementation was that the VT Draco was coupled with an immersed finite element (IFE) solver that utilized such a mesh to solve for potential in presence of interface cuts [86]. Unfortunately, the IFE solver was never fully coupled into Draco, with one of the driving reasons being software language incompatibility. While Draco was developed in C, the IFE solver was written in Fortran 90. Not only did this required two sets of compilers, the difference in memory utilization also dictated the mesh structure to be duplicated in Draco and in IFE, doubling memory requirements. Finally, it was found that the IFE solver was not versatile enough for real-life situation. A basic assumption in the implementation of IFE was that the mesh is fine enough such that each tetrahedron is cut at most one time. Although it is fairly simple to satisfy this requirement for academic test cases consisting of bricks and cylinders, it was found that in practice it was impossible to meet this requirement for CAD-generated geometries. Real world problems typically contain thin features and sharp edges that will result in multiple surfaces cutting through a tetrahedron.

For these reasons, the IFE solver was officially removed from Draco distribution

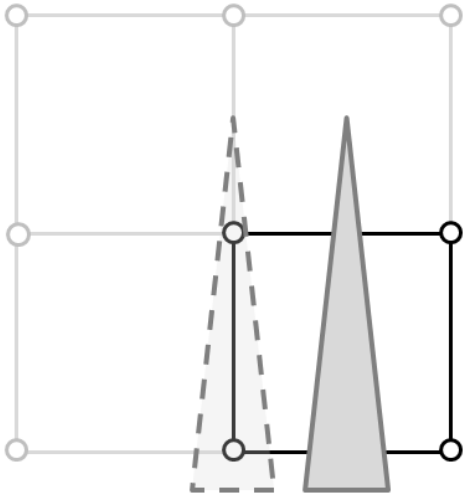
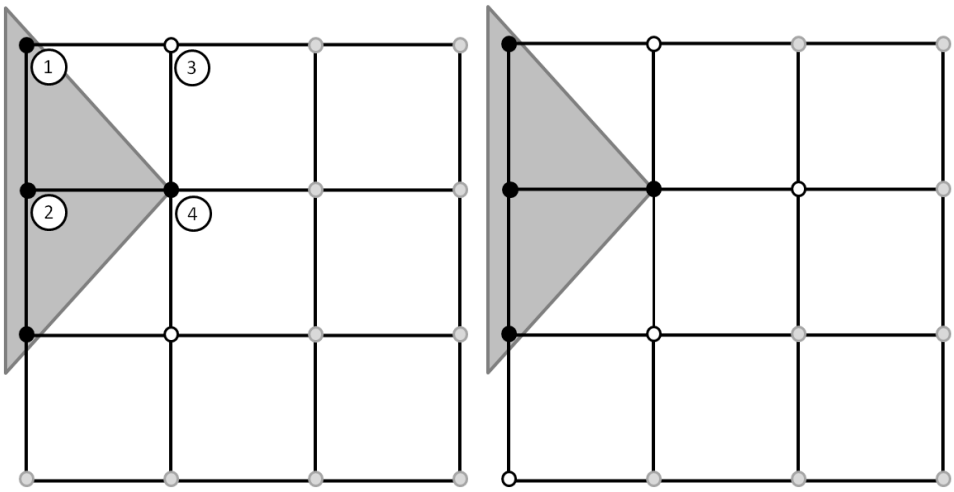


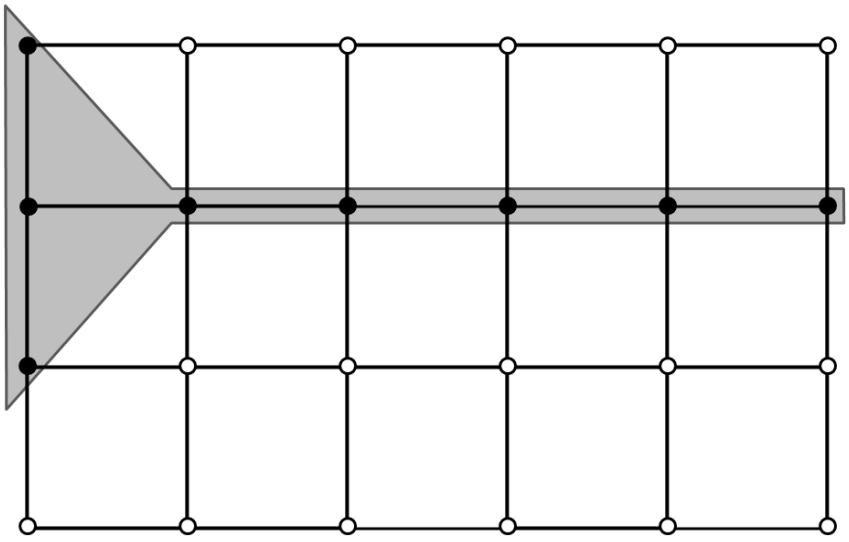
Figure 7.3: A typical thin feature resulting in multiple cell/edge cuts. In order to satisfy the single cut requirement, the wedge would need to be placed as shown by the dashed form.

around 2006. However, much of the mesh generation methodology was left in the code. Most important of these was the fact that Volcar was designed to operate on a cell-by-cell basis. The code looped through the cells and classified each cell as internal, external, or interface. Interface cells were those that had a cut passing through them. Internal and external cells were cells that were completely inside or outside a solid object. This methodology worked well if the single cut requirement was met, however again failed to work for real world cases. To illustrate this, consider a cell containing a thin feature as demonstrated in Figure 7.3. It should be pointed that 2D representation is used in this and subsequent plots for clarity, however, Volcar operates in 3D. This cell is no longer interface in the original sense of the definition, since it does not form an interface between an internal and external region (both left and right side are in free space). However, marking this cell as external would disregard the fact the cell contains surfaces that need to be check for particle interactions.

In addition, Volcar uses interface cuts to classify nodes as internal or external. Situation like the one depicted in Figure 7.3 could in some instances result in an incorrect node classification if the incorrect surface triangle is used. Node classifica-



(a) Incorrect classification (b) Correct classification



(c) Floodfill resulting from the incorrect classification

Figure 7.4: Typical problem in the earlier version of Volcar. Due to the cell-based approach, external nodes (shown by open circles) do not fully enclose the internal (solid) nodes. The subsequent floodfill used to classify remaining unknown (gray) nodes then results in propagation of "spikes". Since internal nodes are used to fix object potential, this error distorted the object shape.

tion is performed using the first visible surface with visibility determined by checking for no other triangle intersecting the ray from the node to the triangle centroid. In some peculiar instances, due to the cell-based approach, not all surfaces needed to be checked were captured. Another issue arose when surfaces terminated at a cell boundary. Consider the example shown in Figure 7.4a. The surface can be seen to "belong" to cells 1 and 2, however, a single vertex is shared with the neighboring cells 3 and 4. Often, due to the cell-by-cell approach and numerical imprecisions, the code would fail to capture the single point located in cells 3 and 4. These cells would then be marked as completely free of surfaces. Classification of node locations in the interface cells 1 and 2 would result in the situation depicted in Figure 7.4a. In this figure solid black markers indicate nodes that have been classified as internal, solid white markers are external nodes, and gray nodes are "unknown" nodes that have not been classified. The unknown nodes are classified by performing a flood fill from the set near-surface nodes. For this step to work correctly, a clear interface must exist around all surfaces. A well defined interface has internal and external nodes set on both sides of the surface, as depicted in Figure 7.4b. The difference is the single externally marked node one cell away from the shared vertex. In order for this node to be set, the cells 3 and 4 had to be marked as interface and contain the correct surface definition. In the absence of this single node being classified, the subsequent flood fill, assuming filling in the +X direction, will result in the situation depicted in 7.4c. The internal nodes will propagate to the mesh boundary or until another previously set node is encountered. Such anomalous "sticks" were a common plague of many older Draco simulations.

Resolving this issue required changing the methodology from a cell-centered to a node-centered approach. Instead of treating a mesh cell as the correct volume to intersect by the surface, a node-centered control volume (NCV) should be used instead. This approach is plotted in Figure 7.5. In the new implementation, the control

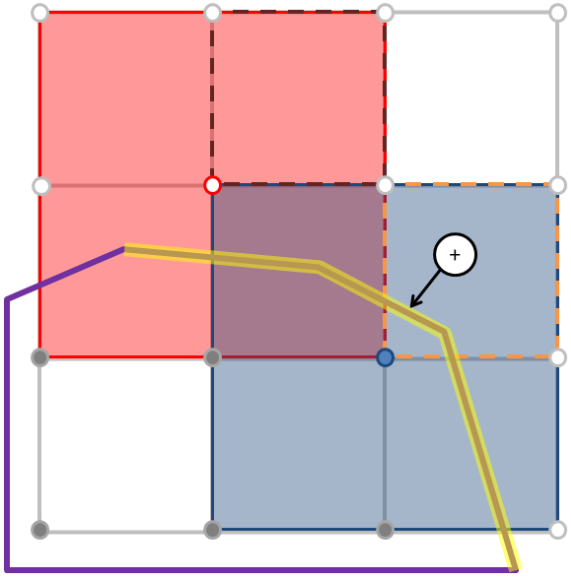


Figure 7.5: The two colored squares show the node control volume (NCV) surrounding each node as well as the overlap between neighboring NCVs. Node classification is performed using surfaces located in the NCV. The dashed boundary shows the cell in the +X,+Y and +Z direction. Particle located in this cell are checked for surface intersections using surfaces located in the NCV (highlighted in yellow).

volume extends by one cell in each dimension. This definition guarantees an overlap between control volumes and reduces the risk of some surfaces not being captured during particle surface intersection check. Although additional testing remains, this new implementation seems to have fully mitigated the issues present in previous Draco simulations.

7.2.3 Time-Dependent Source Model

Hall thrusters are known to be non-steady devices demonstrating a wide range of fundamentals frequencies [24]. The oscillations most easily observed in numerical and experimental studies is a prominent low frequency "breathing mode". This oscillation arises from a prey-predator type depletion and repletion of neutrals and ions. In [24] the frequency range for this type of oscillations is given as 15-22 kHz. Figure 7.6 shows the temporal variation in discharge current from a typical HPHall study of the Princeton Cylindrical Hall thruster. The frequency of the oscillations, as computed

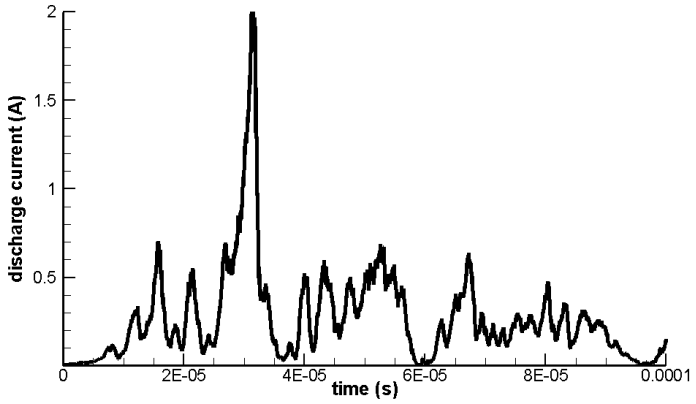


Figure 7.6: Discharge current from a typical HPHall simulation of the CHT showing a strong temporal dependence

by inverting time differences between several prominent peaks, is seen to be in the 17 to 30 kHz range, indicating that this indeed is the breathing mode.

Draco has been coupled previously coupled with HPHall in order to capture the details of the ion beam population [52]. This coupling is accomplished by sampling ions crossing a user defined radial grid line and storing their positions and velocity components into a file. This file thus gives us a radially-varying discretized velocity distribution function. In Draco simulations, particles are injected into the domain by associating *source models* with surface elements. These surfaces then emit particles according to the source-specific distribution functions and according to user defined parameters such as mass flow rate and mean velocity. Standard analytical models such as the Maxwellian and Lambertian (cosine) distribution do not offer a good representation of Hall thruster ions. The reasons are twofold. First, flux is not radially uniform due to a difference in density and velocity as a function of thruster radius. Secondly, ion beam velocity distribution function is also radially varying and non-Gaussian due to the self-induced accelerating forces. This starkly contrasts Hall thrusters from ion thrusters in which a cold ion population is accelerated through grids with a fixed applied potential gradient.

As described in [52], the HPHall source was indeed able to better reproduce the

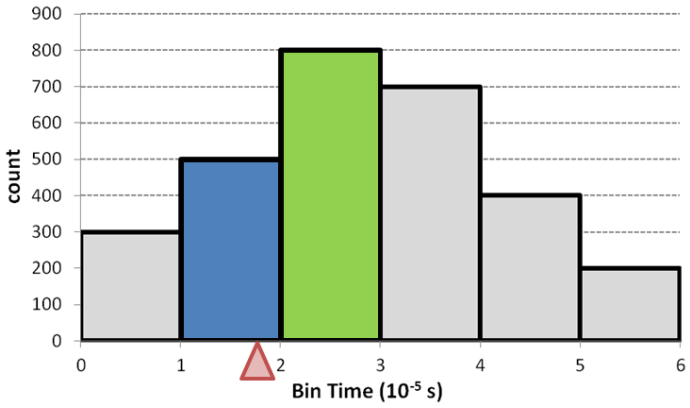


Figure 7.7: At a time indicated by the triangle, a particle will be sampled randomly from one the two colored bins, with a stronger preference given to the green bin.

ion beam population. However, in this previous implementation, no note was paid to the temporal behavior. Instead, the thruster was assumed to operate in a steady state, and ions sampled over several breathing mode oscillations were conglomerated. It is not clear how much importance does the temporal behavior play on plume dynamics. Due to the relaxing nature of collisions, it is possible that the variations in the source will actually be dissipated in the plume. To study this behavior in more detail, the HPHall source was modified to optionally read a time stamp at which the particle data was sampled. The sampling function in HPHall was also modified to export the time, given by `it*DT`, and also the azimuthal velocity, computed in the code as $v_\theta = \hat{h}/r$, where $\hat{h}$ is a mass-normalized angular momentum. To reduce the size of the collected list, a user-specified probability is used to select particles. Since in HPHall ions have a variable specific weight, the probability that a particle will be sampled is obtained by comparing the particle weight to a user specified output weight, $n_{sample} = w_p/(m_{Xe+} * sw_{sample}) + R$. Here w_p is the particle weight, m_{Xe+} is the mass of an actual Xenon ion, sw_{sample} is the desired specific weight, and R is a random number used to reduce round off errors.

If the time information is present, the source bins particles into a user specified number of Δt_b time slots (default is 100). This binning is illustrated in Figure 7.7. At

each call to sample a particle, the source determines the bin containing the current simulation time, $j = \text{int}((t - t_0)/\Delta t_b)$. To assure a smooth transition between bins, particle will be sampled from bin j or $j + 1$ based on its proximity to the boundary. More specifically, the cell coordinate $f = (t - t_0)/\Delta t_b - j\Delta t_b$ is compared to a random number R. If $f \leq R$, a particle is sampled from bin j, otherwise it sampled from bin $j + 1$. The sampled particle position Δz and Δr form an offset from the source centroid. The position and the velocity components v_r, v_θ, and v_z are also rotated through a random angle about the thruster centerline. The math used in the rotation is described in more detail in [52] and [68].

The time data is optional, and if it is not present, the source will sample particles from the entire given list. A simple, manually-generated example of input is given below:

```
VARIABLES = z r vz vr vt t
ZONE T=HPHALL_XE+
0 0.50 10000 0 0 0
0 0.50 10000 0 0 0
0 0.50 10000 0 0 0
0 0.50 10000 0 0 0
0 0.50 10000 0 0 0
0 0.25 8000  0 0 2e-4
0 0.25 8000  0 0 2e-4
0 0.25 8000  0 0 2e-4
0 0.00 2000  0 0 4e-4
0 0.25 8000  0 0 6e-4
0 0.25 8000  0 0 6e-4
```

This input shows two important features of the time-variable source. Both position/velocity and also mass flow rate can be a function of time. The variation in

mass flow is captured automatically by the number of entries for different time slots. The mass flow rate at a specific time is obtained from $\dot{m}_j = \dot{m}_0(c_j/\bar{c})$ where c_j is the number of particles in bin j and $\hat{c}$ is the average across all bins. $\dot{m}_0$ is the user specified mass flow rate that will be generated over the entire sampling period. The result is that, after sampling over the entire given time range, the source will produce the defined mass flow rate.

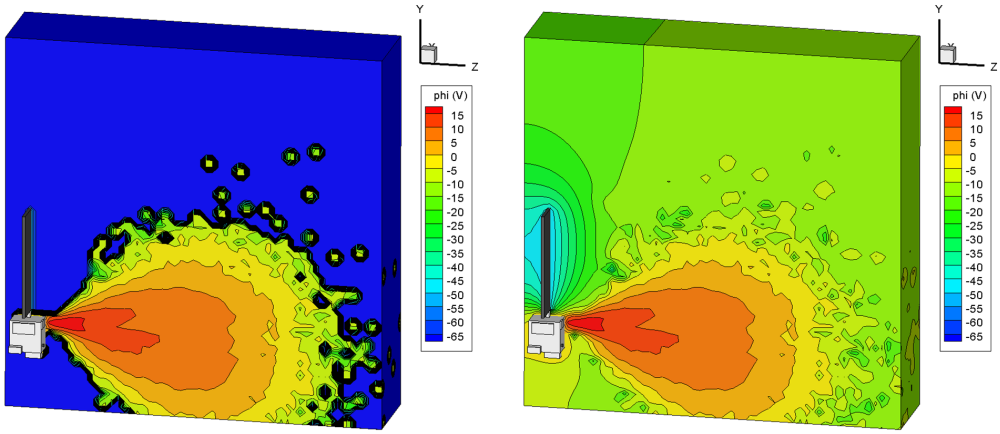
Besides specifying the number of segments and the average mass flow rate, the source also take two additional parameters specifying the initial time t_0 and a cutoff time t_f. The first parameter t_0 can be used to shift the place in the input file where the sampling will begin. This parameter is useful if for instance we are interested at beginning the particle injection at the peak of discharge current, in which case t_0 would be set to the time at which the first peak occurs. The second parameter is used to set the time in the input file at which the sampling will loop to the beginning. Typically, the input data to the source will contain HPHall exit plane data over a few thruster oscillations. However, the plume simulation will be run over a greater time scale, making it necessary to loop through the input data multiple times. The source will perform this looping automatically once the simulation time exceeds the sampling range. However, to avoid anomaly in the plume results, it is necessary to match the end and start data points. The matching can be performed using several ways, one of which is by matching the phases in the discharge current. As such, the recommended procedure is to visualize the discharge current log data from HPHall over the time range exit plane sampling was performed. As an example, these parameters could be used to bracket the data in Figure 7.6 between 4×10^{-5} and 9×10^{-5} seconds. After the Draco simulation completed 5×10^5 seconds, the sampling would continue from the 4×10^{-5} marker. It should be pointed out that in order to capture all sampled particles, the bin size must be set such that $\Delta t_b \geq \Delta t_{sim}$. If these optional parameters are not specified, no time offset is used and the entire time sequence is

used for sampling.

7.2.4 Quasineutral Potential Fix

Finally, a "quasi-neutral" switch has been implemented in the Draco Gauss-Seidel Solver. Implementation in the PCG solver is pending. The switch allows the use of the Poisson solver on meshes too coarse to solve otherwise. A Poisson solver will not converge in the cell spacing is significantly larger than the local Debye length. In the plume core where $n \sim 10^{17}$ m^{-3} and $kT_e \sim 5$ eV, the Debye length $\lambda_D = \sqrt{\epsilon_0 kT_e/ne^2} \sim 5 \times 10^{-5}$ m. Even with the use of a stretched mesh, it becomes computationally infeasible to resolve the Debye length while at the same time resolving the plume far-field. As such, the plume potential is typically obtained by assuming quasi-neutrality $n_e = n_i = n$ and inverting the Boltzmann relationship to obtain $\phi = \phi_0 + kT_e \ln(n/n_0)$. Although this approach is valid in the plume, it has the serious handicap in that it does not resolve the non-neutral sheath. As such potential will be correct in the plume, but electric fields surrounding charged objects in the low-density region will not be included. Resolving these fields is on the other hand imperative, since they drive the trajectories of the backflowing ions.

The QN switch was implemented to resolve this limitation. It follows the implementation in Aquila's field solver [81]. At the start of each Poisson solver call, the local Debye length is calculated at each node using the local value of plasma density and electron temperature. If the node volume is greater than the volume of the Debye sphere, the node is flagged as Dirichlet and potential is fixed on it from the Boltzmann inversion. Poisson equation is solved on the remaining nodes. Hence, the potential in high-density plume is computed directly from the inversion, and the Poisson solver backfills the low-density sheath-dominated region. The Poisson backfill is performed using a different set of reference parameters for the Boltzmann electron relationship, $n_e = n_0 \exp((\phi - \phi_0)/kT_e)$ to capture the properties of the ambient plasma. This fea-



(a) Boltzmann Inversion (b) QN switch

Figure 7.8: Comparison in potential between standard Boltzmann inversion and QN-switched Poisson solver. The sheath structure around the solar wing is clearly absent from the inverted solution.

ture is demonstrated in Figure 7.8. The first plot shows a potential profile computed using the standard Boltzmann inversion. The second plot shows the solution using the QN switch approach. Both of these plots were generated by loading a particle distribution from a restart file and outputting the initial potential.

To better illustrate the difference, the plume simulation was performed without including collisions. The potential in the plume is seen to be identical, as expected. However, the sheath surrounding the negatively charged solar wing is clearly absent in the Boltzmann inversion solution. As such, the first solver will underpredict the current collected by the wing. The sheath acts to increase the effective volume from which ions are extracted and accelerated towards the surface. In the absence of the sheath, ions are unaware of the existing potential drop until they arrive in a cell adjacent to the surface.

7.3 Results

We demonstrate the plume modeling step using a hypothetical space weather satellite. Small Hall thrusters such as teh Princeton CTH may be an attractive replacement for arcjets commonly used for station keeping. The surface model was shown previously in Figure 7.8. This generic satellite consists of the bus, a solar panel, a cluster of two thrusters, and several instrument boxes. The bottom left box on the exposed face in Figure 7.8 is assumed to contain a space weather instrument designed to measure the flux of low energy ions. We are interested in determining the extent to which the Hall thruster will modify the local plasma environment, and the amount of additional current collected due to the thruster plume backflow (here we are assuming that this instrument does not contain any additional mass-spectrometers that could differentiate the heavy thruster ions from the light solar protons). In addition, we are interested to determine whether the temporal characteristic of the thruster discharge will be captured by the instrument.

We assume the spacecraft is operating in the GEO environment where the baseline ambient plasma environment is almost negligible, $n_i = 5.8 \times 10^5 \mathrm{m}^{-3}$ [87]. For simplicity we do not consider the large surface potentials that often arise in GEO satellites due to charging, and set all spacecraft surfaces, including the solar wing, to 0V. The only exception is the space weather instrument, which is assumed to operate at 50V below the ambient plasma to repel low-energy electrons. Ion particles are injected using the data sampled from the thruster simulation corresponding to the discharge current in Figure 7.6. The QN-switched potential solver is used to obtain the plasma potential. The reference parameters used were $(\phi_0, n_0, kTe_0) = 20\mathrm{V}$, 10^{17} m^{-3}, and 3 eV in the beam plasma, and 0V, 5.8^5 m^{-3}, and 1 eV in the ambient backfill region. Collisions are modeled with the Monte Carlo Collisions (MCC) approach. Only CEX interaction were considered in this work. In addition, neutrals were not modeled directly, instead the neutral plume was represented by a fluid projection.

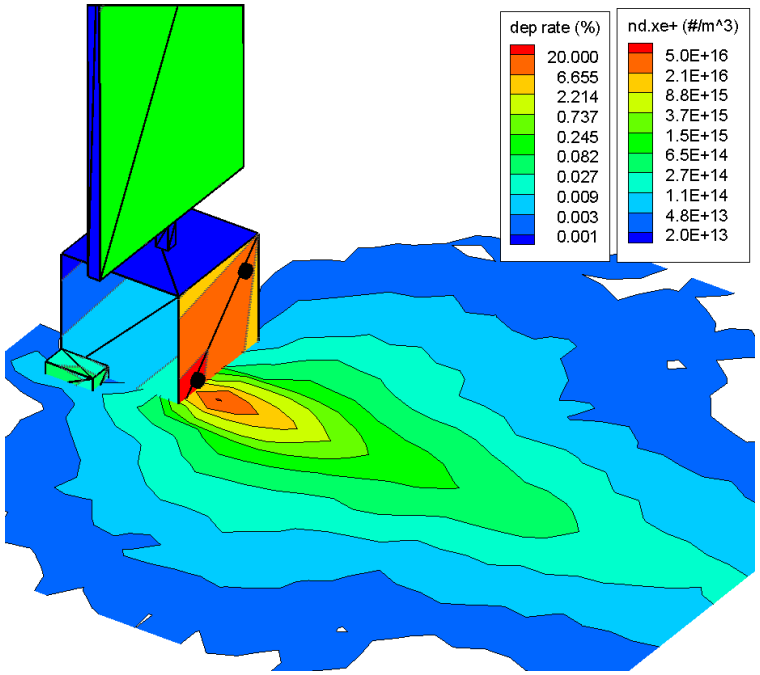


Figure 7.9: Simulation results showing the surface deposition rate, normalized by thruster mass flow rate, and the plume density in m^{-3}.

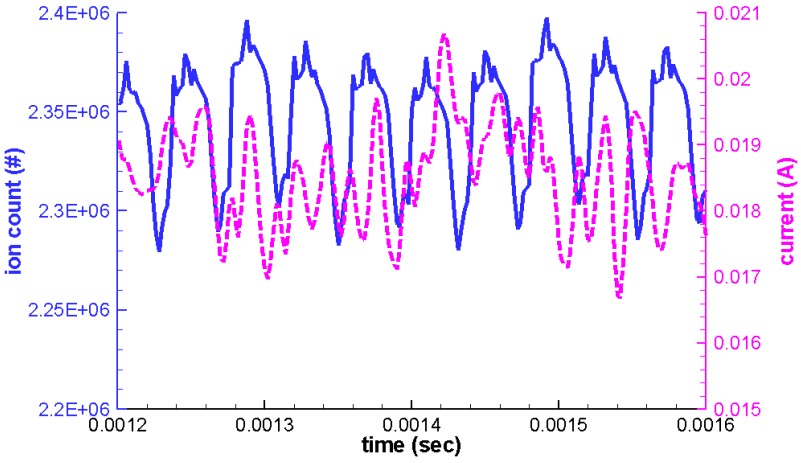


Figure 7.10: Comparison between the simulation particle count at steady state and the instrument collected current.

The simulation was let run for 4000 time steps. Total of 2 million particles were used in the simulation. Such a large number was needed in order to obtain a statistically significant current in the low-density wake region. Results averaged starting at the steady-state are shown in Figure 7.9. This figure contains two sets of contours. The contour levels on the slice through the plume show the ion number densities. The radial expansion of the plume is clearly evident and so is the increased density near the negatively charged instrument. The ion density in the vicinity of the instrument is 5×10^{13} m^{-3}, a four-order reduction from the plume densities, but an O(8) increase from the ambient environment.

The contour levels on the surface correspond to the deposition rate, normalized by the source mass flow rate. As can be seen from this figure, the surface is described by a very coarse mesh (the spacecraft geometry was generated analytically by combining basic building blocks since a CAD package was not available). While this coarse mesh does not provide us with details of the surface collection, it at least allows us to quantify the collection. We can see that majority of backflow occurs in the region behind the thruster, which can be expected. The solar wing also collects a fraction of the backflowing plume. By summing up the exposed faces of the instrument of interest, we can seen that it receives 0.3% of the emitted flux. It should be noted that in a real system, the collection rate would be significantly lower since the instrument aperture represents only a small fraction of the total instruments surface area.

Figure 7.10 shows the temporal response. The blue curve plots the total number of particles in the simulation as a function of time. The count is seen to oscillate due to the time-dependent source. The purple curve shows the current collected by the instrument. The temporal behavior is clearly also transferred to the backflowing plume. However, it appears that the frequency present in the backflowing plume has doubled from the source frequency, since on average two prominent peaks can be seen for each peak in the particle count. The physical reason behind this, as well as the

additional detail of the temporal response of the CEX ions remains as future work.

Chapter 8

Conclusion and Future Work

8.1 Conclusions

This thesis described a novel method for analyzing Hall thrusters, in which different spatial scales are considered independently by codes designed to tackle them in an efficient manner. Such a coupling allows for properties of interest to be determined self-consistently. The spatial scales considered include the microscopic scale of electrons orbiting about the field line, the spatial scale of the thruster, the near wall region, and the plume environment.

The electron orbits were studied by a kinetic code Lynx. This code simulates the cyclotron motion of electrons about a magnetic field line and takes into account electron diffusion events such as collisions and wall interactions. It does not include any anomalous scattering events. The code was first used to analyze transport along a single magnetic field line in a planar configuration. These studies indicate that in the presence of strong secondary electron emission, plasma temperature becomes non-isothermal, with $kT_{e,\parallel} < kT_{e,\perp}$. In addition, the code indicates that electron diffusion is driven by synergistic effects, and that the general relationship used in Hall thruster codes, $\mu = \mu_{collisions} + \mu_{walls}$ does not hold in the near-wall region. In a classical sheath configuration, the transport due to wall effects is minimal, since majority of electrons are screened from the wall by the potential drop in the sheath. Collisions

act to modify the velocity space, allowing more electrons to penetrate the wall. This effect is particularly important when $kT_{e,\parallel} < kT_{e,\perp}$, which is the case in Hall thrusters. The result is that in the near wall region, $\mu_{collisions+wall} > \mu_{collisions} + \mu_{walls}$.

The actual application of the kinetic code was demonstrated by modeling transport in the Princeton Cylindrical Hall thruster. The HPHall code developed at MIT in 1996 by Fife was used to first compute global discharge parameters using analytical mobility. However, it was found that HPHall is not appropriate for modeling thrusters with converging magnetic field line topologies due to the numerical mesh used by the code. As such, the simulation results produced by HPHall did not offer a good agreement with experimental measurements. They were still utilized in the kinetic analysis to demonstrate the feasibility of the code to work with such inputs. The Lynx simulation produced mobility of the same order of magnitude as the values used in HPHall. Of interest is that, despite each magnetic field line being analyzed independently, the individual solutions demonstrated a spatial coherence. These mobilities were then loaded into HPHall resulting in a slight improvement in the agreement with the experiments. HPHall was also used to sample particle leaving the thruster to form a disretized velocity distribution function for a plume analysis.

Next, a 2D kinetic ion and fluid electron hybrid code was described for analyzing the near wall behavior. Such a code is of interest since loss of ions to the walls of the thruster is a significant contributor to inefficiencies. It also contributes to limited thruster life time due to wall sputtering. In addition, novel thrusters have started utilizing magnetic field line topologies in which magnetic field lines impact the wall at highly oblique angles. In such configurations, repulsion of ions from the wall, and a full sheath collapse is predicted. The code was used to investigate these effects, and found that ions indeed become repelled from the walls as the magnetic field incidence angle is increased. The presence of a magnetic mirror or an inclined magnetic field further reduce the wall flux and wall sputter yield by approximately an order of

magnitude.

Finally, the the spatial scale of a satellite was considered. On this spatial scale the important behavior is the expansion of the plasma plume and the backflow of slowly-moving charge exchange ions. This analysis was performed using a time-dependent plume model. Typical plume simulations assume that the thruster operates in steady state, however, such a model does not accurately describe the temporal differences in Hall thruster discharge currents. In this work, a simple hypothetical satellite was used and time-resolved flux to a negatively-biased probe was computed. The probe current closely correlated with the thruster discharge variations.

8.2 Future Work

This thesis identified the basic components for a multiscale approach for modeling Hall thrusters. Although the individual components were demonstrated, a true coupling was not achieved due to the lack of an iterative capability between the kinetic and the 2D thruster codes. Unfortunately, as shown here, HPHall also suffers from a number of limitations that make it an unsuitable candidate for the coupled analysis, especially if non-standard geometries are to be considered. As such, the primary future effort should concentrate on development of an HPHall-like code capable of solving the discharge dynamics of the CHT. That such a code can be developed was demonstrated previously by Garrigues[88]. True implementation of the coupled approach will require for the thruster and the field line codes to communicate with each other, which dictates the need for a single integrated code. The suggested initial effort should concentrated on developing a simplistic code utilizing the basic equations used in HPHall, but operating on a rectilinear mesh. Such a code can be first used to study a simple annular $\vec{E} \times \vec{B}$ discharge to demonstrate the iterative coupling. The code should be developed following modular methodology, such that each computational code becomes a module that can be executed by a master controller.

Secondly, the kinetic code requires additional validation as well as studies of improved wall and collision models. Currently, collisions are modeled using simplistic approaches with the MCC method. Of primary interest here are the Coulomb collisions which become dominant at low energies, yet unlike electron-neutral collisions, cannot be modeled using the MCC approach. For this reason, a mixed MCC/DSMC collision handler should be implemented. The collision handlers should also be investigated further to assure that the correct physical collisional behavior is captured. In addition, the ability of the kinetic code to properly treat the wall flux should also be improved by implementing a variable weight model. Currently, all electrons have the same weight. This results in majority of computational effort spent advancing the bulk population which is of little interest to wall effects. A variable weight model should be able to describe the high energy tail of the distribution function with a higher detail.

Finally, the wall-model code needs to be coupled with the thruster discharge code. The wall code as described in this work utilized analytical magnetic field profile as well as a linearly decaying plasma potential. These properties need to be replaced with the inputs obtained from the 2D discharge code. This modification will likely require change of the underlying numerical mesh representation such that realistic magnetic field line topologies can be captured.

8.3 Source Code

The source code for the 1D kinetic code Lynx described in Chapters 3 and 4 is available upon request by contacting the author at `lubos.brieda@particleincell.com`.

Bibliography

[1] "PPPL Hall thruster experiment (HTX)." `http://htx.pppl.gov`.

[2] V. Zhurin, H. Kaufman, and R. Robinson, "Physics of closed drift thrusters," *Plasma Sources Science and Technology*, vol. 8, no. R1, 1999.

[3] Y. Raitses and N. Fisch, "Parametric investigations of a nonconventional Hall thruster," *Physics of Plasmas*, vol. 8, no. 5, 2001.

[4] M. Keidar and I. I. Beilis, "Sheath and boundary conditions for plasma simulations of a Hall thruster discharge with magnetic lenses," *Applied Physics Letters*, vol. 94, no. 191501, 2009.

[5] D. Sydorenko, A. Smolyakov, I. Kaganovich, and Y. Raitses, "Kinetic simulation of secondary electron emission effects in Hall thrusters," *Physics of Plasmas*, vol. 13, no. 014501, 2006.

[6] I. Kaganovich, Y. Raitses, D. Sydorenko, and A. Smolyakov, "Kinetic effects in a Hall thruster discharge," *Physics of Plasmas*, vol. 14, no. 047104, 2007.

[7] A. Smirnov, Y. Raitses, and N. Fisch, "Electron cross-field transport in a miniaturized cylindrical Hall thruster," *IEEE Transactions on Plasma Science*, vol. 34, no. 2, 2006.

[8] A. I. Morozov and V. Savel'ev, "Theory of the near-wall conductivity," *Plasma Physics Report*, vol. 27, no. 7, 2001.

[9] S. Barral, K. Makowski, Z. Peradzynski, and M. Dudeck, "Is Near-Wall Conductivity a misnomer?," in *39th AIAA Joint Propulsion Conference*, (Huntsville, Al), 2003. AIAA–2003–4855.

[10] G. S. Janes and R. S. Lowder, "Anomalous electron diffusion and ion acceleration in a low-density plasma," *Physics of Fluids*, vol. 9, no. 6, 1966.

[11] F. Taccogna, S. Longo, M. Capitelli, and R. Schneider, "Anomalous transport induced by sheath instability in Hall effect thrusters," *Applied Physics Letters*, vol. 94, no. 251502, 2009.

[12] D. Yu, Z. Wu, and X. Wang, "Numerical simulation for near wall conductivity effect on current profiles in the annular channel of Hall-type stationary plasma thrusters," *Physics of Plasmas*, vol. 12, no. 043507, 2005.

[13] M. Hirakawa, "Electron transport mechanism in a Hall thruster," in *25th International Electric Propulsion Conference*, (Clevelend, OH), 1997.

[14] C. Boniface, L. Garrigues, G. Hagelaar, J. Boeuf, D. Gawron, and S. Mazouffre, "Anomalous cross field electron transport in a Hall effect thruster," *Applied Physics Letters*, vol. 89, no. 161503, 2006.

[15] J. W. Koo and I. D. Boyd, "Modeling of anomalous electron mobility in Hall thrusters," *Physics of Plasmas*, vol. 13, no. 033501, 2006.

[16] M. K. Scharfe, C. Thomas, D. Scharfe, N. Gascon, M. Cappelli, and E. Fernandez, "Comparison of hybrid Hall thruster model to experimental measurements," *IEEE Transactions on Plasma Science*, vol. 36, no. 5, 2008.

[17] L. Garrigues, G. Hagelaar, C. Boniface, and J. P. Boeuf, "Anomalous conductivity and secondary electron emission in Hall effect thrusters," *Journal of Applied Physics*, vol. 100, no. 123301, 2006.

[18] G. Hagelaar, J. Bareilles, L. Garrigues, and J. Boeuf, "Role of anomalous electron transport in a stationary plasma thruster simulation," *Journal of Applied Physics*, vol. 93, no. 1, 2003.

[19] A. Lazurenko, T. Dudok de Wit, C. Cavoit, V. Krasnoselskikh, A. Bouchoule, and M. Dudeck, "Determination of the electron anomalous mobility through measurements of turbulent magnetic field in Hall thrusters," *Physics of Plasmas*, vol. 14, no. 033504, 2007.

[20] C. A. Thomas, *Anomalous Electron Transport In The Hall-Effect Thruster*. PhD thesis, Stanford University, 2006.

[21] J. Szabo, *Fully Kinetic Numerical Modeling of a Plasma Thruster*. PhD thesis, Massachusetts Institute of Technology, 2001.

[22] M. Hirakawa and Y. Arakawa, "Particle simulation of plasma phenomena in Hall thrusters," in *Proceedings of the 24th International Electric Propulsion Conference*, (Moscow, Russia), 1995. IEPC-95-164.

[23] Y. Raitses, D. Staack, M. Keidar, and N. J. Fisch, "Electron-wall interaction in Hall thrusters," *Physics of Plasmas*, vol. 12, no. 047104, 2005.

[24] E. Choueiri, "Plasma oscillations in Hall thrusters," *Physics of Plasmas*, vol. 8, p. 1411, 2001.

[25] F. Darnon, L. Garrigues, J. Boeuf, A. Bouchoule, and M. Lyszyk, "Spontaneous oscillations in a Hall thruster," *IEEE Transactions on Plasma Science*, vol. 27, no. 1, 1999.

[26] J. Parker, Y. Raitses, and N. Fisch, "Transition in electron transport in a cylindrical Hall thruster," *Applied Physics Letters*, vol. 97, no. 091501, 2010.

[27] A. Lazurenko, V. Vial, M. Prioul, and A. Bouchoule, "Experimental investigation of high-frequency drifting perturbations in Hall thrusters," *Physics of Plasmas*, vol. 12, no. 013501, 2005.

[28] F. Chen, *Introduction to Plasma Physics and Controlled Fusion*. Plenum Press, 1983.

[29] "Patience required as AEHF 1 recovery begins new mode." `http://spaceflightnow.com/atlas/av019/101017hct.html`.

[30] M. Keidar, I. D. Boyd, and I. I. Beilis, "Modeling of a high-power thruster with anode layer," *Physics of Plasmas*, vol. 11, no. 4, 2004.

[31] J. Yim, M. Keidar, and L. Boyd, "A hydrodynamic-based erosion model for Hall thrusters," in *29th International Electric Propulsion Conference*, (Princeton, NJ), 2005. IEPC–2005–013.

[32] S. Roy and B. P. Pandey, "Development of a finite element-based Hall-thruster model," *Journal of Propulsion and Power*, vol. 19, no. 5, 2003.

[33] C. Birdsall and A. Langdon, *Plasma physics via computer simulation*. Institute of Physics Publishing, 2000.

[34] Z. Wu, X. Liu, N. Wang, Z. Ning, and D. Yu, "Monte Carlo simulation for electron Near Wall Conductivity in Hall thrusters," in *2010 2nd International Conference on Computer Engineering and Technology*, 2010.

[35] D. Yu, H. Li, Z. Wu, and W. Mao, "Effect of oscillating sheath on near-wall conductivity in Hall thrusters," *Physics of Plasmas*, vol. 14, no. 064505, 2007.

[36] D. Sydorenko, *Particle-In-Cell Simulations of Electron Dynamics in Low Pressure Discharges with Magnetic Fields*. PhD thesis, University of Saskatchewan Saskatoon, 2006.

[37] J. Fife and M. Martinez-Sanchez, "Two-dimensional hybrid particle-in-cell (PIC) modeling of Hall thrusters," in *24th International Electric Propulsion Conference*, (Moscow, Russia), pp. 1213–1224, 1995.

[38] J. Boeuf and L. Garrigues, "Low frequency oscillations in a stationary plasma thruster," *Journal of Applied Physics*, vol. 84, no. 7, 1998.

[39] J. Koo and I. Boyd, "Computational model of a Hall thruster," *Computer physics communications*, vol. 164, no. 1-3, pp. 442–447, 2004.

[40] F. Parra, E. Ahedo, J. Fife, and M. Martinez-Sanchez, "A two-dimensional hybrid model of the Hall thruster discharge," *Journal of Applied Physics*, vol. 100, no. 023304, 2006.

[41] R. Hofer, I. Mikellides, I. Katz, and D. Goebel, "BPT-4000 Hall thruster discharge chamber erosion model comparison with qualification life test data," in *30th International Electric Propulsion Conference*, pp. 2007–267, 2007.

[42] M. Scharfe, *Electron Cross Field Transport Modeling in Radial-Axial Hybrid Hall Thruster Simulations*. PhD thesis, Staford University, 2009.

[43] E. Ahedo, R. Santos, and F. Parra, "Fulfillment of the kinetic Bohm criterion in a quasineutral particle-in-cell model," *Physics of Plasmas*, vol. 17, p. 073507, 2010.

[44] M. Nakles, W. Hargus, and V. D., "Comparison of numerical and experimental near-field ion velocity distributions of the BHT-200-X3 Hall thruster," in *42nd AIAA Joint Propulsion Conference*, (Sacramento, CA), 2006. AIAA-2006-4479.

[45] S. Y. Cheng, *Modeling of Hall thruster lifetime and erosion mechanisms*. PhD thesis, Massachusetts Institute of Technology, 2007.

[46] A. I. Morozov, Y. V. Esinchuk, G. N. Tilinin, A. Trofimov, Y. A. Sharov, and G. Y. Shchepkin, "Plasma accelerator with closed electron drift and extended acceleration zone," *Soviet Physics – Technical Physics*, vol. 17, no. 1, 1972.

[47] M. Lampe, G. Joyce, W. Manheimer, and S. P. Slinker, "Quasi-neutral particle simulation of magnetized plasma discharges: General formalism and application to ecr discharges," *IEEE Transcations on Plasma Science*, vol. 26, no. 6, 1998.

[48] L. Brieda, R. Kafafy, J. Pierru, and J. Wang, "Development of the Draco code for modeling electric propulsion plume interactions," in *Proceedings of the 40th Joint Propulsion Conference*, (Fort Laudardale, FL), pp. 1–21, 2004.

[49] I. Boyd and R. A. Dressler, "Far field modeling of the plasma plume of a Hall thruster," *Journal of Applied Physics*, vol. 92, no. 4, 2002.

[50] D. B. VanGilder, I. Boyd, and M. Keidar, "Particle simulations of a Hall thruster plume," *Journal of Spacecraft and Rockets*, vol. 37, no. 1, 2000.

[51] M. Santi, S. Cheng, M. Celik, M. Martinez-Sanchez, and J. Peraire, "Further development and preliminary results of the Aquila hall thruster plume model," in *39th AIAA Joint Propulsion Conference and Exhibit*, (Huntsville, Alabama), 2003. AIAA-2003-4873.

[52] M. Nakles, L. Brieda, G. D. Reed, W. A. Hargus, and R. Spicer, "Experimental and numerical examination of the BHT-200 Hall thruster plume," in *43rd AIAA Joint Propulsion Conference*, (Cincinnati, OH), 2007. AIAA-2007-5305.

[53] D. Sydorenko, A. Smolyakov, I. Kaganovich, and Y. Raitses, "Modification of electron velocity distribution in bounded plasmas by secondary electron emission," *IEEE Transactions on Plasma Science*, vol. 34, no. 3, 2006.

[54] C. Birdsall, "Particle-in-cell charged-particle simulations, plus Monte Carlo Collisions with neutral atoms, PIC-MCC," *Plasma Science, IEEE Transactions on*, vol. 19, no. 2, pp. 65–85, 1991.

[55] Wikipedia, "Tridiagonal matrix algorithm," 2011. `http://en.wikipedia.org/wiki/Tridiagonal_matrix_algorithm`.

[56] J. Fox, *Advances in Fully-Kinetic PIC Simulations of a Near-Vacuum Hall Thruster and Other Plasma Systems*. PhD thesis, Massachusetts Institute of Technology, 2007.

[57] J. Boris and R. Lee, "Non-physical self forces in electromagnetic plasma-simulation algorithms," in *NRL Memorandum Report 2418*, pp. 1–10, 1972.

[58] M. Lieberman and A. Lichtenberg, *Principles of plasma discharges and materials processing*. Wiley-interscience, 2005.

[59] G. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford Science Publications, 1994.

[60] J. M. Fife, *Hybrid-PIC Modeling and Electrostatic Probe Survey of Hall Thrusters*. PhD thesis, Massachusetts Institute of Technology, 1998.

[61] R. Hofer, I. Mikellides, I. Katz, and D. Goebel, "Wall sheath and electron mobility modeling in hybrid-PIC Hall thruster simulations," in *43rd AIAA Joint Propulsion Conference*, (Cincinnati, OH), 2007. AIAA–2007–5267.

[62] E. Ahedo and J. M. Gallardo, "Effects of the radial plasma-wall interaction on the Hall thruster discharge," *Physics of Plasmas*, vol. 10, no. 8, 2003.

[63] L. Brieda, M. Keidar, Y. Raitses, and N. Fisch, "Self-consistent calculation of electron transport in a cylindrical Hall thruster," in *31st International Electric Propulsion Conference*, (Ann Arbor, MI), 2009.

[64] Y. Raitses, A. Smirnov, and N. Fisch, "Effects of enhanced cathode electron emission on Hall thruster operation," *Physics of Plasmas*, vol. 16, no. 057106, 2009.

[65] A. Smirnov, Y. Raitses, and N. Fisch, "Plasma measurements in a 100 w cylindrical Hall thruster," *Journal of Applied Physics*, vol. 95, no. 5, 2004.

[66] R. I. Samanta Roy, *Numerical Simulation of Ion Thruster Plume Backflow for Spacecraft Contamination Assessment*. PhD thesis, Massachusetts Institute of Technology, 1995.

[67] M. Keidar and L. Brieda, "Modeling electron transport within the framework of hydrodynamic description of Hall thrusters," in *44th AIAA Joint Propulsion Conference*, (Hartford, CT), 2008. AIAA–2008–5186.

[68] R. Spicer, "Validation of the Draco particle-in-cell code using Busek 200W Hall thruster experimental data," Master's thesis, Virginia Polytechnic Institute and State University, 2007.

[69] B. Reid and A. Gallimore, "Plasma potential measurements in the discharge of a 6-kw Hall thruster," in *44th Joint Propulsion Conference*, (Hartford, CT), 2008.

[70] A. Fruchtman and A. Cohen-Zur, "Plasma lens and plume divergence in the Hall thruster," *Applied Physics Letters*, vol. 89, no. 111501, 2006.

[71] M. Keidar, I. D. Boyd, and I. I. Beilis, "Plasma flow and plasma-wall transition in Hall thruster channel," *Physics of Plasmas*, vol. 8, no. 12, 2001.

[72] M. Keidar and I. D. Boyd, "On the magnetic mirror effect in Hall thrusters," *Applied Physics Letters*, vol. 87, no. 121501, 2005.

[73] A. Dunaevsky, Y. Raitses, and N. Fisch, "Secondary electron emission from dielectric materials of a Hall thruster with segmented electrodes," *Physics of Plasmas*, vol. 10, p. 2574, 2003.

[74] Y. Garnier, V. V., J. F. Roussell, and J. Bernard, "Anomalous conductivity and secondary electron emission in Hall effect thrusters," *Journal of Vacuum Science and Technology A*, vol. 17, p. 3246, 1999.

[75] J. Linnell and A. D. Gallimore, "Internal plasma potential measurements of a Hall thruster using plasma lens focusing," *Physics of Plasmas*, vol. 13, no. 103504, 2006.

[76] B. Beal and A. Gallimore, "Energy analysis of a Hall thruster cluster," in *28th International Electric Propulsion Conference*, 2003. IEPC-2003-055.

[77] J. Wang, D. Brinza, and M. Young, "Three-dimensional particle simulations of ion propulsion plasma environment for deep space 1," *Journal of Spacecraft and Rockets*, vol. 38, pp. 433–440, 2001.

[78] A. Tribble, *Fundamentals of contamination control*. SPIE, 2000.

[79] M. Mandell, V. Davis, D. Cooke, A. Wheelock, and C. Roth, "Nascap-2k spacecraft charging code overview," *IEEE Transactions on Plasma Science*, vol. 34, no. 5, 2006.

[80] M. Tajmar, W. Meissl, J. del Amo, B. Foing, H. Laakso, G. Noci, M. Capacci, A. Malkki, W. Schmidt, and F. Darnon, "Charge-exchange plasma contamination on SMART-1: First, measurements and model verification," in *40th AIAA Joint-Propulsion-Conference*, (Fort Lauderdale, FL), 2004.

[81] M. Santi, S. Cheng, M. Celik, M.-S. M., and P. J, "Further development and preliminary results of the Aquila Hall thruster plume model," in *39th Joint Propulsion Conference*, (Huntsville, Al), 2003.

[82] J. Forest, A. Hilgers, B. Thiébault, L. Eliasson, J. Berthelier, and H. de Feraudy, "An open-source spacecraft plasma interaction simulation code PicUp3D: tests and validations," *Plasma Science, IEEE Transactions on*, vol. 34, no. 5, pp. 2103–2113, 2006.

[83] M. Gibbons, D. Kirtley, D. VanGilder, and J. Fife, "Flexible three-dimensional modeling of electric thrusters in vacuum chambers," in *39th AIAA Joint Propulsion Conference*, (Huntsville, Al), 2003. AIAA-2003-4872.

[84] L. Brieda, "Development of the Draco ES-PIC code and fully-kinetic simulation of ion beam neutralization," Master's thesis, Virginia Tech, 2005.

[85] J. Pierru, "Development of a parallel electrostatic PIC code for modeling electric propulsion," Master's thesis, Virginia Polytechnic Institute and State University, 2005.

[86] R. Kafafy, T. Lin, Y. Lin, and J. Wang, "Three-dimensional immersed finite element methods for electric field simulation in composite materials," *International journal for numerical methods in engineering*, vol. 64, no. 7, pp. 940–972, 2005.

[87] D. Hastings and H. Garrett, *Spacecraft-environment interactions*. Cambridge University Press, Cambridge, UK, 2004.

[88] L. Garrigues, G. Hagelaar, J. P. Boeuf, Y. Raitses, A. Smirnov, and N. J. Fisch, "Simulations of a miniaturized cylindrical Hall thruster," *IEEE Transaction on Plasma Science*, vol. 36, no. 5, 2008.