Computational Modeling of a Hall Thruster Plasma Plume in a Vacuum Tank

by

Shannon Yun-Ming Cheng

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2002

Author . . . .
Department of Aeronautics and Astronautics
February 1, 2002

Certified by . . . .
Manuel Martinez-Sanchez
Professor
Thesis Supervisor

Accepted by . . . .
Wallace E. Vander Velde
Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Computational Modeling of a Hall Thruster Plasma Plume in a Vacuum Tank

by

Shannon Yun-Ming Cheng

Submitted to the Department of Aeronautics and Astronautics
on February 1, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Hall thrusters have become a tempting alternative to traditional chemical propulsion systems due to the great mass savings they provide through high specific impulses. However, a major stumbling block to their widespread integration is uncertainty about the thruster plume's interaction with spacecraft components. While in-space data is difficult to collect, much experimental data from vacuum tank tests is readily available. Effectively taking advantage of this wealth requires understanding of the effects from imperfect ground test conditions. A previous plume model, *Quasi3*, has been upgraded to better simulate the vacuum tank environment primarily through improvements to the source model, the collision method, and the sputtering method. The code is now more accurate and provides insight into phenomena such as background pressure consequences, sputtering and sputtered material deposition.

Thesis Supervisor: Manuel Martinez-Sanchez
Title: Professor

Acknowledgments

I would like to thank my advisor, Professor Martinez-Sanchez, for both his great patience and his enthusiasm. An open door always greets me and my incessant questions, regardless of time of day or the huge piles of other work he has. His genuine interest and curiosity in my work is contagious and keeps me motivated and excited about research.

Thanks are also in order for my whole family — Mom, Dad, Serena, and Adrian. They all possess that wonderful gift of being able to raise my spirits even from thousands of miles away, and I hope they know how much they mean to me.

I owe a great debt to Mike Fife for sending me the original version of *Quasi3* and helping me get *HPHall* working with the BHT-200 grid. I literally wouldn't have gotten anywhere with my research without his help.

One of the great perks of being a student is all the cool people you get to hang out with. Thank you to everyone in SPL — Tatsuo, Paulo, Luis Fernando, Jorge, Vincent, Anne, and Murat — for helping me whenever I need it, making the lab a fun environment to come and work in, and generally enhancing grad student life. Thank you to my roommates, Andrew, Chris, and Gong Ke, for understanding when I wasn't around much, giving me great advice, and always making me smile. Special thanks to Raffi, Jessica, and Ethan for all the moral support and for bailing me out of numerous computer troubles. And thank you to Al and Julie for getting me through the last month.

Last, but certainly not least, thank you to Leo and Pinky for being with me from the very beginning and never once complaining.

Contents

1 Introduction **19**

1.1 Hall Thrusters 19

1.2 Motivation 19

1.2.1 Advantage of Electric Propulsion 19

1.2.2 Hall Thruster Integration Issues 20

1.2.3 ETEEV 22

1.3 Outline of Research 23

2 Original Plume Model **25**

2.1 Quasi3 25

2.1.1 PIC method 25

2.1.2 Collision Model 27

2.1.3 Computational Grid 27

2.1.4 Source Model 28

2.1.5 Surface Interaction Model 30

2.2 Later Modifications to Quasi3 31

2.3 Starting Point of this Research 32

3 Source Model **33**

3.1 Overview 33

3.2 Preliminary Source Model 35

3.2.1 Exit Plane Distributions 39

3.2.2 Additional Parameters 39

3.3 Improved Source Model 44

3.3.1 Exit Plane Distributions 46

3.3.2 Additional parameters 71

3.4 Comparison of Source Models 71

4 Collisons 75

4.1 Overview 75

4.2 Charge Exchange Cross-section 75

4.3 Elastic collisions 76

4.4 Neutral weighting selection 78

5 Sputtering 81

5.1 Overview 81

5.2 Normal Sputtering Yield 82

5.3 Angular Dependence of Sputtering Yield 84

5.4 Low-energy Sputtering Method 88

5.5 Angular Distribution of Sputtered Material 88

5.6 Implementation of Sputtering Method 92

6 Results and Conclusion 95

6.1 Overview 95

6.2 Current Density Distribution 95

6.2.1 Comparison to Experimental Data 99

6.2.2 Effect of Background Pressure 100

6.3 Simulation in Vacuum Tank 114

6.3.1 Simulation Geometry 114

6.3.2 Tank Simulation Results 118

6.3.3 Tank Filling 135

6.4 Sources of Error 135

6.4.1 Charge Exchange 135

6.4.2 Linking of Source and Plume Models 142

6.4.3 Lack of Experimental Data 144

6.5 Conclusions 146

6.6 Future Work 146

A Other Code Modifications **149**

A.1 Sheath Model 149

A.1.1 Sheath Drop 149

A.1.2 Conducting Surface Sheaths 150

A.2 Potential Reference Method 151

A.3 Visualization Software 152

B Quasi3 Manual **153**

B.1 Overview 153

B.2 Mesh3 154

B.3 Quasi3 162

B.4 Visualization Software 164

B.5 Adding Sputtered Materials to Simulation 166

List of Figures

1-1 Cross-section of typical Hall thruster. 20

2-1 Particle-In-Cell (PIC) methodology. 26

2-2 Experimental measurements of near-field current density. [6] 30

3-1 BHT-200. 34

3-2 Spatial grid for the BHT-200 geometry. 36

3-3 Side view of BHT-200. 37

3-4 Simulated performance of BHT-200 from *Hall*. 38

3-5 Space potential for BHT-200. 38

3-6 Simulated Xe$^+$ flux. 40

3-7 Modeling of BHT-200 exit plane. 41

3-8 Calculation of azimuthal ion drift velocity. 43

3-9 Simulated performance of BHT-200 from *HPHall*. 45

3-10 Exit plane for improved source model. 46

3-11 Xenon single ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of Xe$^+$ ranging from 0 to 3×10^{12}. 47

3-12 Xenon single ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of Xe$^+$ ranging from 0 to 4×10^{12}. 48

3-13 Derivation of Xe$^+$ number density distribution. 49

3-14 Split of single ion populations. 50

3-15 Single ion near-side population velocity distributions. 51

3-16 Near-side population single ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of near-side Xe$^+$ ranging from 0 to 3×10^{12}. 52
3-17 Near-side population single ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of near-side Xe$^+$ ranging from 0 to 4×10^{12}. 53
3-18 Axial near-side ion temperature. 54
3-19 Single ion far-side population velocity distributions. 55
3-20 Far-side population single ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of far-side Xe$^+$ ranging from 0 to 1×10^{12}. 57
3-21 Far-side population single ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of far-side Xe$^+$ ranging from 0 to 2×10^{12}. 58
3-22 Axial far-side ion temperature. 59
3-23 Derivation of Xe^{++} number density distribution. 61
3-24 Split of double ion populations. 62
3-25 Double ion near-side population velocity distributions. 63
3-26 Near-side population double ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of near-side Xe^{++} ranging from 0 to 3×10^{11}. 64
3-27 Near-side population double ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of near-side Xe^{++} ranging from 0 to 4×10^{11}. 65
3-28 Axial near-side double ion temperature. 66
3-29 Double ion far-side population velocity distributions. 67
3-30 Far-side population double ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of far-side Xe^{++} ranging from 0 to 1×10^{11}. 68

3-31 Far-side population double ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of far-side Xe^{++} ranging from 0 to 15×10^{10}. 69
3-32 Axial far-side double ion temperature. 70
3-33 Comparison of source models - single ions. 72
3-34 Comparison of source models - double ions. 72
3-35 Comparison of source models. 73
3-36 Ion current density versus divergence angle, measurements scaled to 1 m radius from the thruster exit plane. [12] 74

4-1 Simulated retarding potential analyzer (RPA) measurements for P = 5×10^{-6} $Torr$. 77

5-1 Normal incidence sputtering yield for silicon by xenon. 83
5-2 Sputtering yield of silicon by incident xenon at 200 eV. 86
5-3 Dominant sputtering mechanism near normal incidence. 87
5-4 Dominant sputtering mechanism at oblique incidence. 87
5-5 Sputtering yield of silicon by xenon for various incident angles. 89
5-6 Near-threshold energy sputtering curves for xenon striking silicon. . . 90
5-7 Sputtering angles. 91
5-8 Angular distribution of sputtered material for $E/E_{th} = 2$. 93

6-1 3D view of current density simulation domain. 96
6-2 Embedded meshes for current density simulation domain. 97
6-3 Comparison of current density distributions in vacuum. 98
6-4 Comparison of scaled and simulated data at 60 cm. 99
6-5 Current density distributions for various background pressures at 60 cm.100
6-6 Simulated RPA data at 70° for various background pressures. 101
6-7 Velocity phase-space plot for single ions. 102
6-8 Velocity phase-space plot for double ions. 103
6-9 Charge exchange ions for different background pressures. 104

6-10 Source ions for different background pressures. 105
6-11 Current density distributions for various background pressures at 60 *cm* with inclusion and exclusion of source neutrals. 107
6-12 Current density [mA/cm^2] versus background pressure [$Torr$] at various angles. Solid line represents current density at P = 0 $Torr$. Dashed line represents current density without source neutrals at P = 0 $Torr$. 108
6-13 Current density [mA/cm^2] versus background pressure [$Torr$] at various angles. Solid line represents current density at P = 0 $Torr$. Dashed line represents current density without source neutrals at P = 0 $Torr$. 109
6-14 Current density [mA/cm^2] versus background pressure [$Torr$] at various angles. Solid line represents current density at P = 0 $Torr$. Dashed line represents current density without source neutrals at P = 0 $Torr$. 110
6-15 Fractional split of source and CEX ions versus background pressure [$Torr$] for various angles. Horizontal axis is pressure from 0 to 4×10^{-5} $Torr$. 111
6-16 Fractional split of source and CEX ions versus background pressure [$Torr$] for various angles. Horizontal axis is pressure from 0 to 4×10^{-5} $Torr$. 112
6-17 Fractional split of source and CEX ions versus background pressure [$Torr$] for various angles. Horizontal axis is pressure from 0 to 4×10^{-5} $Torr$. 113
6-18 Simulated tank geometry. 115
6-19 Witness plate orientation (top view). 116
6-20 Witness plate orientation (back view). 117
6-21 Xe particles after 1000 iterations. 119
6-22 Xe particles after 1000 iterations. 120
6-23 Witness plate 1, west wall, silver material. 123
6-24 Witness plate 1, east wall, silver material. 124
6-25 Witness plate 2, west wall, silver material. 125
6-26 Witness plate 2, east wall, silver material. 126

6-27 East wall, iron material. 127

6-28 North wall, iron material. 128

6-29 Up wall, iron material. 129

6-30 West wall, iron material. 130

6-31 Up wall, iron material. 131

6-32 West wall, iron material. 132

6-33 Up wall with background density, iron material. 133

6-34 West wall with background density, iron material. 134

6-35 Particle totals for tank filling case. 136

6-36 Species temperatures for tank filling case. 137

6-37 Species pressures for tank filling case. 138

6-38 Xe-Xe$^+$ CEX rate $[1/m^3/s]$. 140

6-39 Xe-Xe^{++} CEX rate $[1/m^3/s]$. 140

6-40 Region used to estimate missing CEX from source model. 141

6-41 Effect of electron temperature. 143

6-42 *HPHall* exit plane averages. 144

6-43 Average potential in cross-section of plume. 145

A-1 Sheath drop. 150

List of Tables

2.1 Collisions simulated in *Quasi3*. 28

2.2 SPT-100 nominal operating parameters. 29

2.3 Source model parameters for SPT-100. 29

3.1 BHT-200 nominal operating parameters. 33

3.2 Simulated performance of BHT-200 from *Hall*. 35

3.3 Preliminary BHT-200 source model parameters. 44

3.4 Simulated performance of BHT-200 from *HPHall*. 45

3.5 Improved BHT-200 source model parameters. 71

5.1 Sputtering yield calculation parameters with Xe as the projectile. . . 85

5.2 Sputtering yield thresholds. 88

6.1 CEX production rates in engine region before exit plane. 141

6.2 CEX production rates in plume region for various background pressures.142

B.1 Variables in a .in file. 163

B.2 *Quasi3* switches. 163

B.3 *Mesh3* switches. 163

B.4 Changes to *Mesh3*. 166

B.5 Changes in *Quasi3*. 167

Chapter 1

Introduction

1.1 Hall Thrusters

Hall thrusters are members of the electric propulsion family, characterized by an annular acceleration channel and a radial magnetic field. Electrons are emitted from an external cathode and backstream towards the anode where they encounter the magnetic field. Because of their small Larmor radius and long collision mean free path, electrons are prevented from diffusing across the magnetic field easily and are effectively captured in azimuthal drifts along field lines. Neutral propellant gas, typically xenon, is injected into the channel near the anode and undergoes ionization upon contact with the trapped electrons. The resultant heavy ions have a Larmor radius much larger than the thruster's dimensions and are accelerated out of the channel by the axial electric field with little influence from the magnetic field. Figure 1-1 shows the cross-section of a typical Hall thruster.

1.2 Motivation

1.2.1 Advantage of Electric Propulsion

Hall thrusters and other types of electric propulsion have recently attracted much interest as they provide high specific impulses at relatively high efficiencies. Unlike

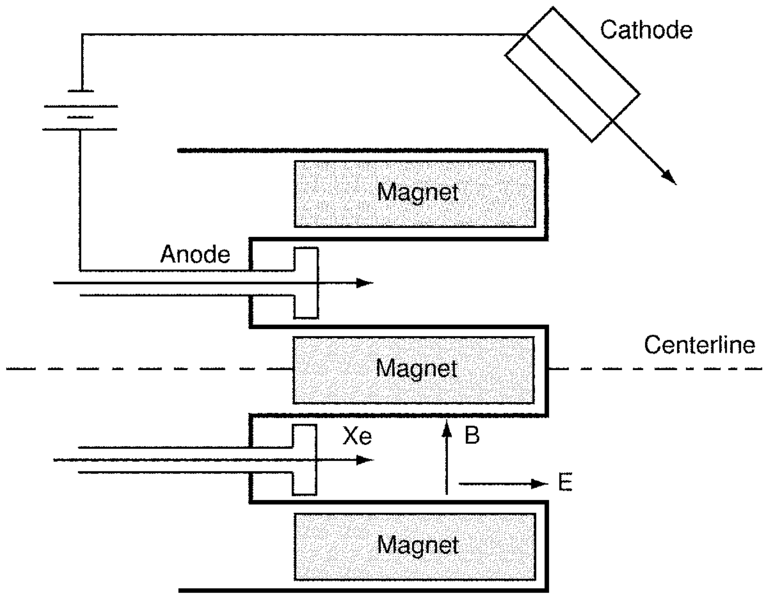

Figure 1-1: Cross-section of typical Hall thruster.

conventional chemical propulsion systems, electric propulsion devices do not rely on the internal energy stored within their propellant. Instead, an external power source is used to impart energy to the working fluid. As no inherent limitation due to the propellant exists, a high I_{sp} is attainable which translates to a highly desirable mass savings. Higher exhaust velocities and thus higher specific impulses are only constrained by the power processing unit (PPU) which provides the external energy and is the primary contributor to the electric propulsion system's mass. By balancing I_{sp} with PPU mass, Hall thrusters are found to fall in an optimum operation regime well-suited for missions such as station-keeping and orbit transfers.

1.2.2 Hall Thruster Integration Issues

Despite numerous advantages, reservations about integrating Hall thruster technology into space systems exist because of various unknowns. One large area of concern is the interaction the thruster plume has with the spacecraft. As a result, many efforts have been made to study these effects through both experiment and simulation. To date, very little in-space plume data has been taken as such experiments are both difficult and expensive. Conversely, a multitude of vacuum tank experiments have been performed or are planned for the near future. While these on-ground tests provide much insight into the behavior of the Hall thruster and its plume, interpretation of the collected data also needs to take into account effects due to imperfect experimental conditions.

One such limitation is the finite capability of the vacuum pumps. Because perfect vacuum conditions cannot be achieved, a nominal background neutral density exists that interacts with the plume through collisions. While ionizing collisions with these background neutrals cause an increase in measured thrust if ions are created at higher than background potential, the effects of charge exchange (CEX) collisions are of greater concern. These collisions occur when the positive charge of a fast-moving source ion is given to a slow-moving background neutral in a resonant process, resulting in a high-energy neutral and a low-energy ion. The potential hump of the plume is not as easily overcome by these CEX ions and they are pushed laterally more than their energetic source ion counterparts. This phenomenon is reflected in higher values of current density at outlying angles of the distribution.

An unwanted consequence of CEX ions, whether formed from interaction with a source or a background neutral, is sputtering of objects in contact with the plume. To avoid direct impingement by the plume on spacecraft components such as solar arrays, thrusters are generally canted at an angle. Although the thrust vector is no longer ideally aligned, the main beam is now diverted away from the sensitive surfaces. Unfortunately, though the lower energy CEX ions have less potential to cause harm, their location in the wings of the plume puts them dangerously close to the spacecraft components trying to be avoided. Some CEX ions may even be accelerated backwards and strike the spacecraft's main body. While it is hoped that damage on-orbit is kept to a minimum, it is difficult to discern its magnitude from ground experiments. Separating the effects of a background neutral density and its subsequent elevation of CEX ion levels is not a simple task. Another integration issue is foreign material from the thruster itself depositing on sensitive spacecraft surfaces. In this case, the closed environment of the vacuum tank is an additional drawback as sputtering off of facility walls and instruments may be confused with that from the engine.

The combination of a contained environment and the limited pumping capability causes recirculation of plasma. Plasma that would normally be dispersed in the vacuum of space is retained and may induce unexpected effects in the testing domain.

Due to these uncertainties, drawing conclusions on in-space plume behavior based on vacuum tank experiments is difficult. The goal of this research is to improve the ability of a numerical plume simulation to model a tank environment. If tank effects on thruster operation can be accurately captured by the physics of a simulation and verified against available empirical data taken on the ground, then greater confidence in the projections of the plume's response in a perfect vacuum is earned. Furthermore, the model then can act as a tool for correlation between the plume's behavior on ground and in actual orbital conditions, aiding in integration of the thruster into spacecraft.

1.2.3 ETEEV

The Electric Thruster Environmental Effects Verification (ETEEV) experiment is a joint effort underway between MIT, WPI, Draper Laboratory, Busek, and the Air Force Research Laboratory. The purpose of the project is to collect in-space plume behavior of a Hall thruster and a Pulsed Plasma thruster as a Shuttle Hitchhiker payload scheduled to fly in 2003. More detail about the experiment can be found in the work of Thomas and Pacros [18, 11].

Design of the Hall thruster side of the experiment is being handled by MIT. A BHT-200 provided by Busek is currently being prepared for the flight in a vacuum tank at MIT. This research focuses on simulating the ETEEV setup as better understanding of the plume aids in experiment design matters such as instrument selection and placement. A reciprocal benefit exists in that both on-ground vacuum tank and in-space plume data will be available in the near future for code verification.

1.3 Outline of Research

This research begins with an existing numerical plume model which is then upgraded to better simulate plumes in a tank environment. Chapter 2 gives an overview of the state of the original plume model prior to any changes. Chapter 3 describes the development of better source models to simulate the BHT-200. Chapter 4 presents efforts made to improve modeling of collisions. Chapter 5 covers enhancements to the sputtering methodology. Chapter 6 presents results and conclusions about the research. Finally, additional code modifications and a brief user's manual of the upgraded code are included as appendices.

Chapter 2

Original Plume Model

2.1 Quasi3

Quasi3 is a hybrid PIC-DSMC three-dimensional simulation of a Hall thruster plasma plume written by Oh [10]. Expansion of the plume is modeled in a user-specified geometry and background pressure. Estimates of surface erosion rates for three materials are calculated based on incident particle flux and energy. Contour plots of these rates, particle fluxes and energies on simulation surfaces, and the current density distribution are the main results extracted from the simulation.

2.1.1 PIC method

Particle-In-Cell (PIC) methods treat gases as a collection of particles. Simulated particles represent a much larger number of real particles and are referred to as macroparticles. In the case of a plasma, several species must be accounted for — ions, neutrals, and electrons. Macroparticles representing these species move throughout a computational grid. At each time step, trajectories are altered due to self-induced and externally applied force fields. For a plasma, the self-induced electric field is calculated by weighting charged species to the grid nodes to determine the local charge density as in Figure 2-1. Poisson's equation,

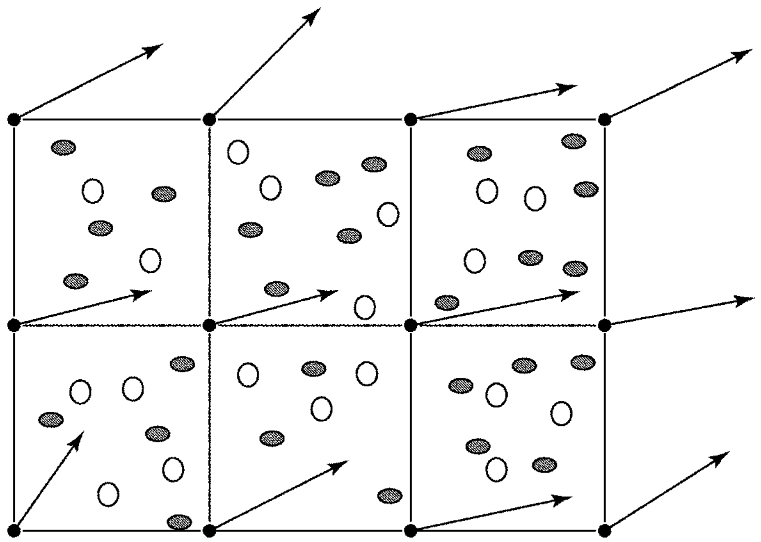

Figure 2-1: Particle-In-Cell (PIC) methodology.

$$\nabla^2\phi(x,y,z) = \frac{-\rho(x,y,z)}{\epsilon_o}, \tag{2.1}$$

is then solved on the grid and the resulting potential is differentiated to yield the electric field,

$$\mathbf{E} = -\nabla\phi. \tag{2.2}$$

Fields are weighted back to macroparticles and the resultant forces are used in integration of the equations of motion to move the macroparticles to their positions for the next time step. This procedure is repeated until the total simulation time is reached.

Quasi3 uses a hybrid-PIC methodology in which the electrons are modeled as a fluid instead of as particles. Further simplification is achieved by assuming quasi-neutrality and reducing the general electron momentum equation to the Boltzmann relationship,

$$n_e = n_o e^{\frac{-e\phi}{kT_e}}. \tag{2.3}$$

Magnetic fields in the plume are considered negligible, so only the electric field is modeled. Xenon ions are weighted to the computational grid and the resulting local ion charge density is equated to the electron charge density because of quasi-neutrality,

$$n_i \approx n_e. \tag{2.4}$$

In this manner, solving of Equation 2.1 is avoided. Equation 2.3 can then be solved for the potential with an assumed constant electron temperature of 2 eV throughout the plume. The electric field is calculated with Equation 2.2 and particle positions and velocities are updated due to the resultant forces as in the traditional PIC method.

2.1.2 Collision Model

In addition to the electric field, collisions also affect macroparticle motion. A Direct Simulation Monte Carlo (DSMC) method is used to model the collisions listed in Table 2.1. During each iteration, a global time counter is incremented by the simulation time step. Collisions are then performed on a cell-by-cell basis in which pairs of collision partners are selected randomly. Each cell is stepped through and if a minimum number of particles is not within the cell, collisionality is deemed improbable, the local time counter is incremented by the time step, and the next cell is proceeded to. If collisions are likely, two particles are chosen randomly and undergo a selection-rejection scheme to determine whether the collision occurs or not. If a collison occurs, the collision frequency for that process is inverted and used to increment the local time counter. A multi-species time counter that supports multiple collision species and types and variable macroparticle weighting calculates this time step. The velocities of particles involved in the collision are then altered based on the collision type. This process repeats until the local time counter exceeds the global time counter at which point the simulation continues to the next cell.

2.1.3 Computational Grid

The entire simulation is performed on a Cartesian grid generated by a program called *Mesh3*. The grid represents the simulation domain and contains information about

Charge Exchange	Elastic
Xe - Xe^+	Xe - Xe^+
Xe - Xe^{++}	Xe - Xe^{++}
	Xe - Xe

Table 2.1: Collisions simulated in *Quasi3*.

the parent and embedded meshes used in the hybrid-PIC and DSMC routines. Embedded meshes are used to obtain finer resolution in areas of interest in the domain. Simulation objects such as the thruster, tank walls, or solar arrays are also included in the grid. Because a Cartesian grid is used throughout, meshes and objects are constrained to being rectangular in shape.

2.1.4 Source Model

The original source model represents the plasma flow from a SPT-100 thruster which has nominal operating parameters as summarized in Table 2.2. This plasma is composed of ions, neutrals, and electrons. Since electrons are described by a simplified fluid model, source electrons are not modeled directly and are accounted for by the Boltzmann equation. Table 2.3 summarizes source model parameters which are used along with those in Table 2.2 to calculate number fluxes of source ions and neutrals using,

$$\dot{n}_i = \dot{m} f_a \eta_u, \tag{2.5}$$

$$\dot{n}_n = \dot{m} f_a (1 - \eta_u). \tag{2.6}$$

At each iteration, these rates and the time step determine the number of particles to be inserted into the simulation. An empirical model of the thruster's exit plane ion distribution is constructed from experimental measurements of near-field current density of a SPT thruster taken by Gavryushin and Kim [6]. The magnitude and direction of ion flow as a function of radial position are extracted from Figure 2-2 and fitted to high-order polynomials.

Specific Impulse, I_{sp}	1610 s
Anode Specific Impulse	1735 s
Thrust	84.9 mN
Discharge Voltage, V_D	300 V
Discharge Current, I_a	4.5 A
Propellant Flow Rate, $\dot{m}$	5.0 mg/s
Propellant Fraction to Cathode	7.5%

Table 2.2: SPT-100 nominal operating parameters.

Exit Plane Outer Radius, r_1	0.100 m
Exit Plane Inner Radius, r_2	0.056 m
Anode Propellant Fraction, f_a	0.929
Anode Utilization Fraction, η_u	0.95
Cathode Orifice Radius	0.0005 m
Cathode Axial Offset	0.01 m
Cathode Radial Offset	0.075 m
Xe$^+$ Axial Drift Velocity	17,020 m/s
Xe$^+$ Azimuthal Drift Velocity	250.0 m/s
Xe$^+$ Axial Temperature	34 eV
Xe$^+$ Radial Temperature	0.689655172 eV
Xe$^+$ Azimuthal Temperature	0.068975517 eV
Xe Temperature	0.086 eV

Table 2.3: Source model parameters for SPT-100.

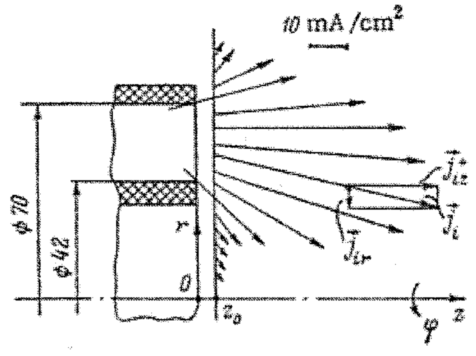

Figure 2-2: Experimental measurements of near-field current density. [6]

The magnitude of the current density is converted to a number density by assuming all ions are singly-charged and leave the thruster with a drift velocity of 17,020 m/s. This number density is integrated across the exit plane to find total ion flow and then normalized, resulting in a probability distribution. Integrating this result gives a radial cumulative distribution function which decides where particles are injected radially. The direction of the current density vectors are used to derive a beam divergence angle function to choose what direction to insert particles in. The assumed drift speed of 17,020 m/s is broken into radial and axial components based on the divergence angle. At each time step, the number of charged particles to be inserted is calculated with Equation 2.5 and a user-specifed fraction is assumed to be doubly ionized. Double ions are assumed to have the same distribution as the single ions and are injected in the same manner, the only difference being their double charge.

2.1.5 Surface Interaction Model

A crude surface interaction model predicts erosion of material off object surfaces in the simulation. When particles cross a cell boundary, the boundary type is determined from the information stored in the grid. If the boundary corresponds to an object boundary, sputtering due to the particle impact is calculated. Because quasi-neutrality is assumed in the model, resolution of the non-quasi-neutral sheath boundary is not accomplished. Thus, a sheath interaction model is used to account for this region. Typically, the object surface will float negative with respect to the pre-sheath plasma potential, causing positively charged ions to be accelerated towards the wall. The potential drop across the sheath is calculated with,

$$\phi = \frac{-kT_e}{e} \ln \left[\frac{4\Gamma_i}{n_e \bar{c}_e} \right]. \tag{2.7}$$

Energy due to acceleration through this sheath potential is added to the particle impacting the object surface. This energy is then used to calculate the sputtering yield of silver, quartz, and silicon from fits to experimental sputtering data. These fits are linear relationships between sputtering yield and incident particle energy generated from data for normally-incident particles. Sputtering yields are then translated to erosion rates for each material and averaged over the duration of the simulation. Ions striking surfaces are deleted from the simulation while neutrals are reflected specularly.

2.2 Later Modifications to Quasi3

Since its completion, Oh's work has been expanded upon. In studying issues related to integrating a Hall thruster acceleration channel model with *Quasi3*, Qarnain [13] modified the original source model. Fife's [5] hybrid PIC-MCC two-dimensional engine code, *Hall*, is used to generate a simulated exit plane current density distribution from which cumulative radial distribution and beam divergence angle functions are derived. The source model's axial ion temperature is also changed from 34 eV to 3.4 eV. Oh had originally chosen 34 eV because this value seemed to give better results for the current density distribution. However, experimental as well as simulated results of ion energy distributions point toward a value smaller by an order of magnitude. Further modification to *Quasi3* includes improvement on the surface interaction model by incorporating an empirical model for angular dependence of sputtering yield developed by Yamamura et. al [20]. This model introduces an incident angle-dependent correction factor to the normal sputtering yield calculated in Oh's original surface model.

Asare [1] made further improvements to the surface interaction model. The Yamamura method for angular dependence of sputtering yield is retained. However, calculation of the normal sputtering yield is performed with a formulation by Matsunami, recommended by Yamamura. In addition to tabulating the erosion rates of surfaces, the sputtering yield is also used to insert sputtered particles into the simulation. These particles are tracked throughout the domain and their subsequent deposition on other objects is recorded. Parameters for xenon sputtering aluminum are used to calculate yields, so all objects are effectively composed of aluminum. Aluminum particles are ejected from the surface in-plane at 45° and travel on straight-line trajectories until they either impact another surface or leave the simulation via an exterior boundary. Incident particles striking surfaces are reflected with 80% of their original energy and incident ions are neutralized before reflection.

2.3 Starting Point of this Research

Unfortunately, while trying to port *Quasi3* from a UNIX to a PC system, the version with the later modifications did not run properly. Thus, the original version of the code written by Oh was procured and serves as the starting point of this research.

Chapter 3

Source Model

3.1 Overview

The *Quasi3* source model is of utmost importance in achieving good plume simulation results. The source model encapsulates the state of the plasma exiting the Hall thruster and must be detailed if accurate results of its expansion as a plume are desired. Past work with *Quasi3* has used a SPT-100 source model derived either from experimental data or from results of a computational simulation. For this work, the source model is of the lower power Hall thruster being used for the ETEEV experiment, the BHT-200. The thruster is pictured in Figure 3-1 and its nominal operating parameters are summarized in Table 3.1.

Anode Specific Impulse	1530 s
Thrust	10.5 mN
Discharge Voltage, V_D	300 V
Discharge Current, I_a	0.65 A
Thruster Mass Flow Rate, $\dot{m}_a$	0.70 mg/s
Cathode Mass Flow Rate, $\dot{m}_c$	0.10 mg/s

Table 3.1: BHT-200 nominal operating parameters.

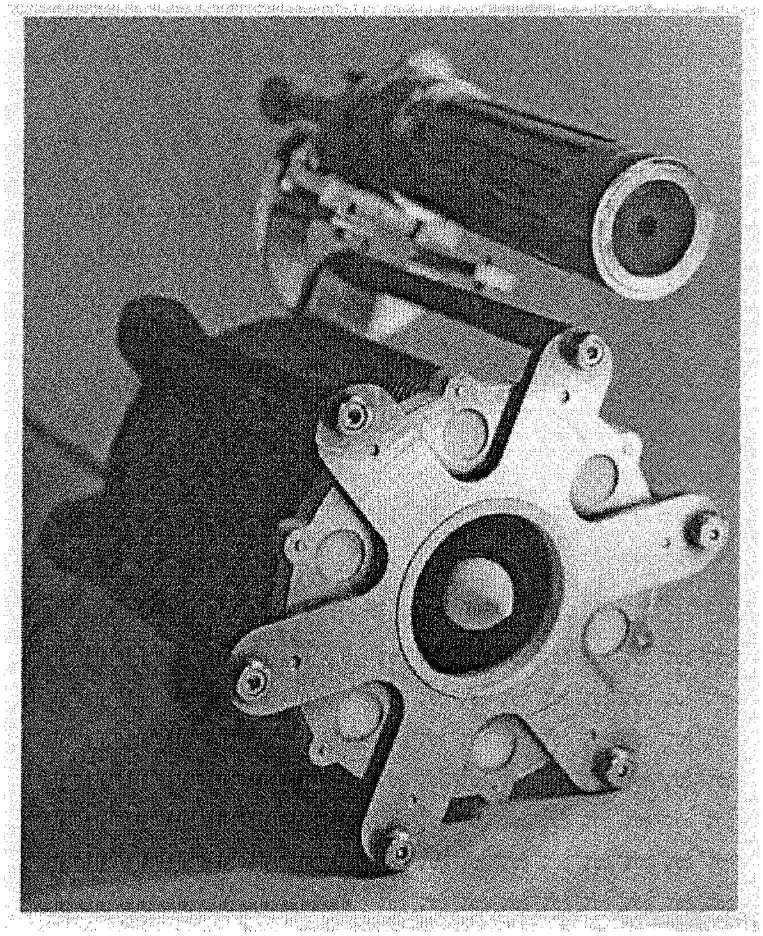

Figure 3-1: BHT-200.

3.2 Preliminary Source Model

A preliminary BHT-200 source model is generated in the same way that Qarnain developed a computational SPT-100 source model. Fife's engine code, *Hall*, had previously been regridded as in Figure 3-2 to simulate the BHT-200 by Szabo [17]. Figure 3-3 shows a schematic of the side view of the BHT-200. The rounded tip can be seen in the curved portion of the bottom of the computational grid which corresponds to the thruster centerline. The left-hand curve of the grid represents a "virtual anode," corresponding to the first magnetic streamline that intersects the anode. Electrons are expected to flow directly down this streamline into the anode since their mobility parallel to the magnetic field is much greater than in the perpendicular direction. Consequently, the thruster channel upstream of this streamline is dominated by neutrals since electrons are not present to cause ionization and is ignored by the simulation. If the grid is rotated about the centerline, the annular BHT-200 is obtained. Runs are performed over several thruster oscillations and time-averaged to provide exit plane information for the new *Quasi3* BHT-200 source model. Simulated thruster performance parameters from *Hall* are shown in Figure 3-4 and Table 3.2. The exit plane location is chosen at $z = .030\ m$ since the space potential as seen in Figure 3-5 has mostly fallen off by the time ions reach this axial position. As *Quasi3* does not directly model ions falling through the potential produced by the thruster, it is important that these effects are already incorporated into the source model.

Thrust, $\langle F \rangle$	$10.0\ mN$
Anode Current, $\langle I_a \rangle$	$0.6821\ A$
Beam Current, $\langle I_b \rangle$	$0.4424\ A$

Table 3.2: Simulated performance of BHT-200 from *Hall*.

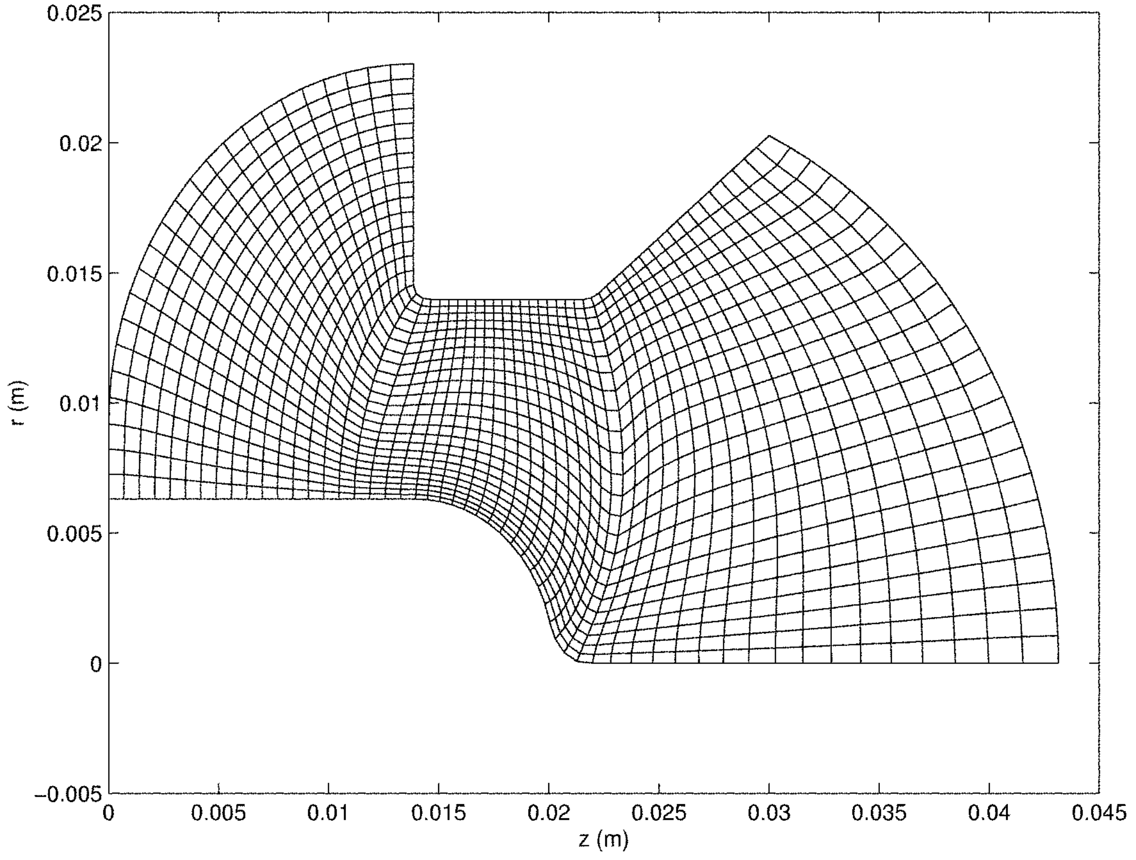

Figure 3-2: Spatial grid for the BHT-200 geometry.

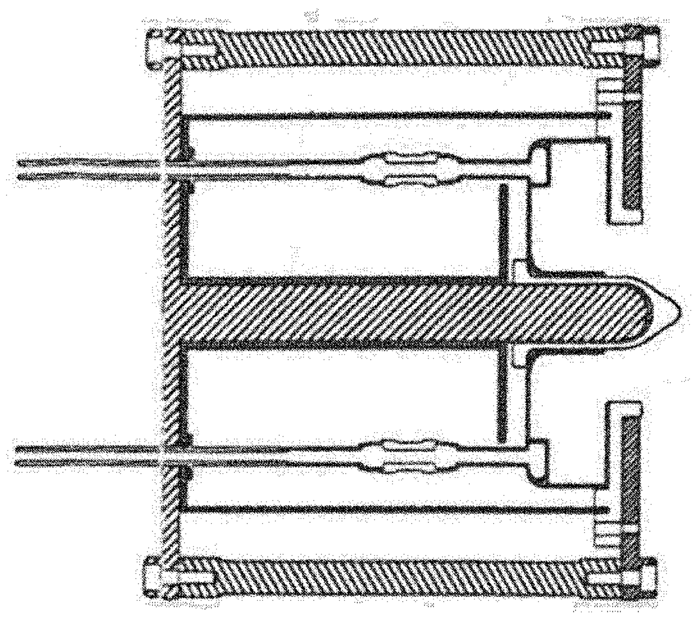

Figure 3-3: Side view of BHT-200.

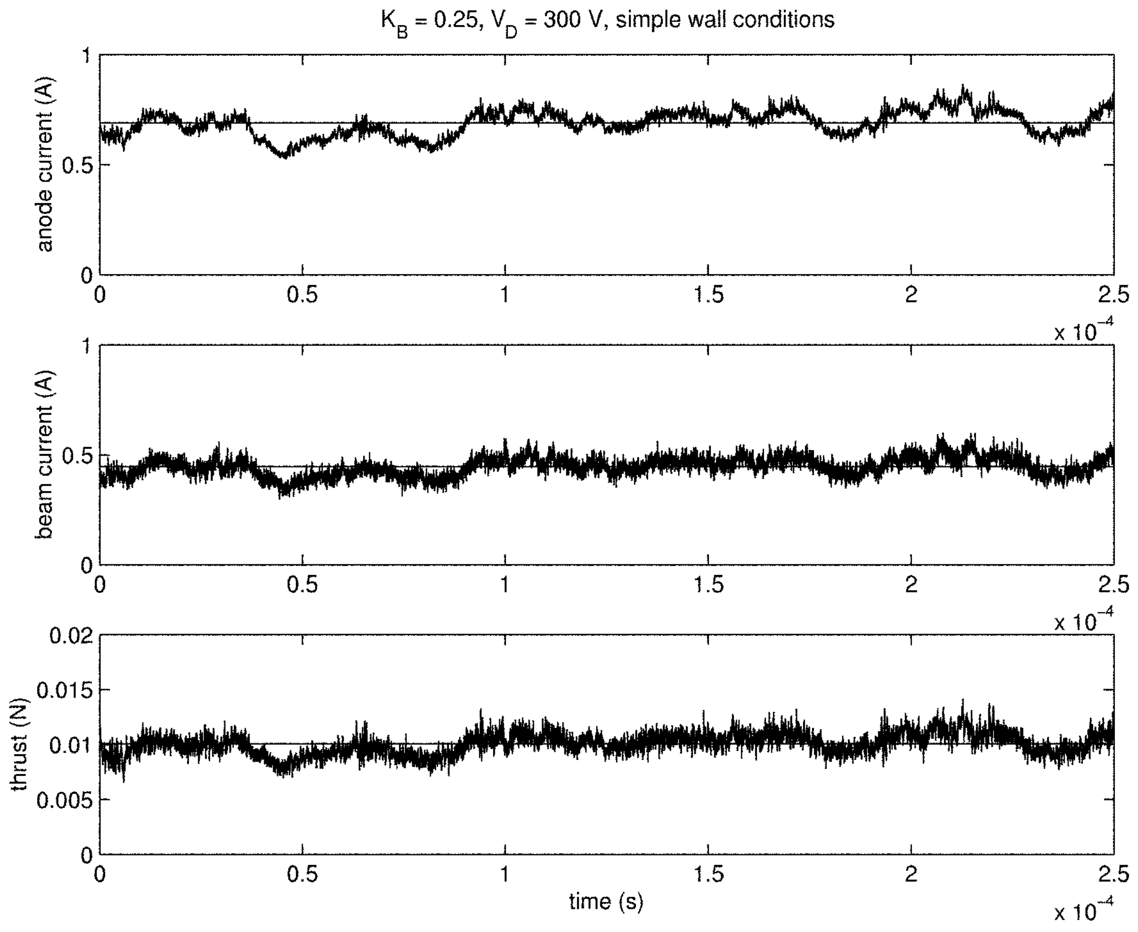

Figure 3-4: Simulated performance of BHT-200 from *Hall*.

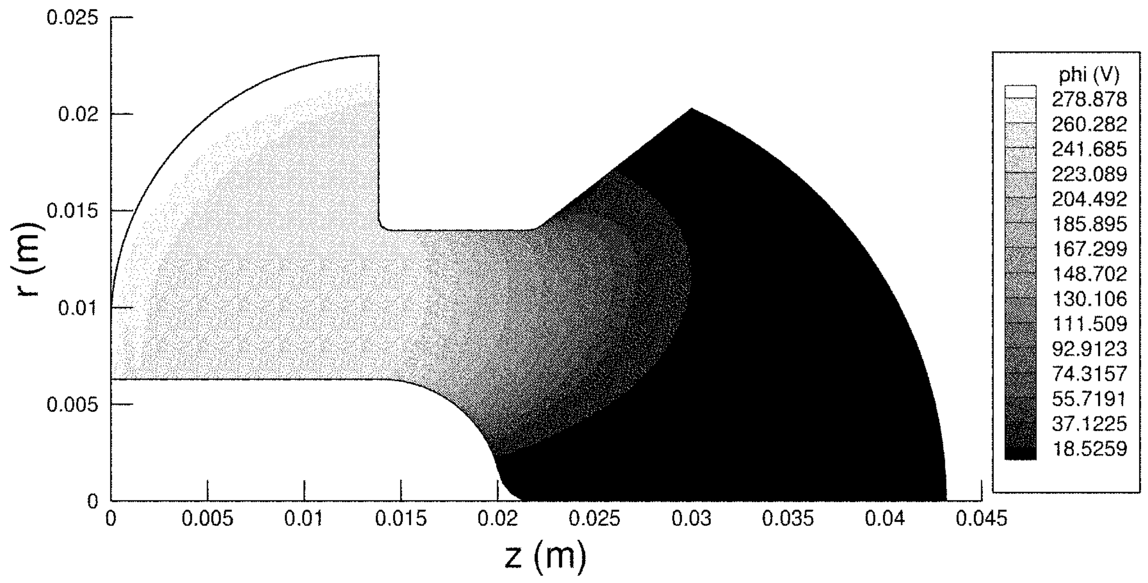

Figure 3-5: Space potential for BHT-200.

3.2.1 Exit Plane Distributions

Figure 3-6 shows the averaged simulated Xe^+ flux. Flux vectors at the grid points closest to the exit plane are extracted to generate source model distributions. The flux along the thruster centerline is not considered in this analysis since the value is unrealistically high as the corresponding area approaches zero. The flux distribution appears to have a $1/r$ relationship, consistent with the convergence of the flow towards the engine's centerline, so a polynomial fit is performed on the $r * flux$ distribution. The resulting flux distribution is divided by the exit ion drift velocity to obtain a number density distribution. A probability density function for radial ion position is obtained by normalizing the number density distribution by its integral over the exit plane. Integrating the probability density function results in,

$$P(r) = 50.1615r + 2049.6r^2 - 1.0315 \times 10^5 r^3, \tag{3.1}$$

a cumulative distribution function that can be used to select the radial position of ions as they are inserted into the simulation. P is the probability that the radial coordinate of the source ion is less than r in meters. Hence, P is 0 at the inner radius of the exit plane and 1 at the outer radius. The direction of the flux vectors is used to generate a mean divergence angle function, resulting in,

$$\begin{aligned}\alpha(r) &= -2.763739 \times 10^9 r^4 + 1.004221 \times 10^8 r^3 - 8.261261 \times 10^5 r^2 \\ &\quad + 4.677020 \times 10^2 r - 1.634443, \end{aligned} \tag{3.2}$$

where α is in degrees. Figure 3-7 shows this analysis.

3.2.2 Additional Parameters

Anode propellant fraction, f_a, is calculated from nominal flow values to the anode and the cathode with,

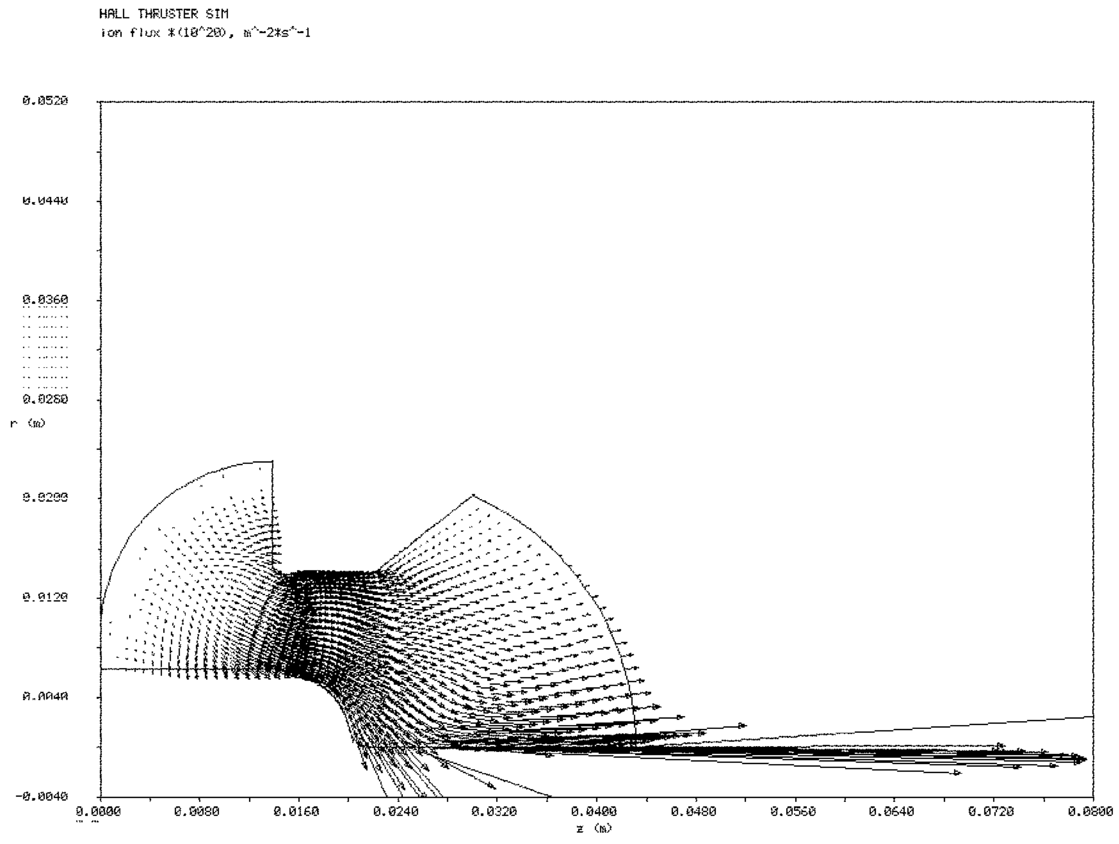

Figure 3-6: Simulated Xe$^+$ flux.

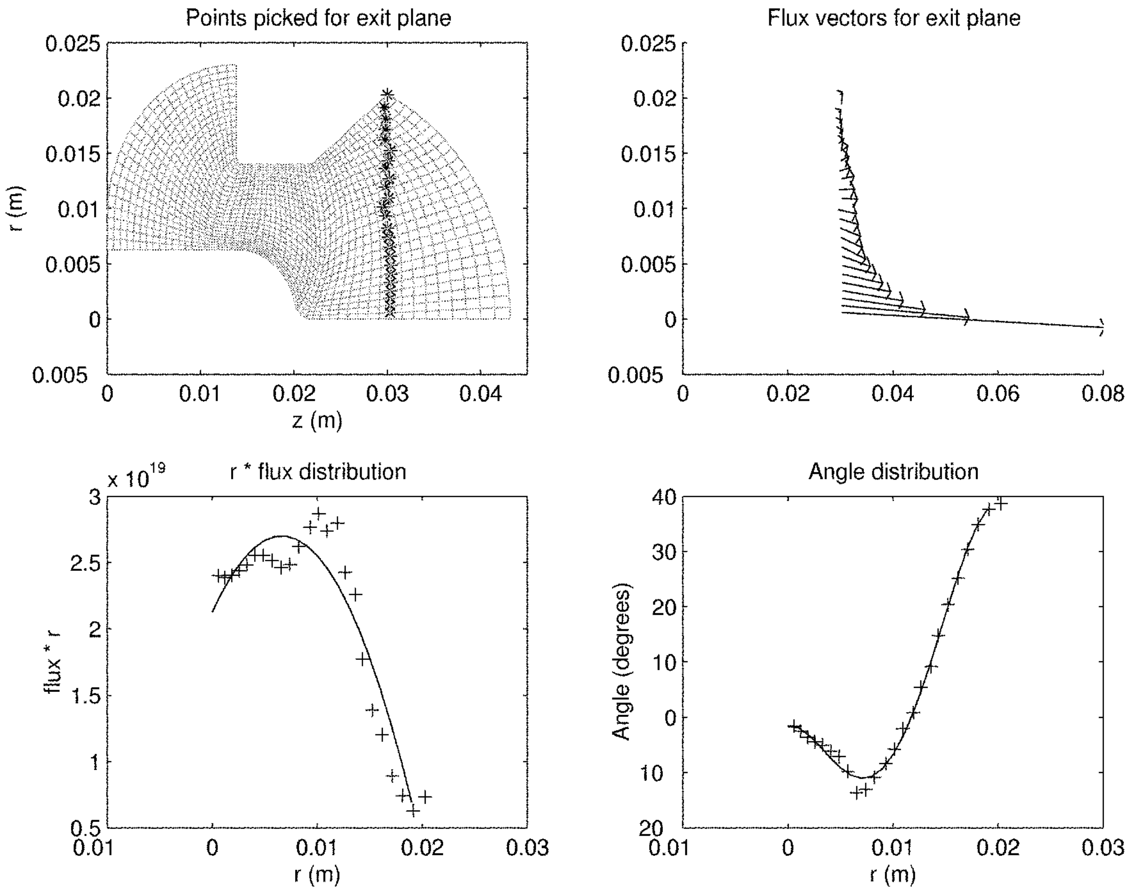

Figure 3-7: Modeling of BHT-200 exit plane.

$$f_a = \frac{\dot{m}_a}{\dot{m}_a + \dot{m}_c}. \tag{3.3}$$

Anode ionization fraction, η_u, is calculated from *Hall* results using,

$$\eta_u = \frac{\dot{m}_i}{\dot{m}_a}, \tag{3.4}$$

where beam current from Table 3.2 is converted to the corresponding ion mass flow, $\dot{m}_i$, assuming only singly-charged species. Axial ion drift velocity is derived from,

$$(v_i)_{RMS} = \sqrt{\frac{2eV_B}{m_i}}, \tag{3.5}$$

where,

$$V_B = \eta_e V_D = \frac{e}{2m_i V_D}\left(\frac{F}{I_B}\right)^2 V_D, \tag{3.6}$$

is calculated from *Hall* results.

By equating angular momentum out and the moment of forces in the azimuthal direction, the experimental value of the original source model is replaced by an expression for azimuthal ion drift velocity derived from,

$$\iint_{A_e} n_e v_{i_x}(m_i v_{i_\theta} r)\, dA = \iiint e(v_{i_x} B_r - v_{i_r} B_x) r\, dV. \tag{3.7}$$

Neglecting the axial magnetic field component gives,

$$\int n_e v_{i_x}(m_i v_{i_\theta} r) 2\pi r\, dr = \int_x \int_r e v_{i_x}(r B_r) r\, dr\, dx, \tag{3.8}$$

which can be written as,

$$m_i v_{i_x} \frac{d(r v_{i_\theta})}{dx} \simeq e v_{i_x} B_r r. \tag{3.9}$$

Finally,

$$\langle v_{i\theta} \rangle_r = \frac{e \langle B_r \rangle_x}{m_i} \cdot x = \langle \omega_i \rangle_{x,r} \cdot x, \tag{3.10}$$

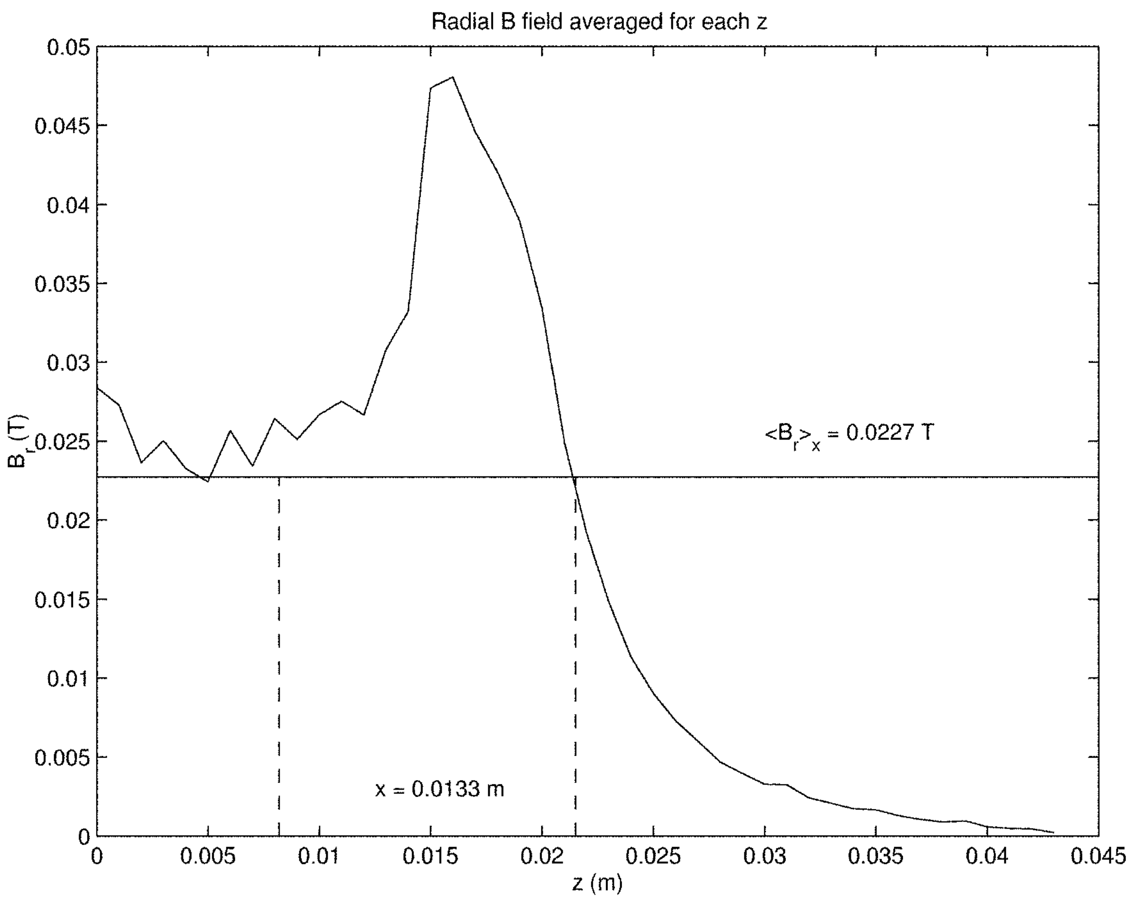

Figure 3-8: Calculation of azimuthal ion drift velocity.

where $\langle \omega_i \rangle_{x,r}$ is the plasma ion cyclotron frequency. The engine's radial magnetic field is averaged at each axial position as shown in Figure 3-8 and x is determined by the distance between the start of ionization (0.0082 m) according to *Hall* and the position corresponding to the overall averaged radial magnetic field (0.0215 m).

Axial and radial ion temperatures are assumed to be 3.5 eV each. The azimuthal ion temperature of .069 eV is kept from the original source model, while neutrals are assumed to be thermalised with the wall at 700 K. Cathode parameters are taken from engineering drawings of the BHT-200.

Because double ions are not directly modeled by *Hall*, they are assumed to have the same exit plane distributions as the single ions. The fraction of ions leaving the engine assumed to be doubly ionized is 0.1. These particles are inserted into the simulation with 1.3 times the single ion drift velocity, corresponding to slightly less

Exit Plane Outer Radius, r_1	0.0203 m
Exit Plane Inner Radius, r_2	0.0000 m
Anode Propellant Fraction, f_a	0.875
Anode Utilization Fraction, η_u	0.8685
Xe^+ Axial Drift Velocity	16,440 m/s
Xe^+ Azimuthal Drift Velocity	221.4 m/s
Xe^+ Axial Temperature	3.5 eV
Xe^+ Radial Temperature	3.5 eV
Xe^+ Azimuthal Temperature	0.068975517 eV
Xe Temperature	0.0603448276 eV
Cathode Orifice Radius	0.0037338 m
Cathode Axial Offset	0.0094 m
Cathode Radial Offset	0.0472 m
Anode Double Ion Fraction	0.1

Table 3.3: Preliminary BHT-200 source model parameters.

than double the energy of a single ion. This value is suggested by computations of Blateau et. al. [2] for another small Hall thruster which conclude that double ion production occurs throughout the acceleration channel in contrast with single ion production which occurs primarily in the near-anode region. Hence, double ions are accelerated through a much wider range of potentials than single ions and exit the thruster with less than twice the energy of the singly-charged species. Table 3.3 summarizes parameters used for the BHT-200 source model.

3.3 Improved Source Model

An improved source model is generated by using output from an upgraded version of Fife's code, *HPHall*, which includes modeling of double ions in the acceleration channel, thus eliminating the assumption that ion species share the same distribution. Using the same grid in Figure 3-2, runs are performed over several thruster oscillations and time-averaged. Figure 3-9 and Table 3.4 show the results for thruster performance parameters.

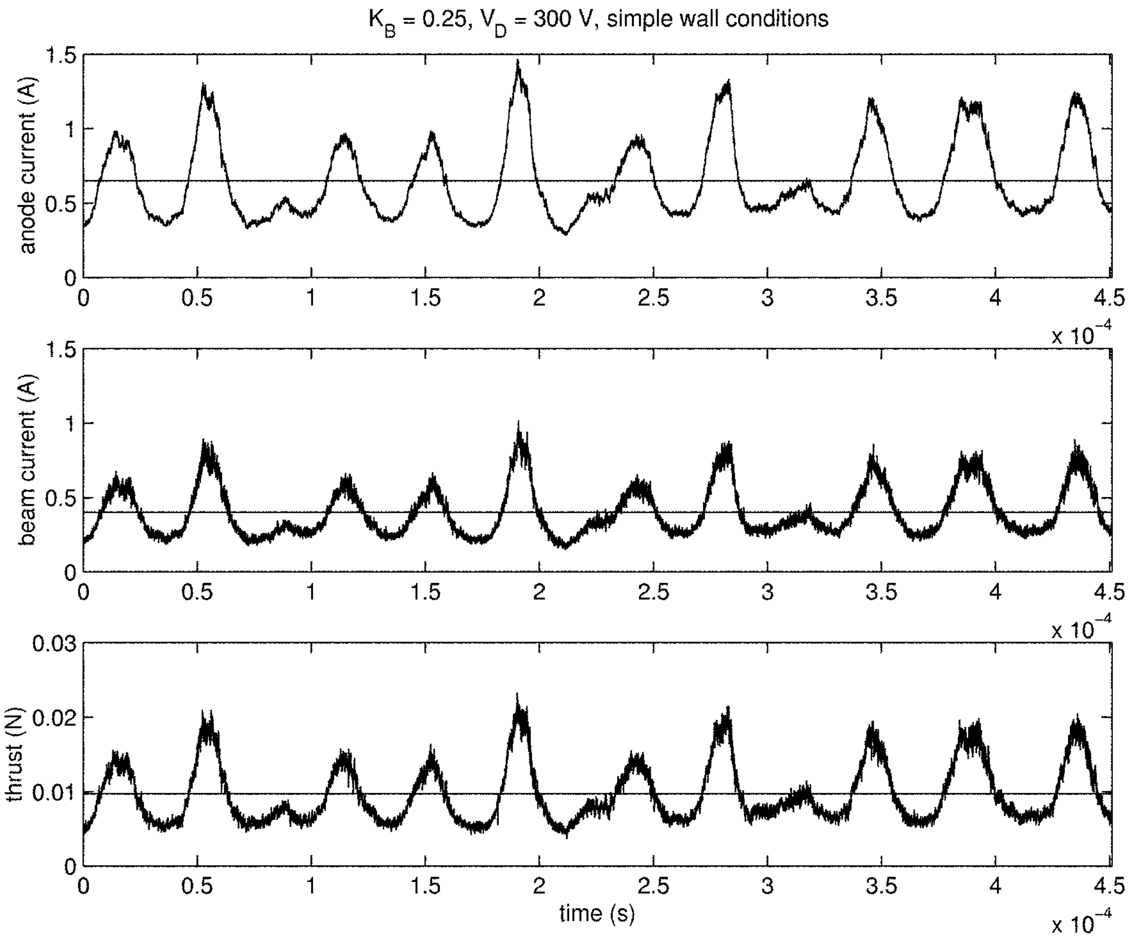

Figure 3-9: Simulated performance of BHT-200 from *HPHall*.

Thrust, $\langle F \rangle$	$9.7\ mN$
Anode Current, $\langle I_a \rangle$	$0.6494\ A$
Beam Current, $\langle I_b \rangle$	$0.4022\ A$

Table 3.4: Simulated performance of BHT-200 from *HPHall*.

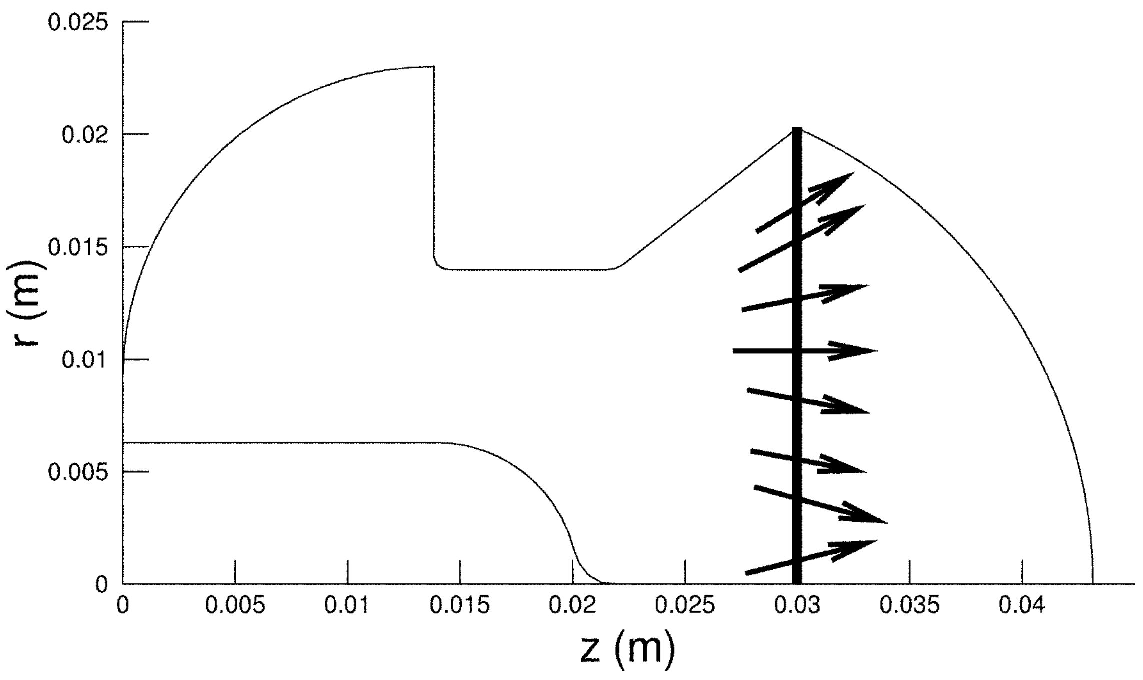

Figure 3-10: Exit plane for improved source model.

3.3.1 Exit Plane Distributions

Instead of reading averaged information off of grid points, *HPHall* is modified to record data about particles crossing the exit plane at $z = .030\ m$ as seen in Figure 3-10. This plane is divided into 25 radial bins which in turn are sub-divided into axial and radial velocity bins. Single and double ion data are recorded separately so that distinct distributions for each species can be derived.

Since detailed information about axial and radial velocities is tracked, the new source model does not assume a constant ion drift velocity across the exit plane. Instead, functions for both velocity components, depending on radial position, are extracted from the *HPHall* simulation data. This approach also eliminates the need for a beam divergence angle function as the direction of the particle is contained in the two velocities.

Figures 3-11 and 3-12 show the axial and radial velocity distributions of single ions at each radial bin. The vertical line in each plot marks the average velocity for that radius. Two distinct peaks are observed in the radial velocity distributions close to the thruster centerline. This phenomenon is attributed to the fact that as particles in

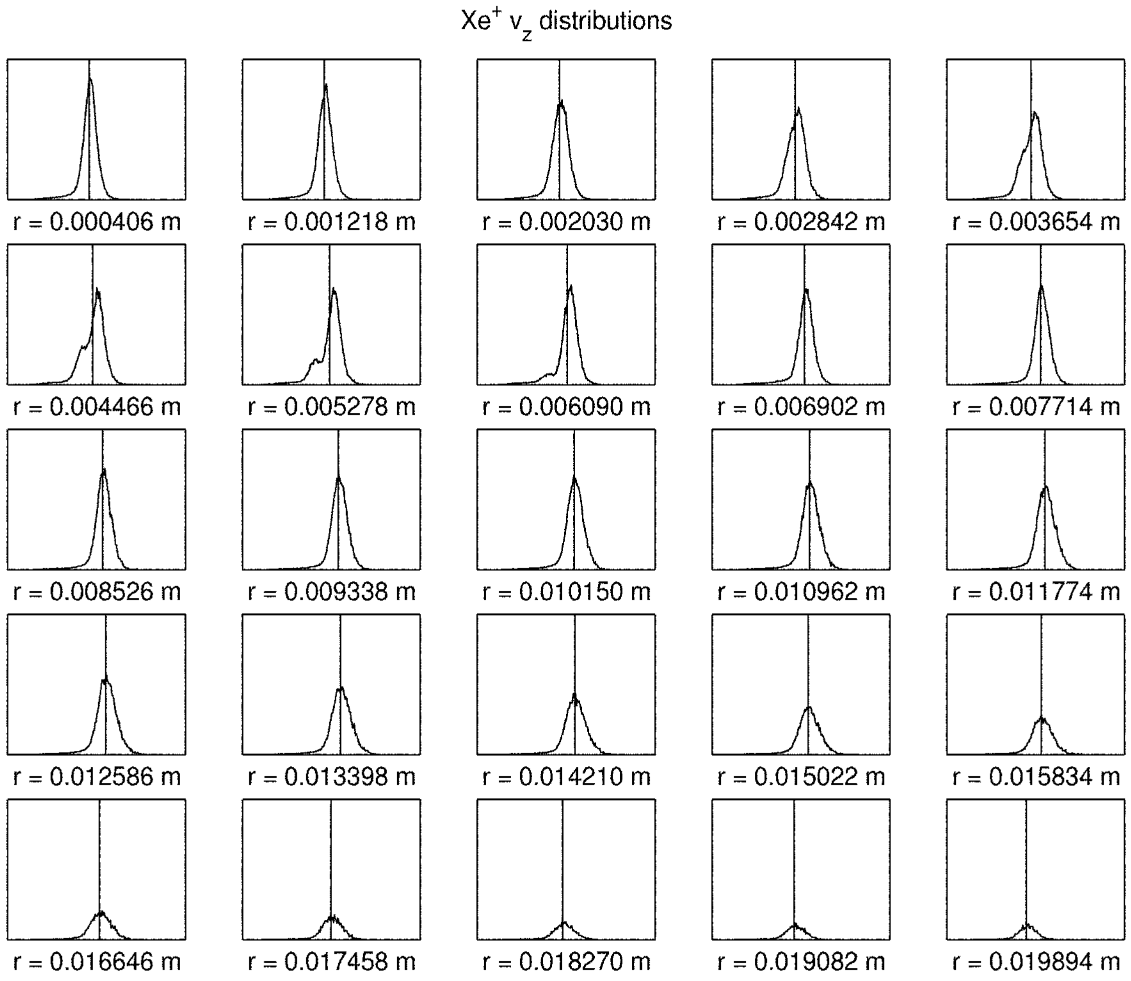

Figure 3-11: Xenon single ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of Xe^+ ranging from 0 to 3×10^{12}.

HPHall cross the centerline grid boundary, they are reflected by reversing the sign of their radial velocity in order to represent particles that are coming from the opposite side of the annular engine. It is clear that the calculated average velocity values for the radial component do not accurately reflect these two distinct populations of ions. Thus, *HPHall* is also modified to label particles reflected at the centerline differently from particles originating in the main engine channel so that the two populations can be investigated independently.

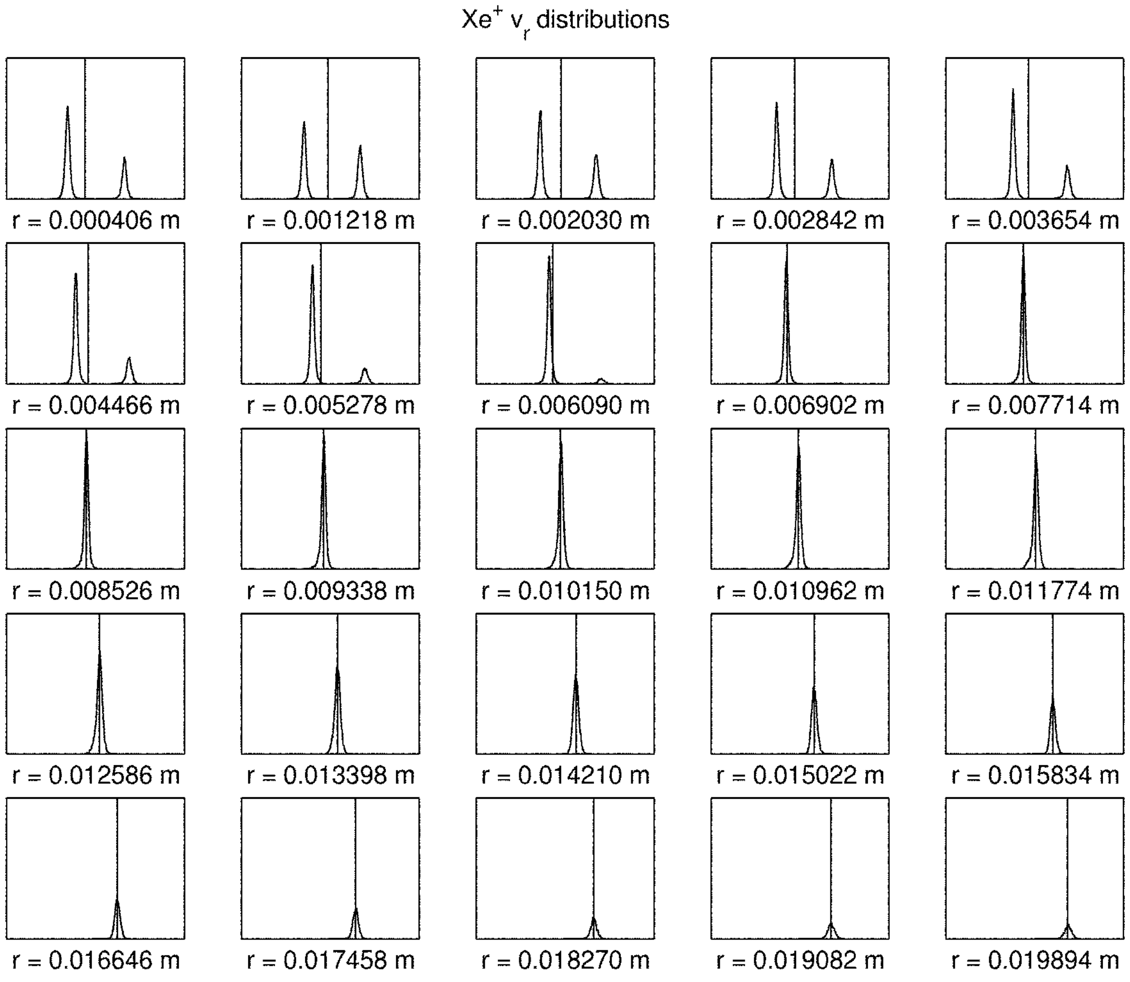

Figure 3-12: Xenon single ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of Xe$^+$ ranging from 0 to 4×10^{12}.

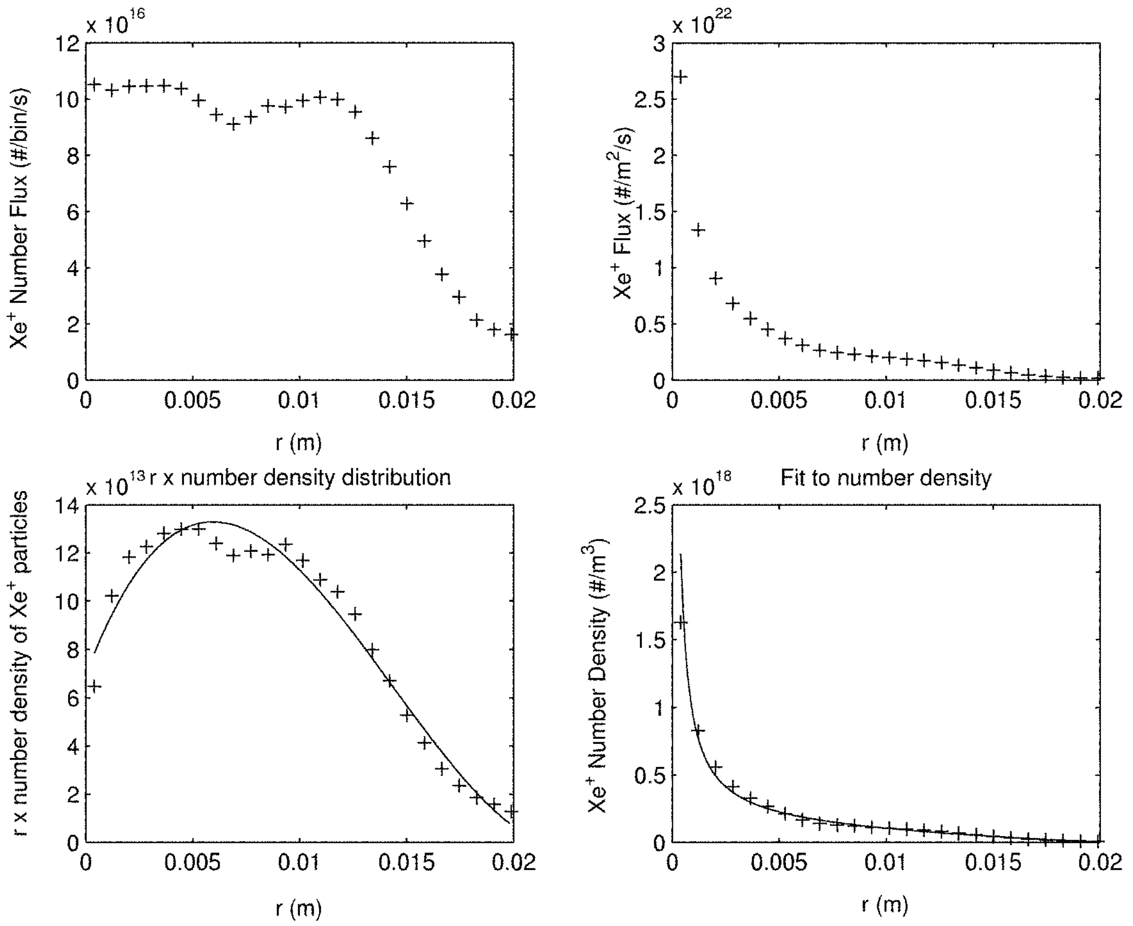

Figure 3-13: Derivation of Xe$^+$ number density distribution.

Single Ions

The same process described for the preliminary source model is used to generate the radial cumulative distribution function, except during conversion of the flux distribution to a number density distribution. Instead of dividing by an assumed ion drift velocity across the entire exit plane, flux at each radial position is divided by its corresponding average velocity. Figure 3-13 shows the procedure to obtain the radial cumulative distribution function given by,

$$P_{Xe^+} = 74.5992r - 1244.3r^2 + 6.4753 \times 10^4 r^3 - 3.1991 \times 10^6 r^4. \tag{3.11}$$

As before, this CDF can be used to determine the radial position of a source ion.

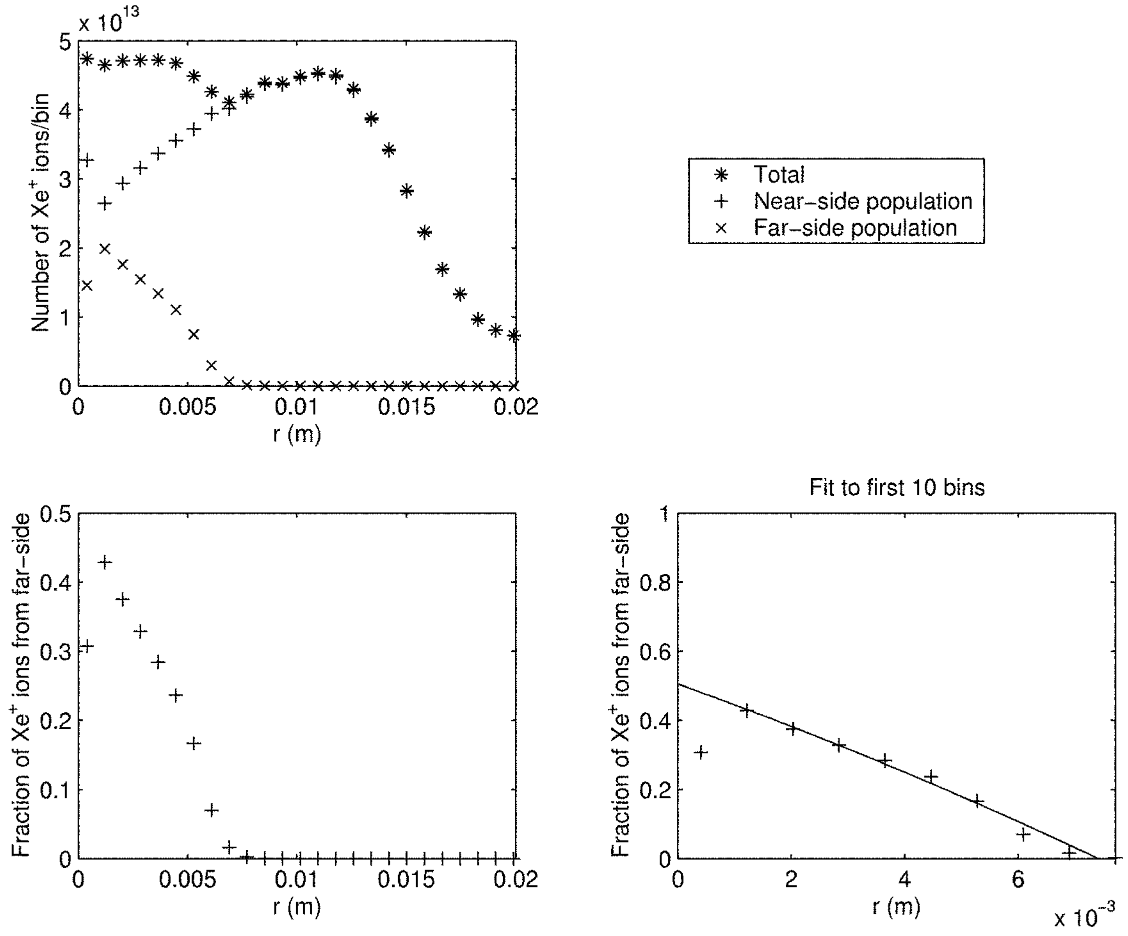

Figure 3-14: Split of single ion populations.

However, for values below $r = 0.007714\ m$, the ion may either be part of the aforementioned near-side or far-side populations. The reason for this sharp cutoff in far-side ions will be seen later. Figure 3-14 shows the split of the two ion populations and the fraction of ions coming from across the centerline as a function of radius. The first 10 radial bins, excluding the first data point, are used to find a fit for this fraction which is given by,

$$f_{far-sideXe^+} = \frac{n_{far-sideXe^+}}{n_{near-sideXe^+} + n_{far-sideXe^+}} = -1.2789 \times 10^3 r^2 - 58.7267r + 0.5055. \tag{3.12}$$

Each population of ions has its own axial and radial velocity functions. Figure 3-15 depicts the velocity distributions for the near-side population where the axial and

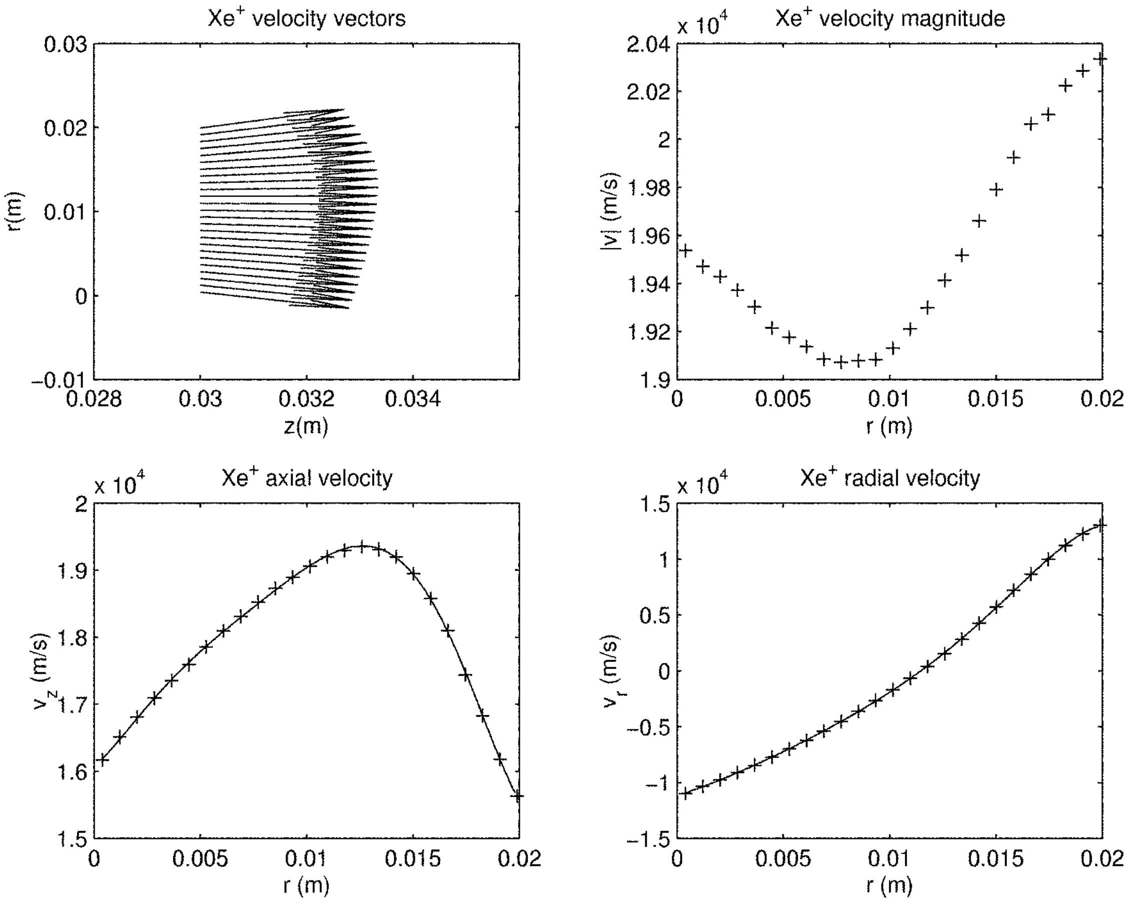

Figure 3-15: Single ion near-side population velocity distributions.

radial velocity functions are given by,

$$\begin{aligned} v_z(r)_{near-sideXe^+} &= 2.4597 \times 10^{15} r^6 - 1.2956 \times 10^{14} r^5 + 2.4102 \times 10^{12} r^4 \\ &\quad - 2.0171 \times 10^{10} r^3 + 6.6624 \times 10^7 r^2 + 2.8743 \times 10^5 r \\ &\quad + 1.6065 \times 10^4, \end{aligned} \tag{3.13}$$

$$\begin{aligned} v_r(r)_{near-sideXe^+} &= -3.2370 \times 10^{13} r^5 + 1.3736 \times 10^{12} r^4 - 1.9589 \times 10^{10} r^3 \\ &\quad + 1.4132 \times 10^8 r^2 + 4.0677 \times 10^5 r - 1.1071 \times 10^4. \end{aligned} \tag{3.14}$$

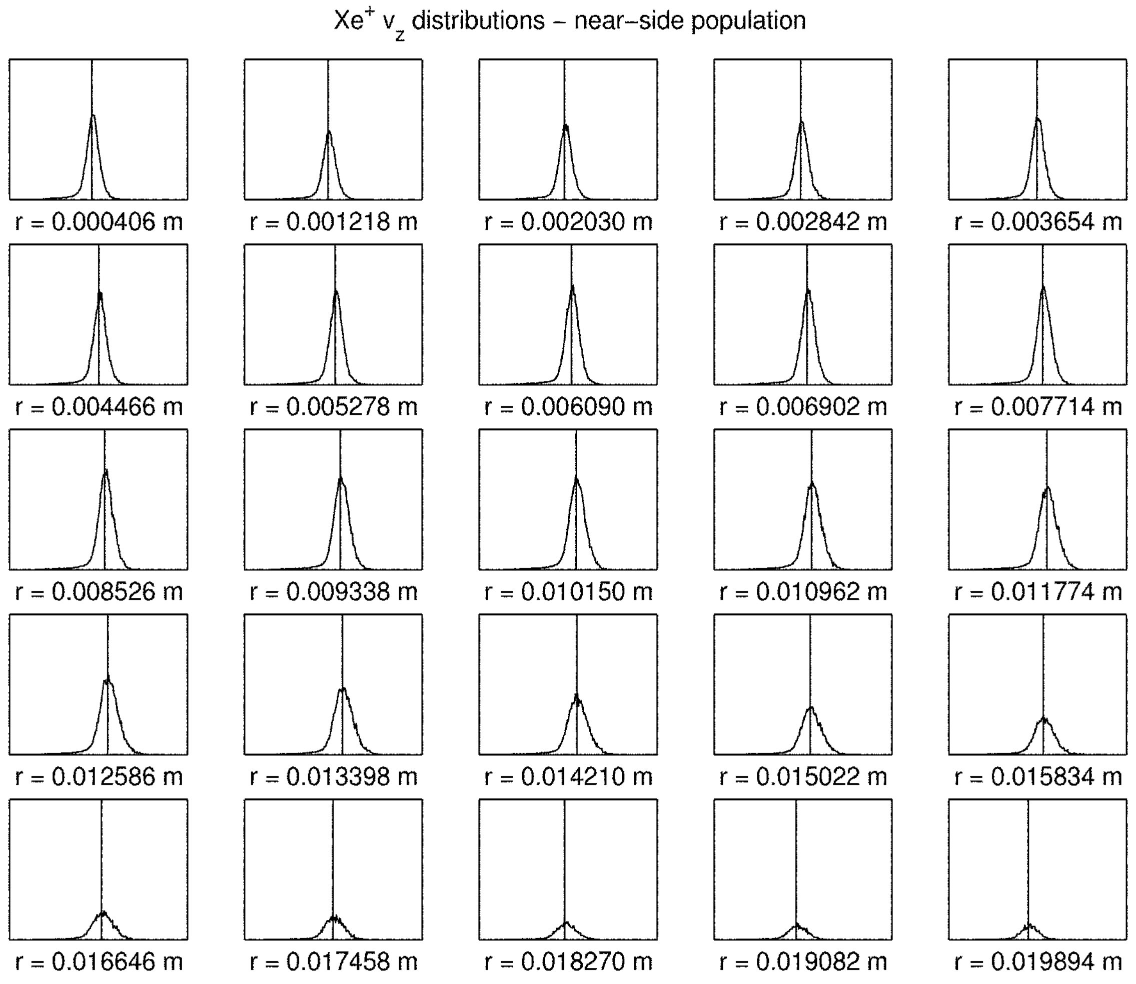

Figure 3-16: Near-side population single ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of near-side Xe$^+$ ranging from 0 to 3×10^{12}.

Figures 3-16 and 3-17 show the axial and radial velocity distributions at each radial position for the near-side population ions. The axial velocity distributions have a noticeable spread, so Gaussian fits are done at each radius to determine an average axial ion temperature which is incorporated into the model by adding a thermal component to the axial velocity. Figure 3-18 shows the temperatures calculated across the exit plane. The axial ion temperature is taken to be 2.96 eV. The radial velocity distributions are narrow enough that no radial ion temperature is assumed.

The same procedure is followed to model the far-side population. Figure 3-19 shows the far-side velocity distributions. Fits for axial and radial velocity are per-

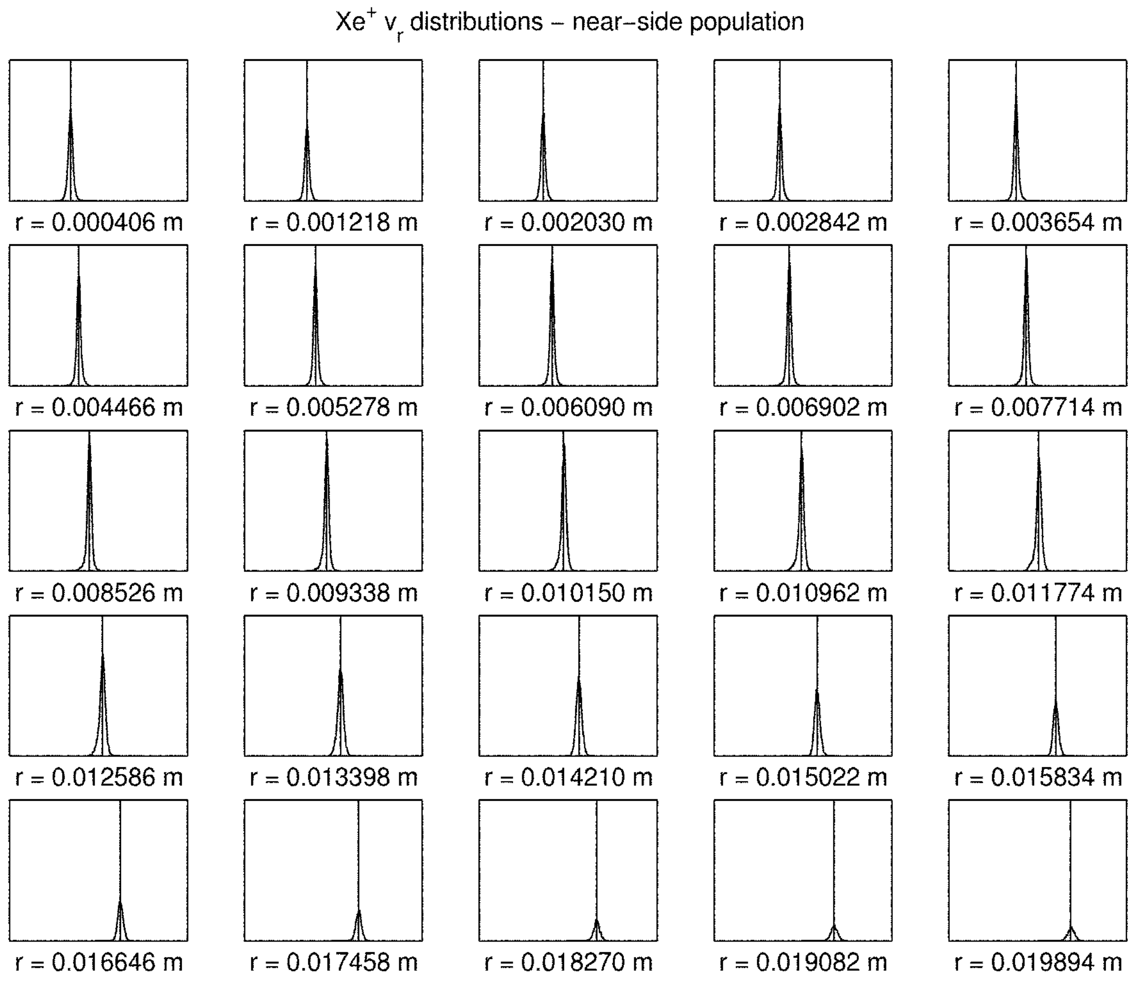

Figure 3-17: Near-side population single ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of near-side Xe$^+$ ranging from 0 to 4×10^{12}.

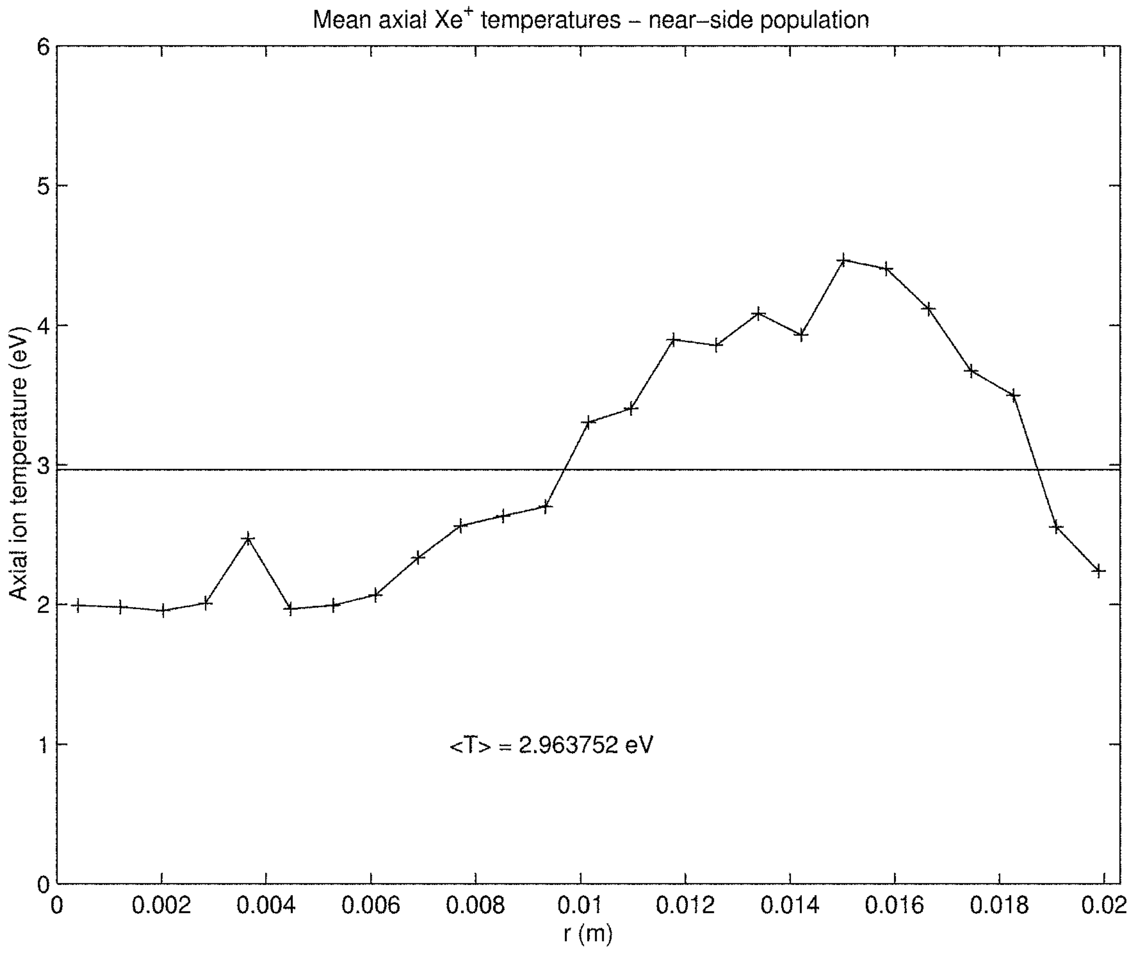

Figure 3-18: Axial near-side ion temperature.

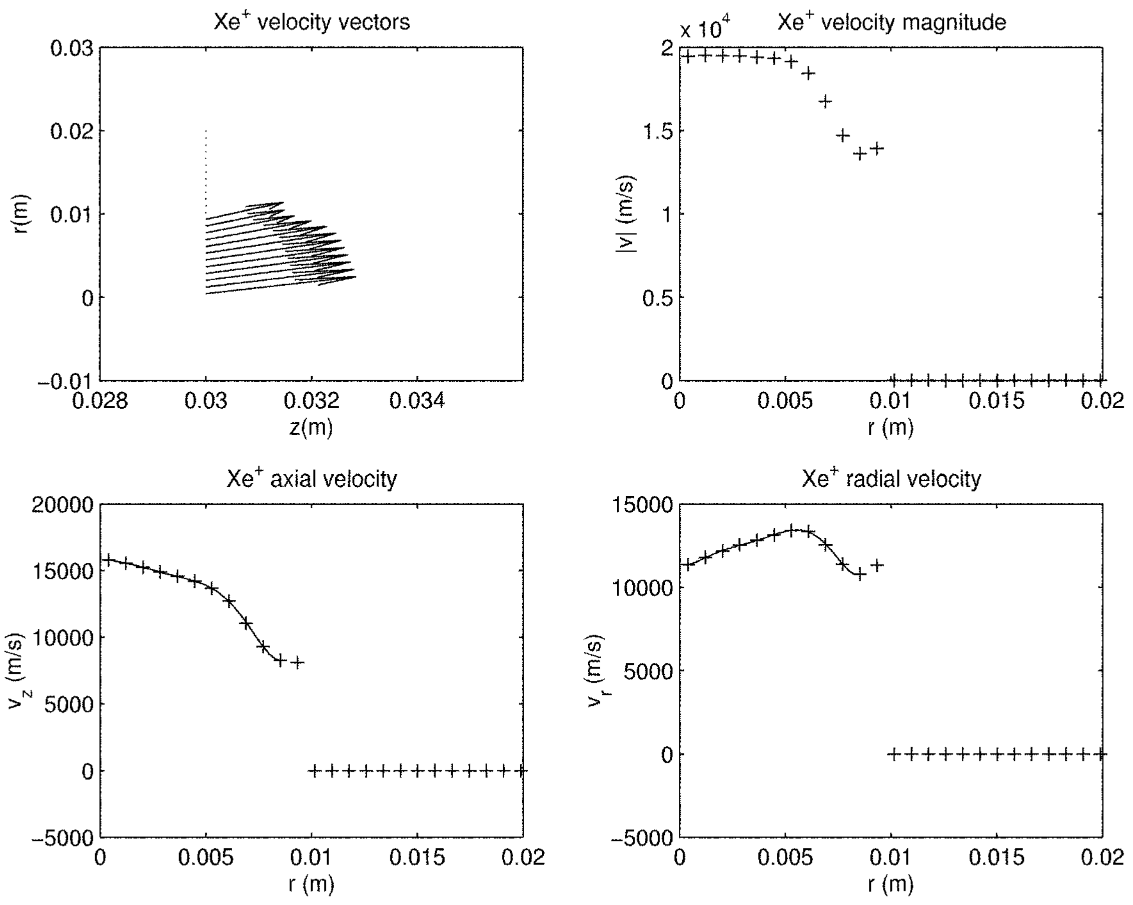

Figure 3-19: Single ion far-side population velocity distributions.

formed on the first 11 bins (the 12th is discarded) to get,

$$\begin{aligned} v_z(r)_{far-sideXe^+} &= 5.2001 \times 10^{17} r^6 - 1.0830 \times 10^{16} r^5 + 7.8699 \times 10^{13} r^4 \\ &\quad -2.4931 \times 10^{11} r^3 + 3.1718 \times 10^{8} r^2 - 4.5502 \times 10^{5} r \\ &\quad + 1.5936 \times 10^{4}, \end{aligned} \tag{3.15}$$

$$\begin{aligned} v_r(r)_{far-sideXe^+} &= 8.2936 \times 10^{17} r^6 - 1.9545 \times 10^{16} r^5 + 1.6987 \times 10^{14} r^4 \\ &\quad - 6.8781 \times 10^{11} r^3 + 1.3017 \times 10^{9} r^2 - 5.3546 \times 10^{5} r \\ &\quad + 1.1409 \times 10^{4}. \end{aligned} \tag{3.16}$$

Figures 3-20 and 3-21 are the velocity distributions for the far-side population at each radial position. The first 8 radial bins are used to calculate the axial ion temperature of the far-side population as shown in Figure 3-22 — the resulting axial far-side ion temperature is 3.47 eV.

As seen in Figure 3-19, a maximum in far-side ion radial velocity occurs, corresponding roughly to the location of the the thruster's curved lip. Far-side ions found radially further from the centerline originate from further back in the accleration channel. Those particles coming from above the lip with too much radial velocity will impact the thruster wall and not make it to the centerline, while ions with too little radial velocity will not make it across the centerline before the exit plane position. Thus, a cut-off in far-side ions at $r = 0.007714\ m$ is observed.

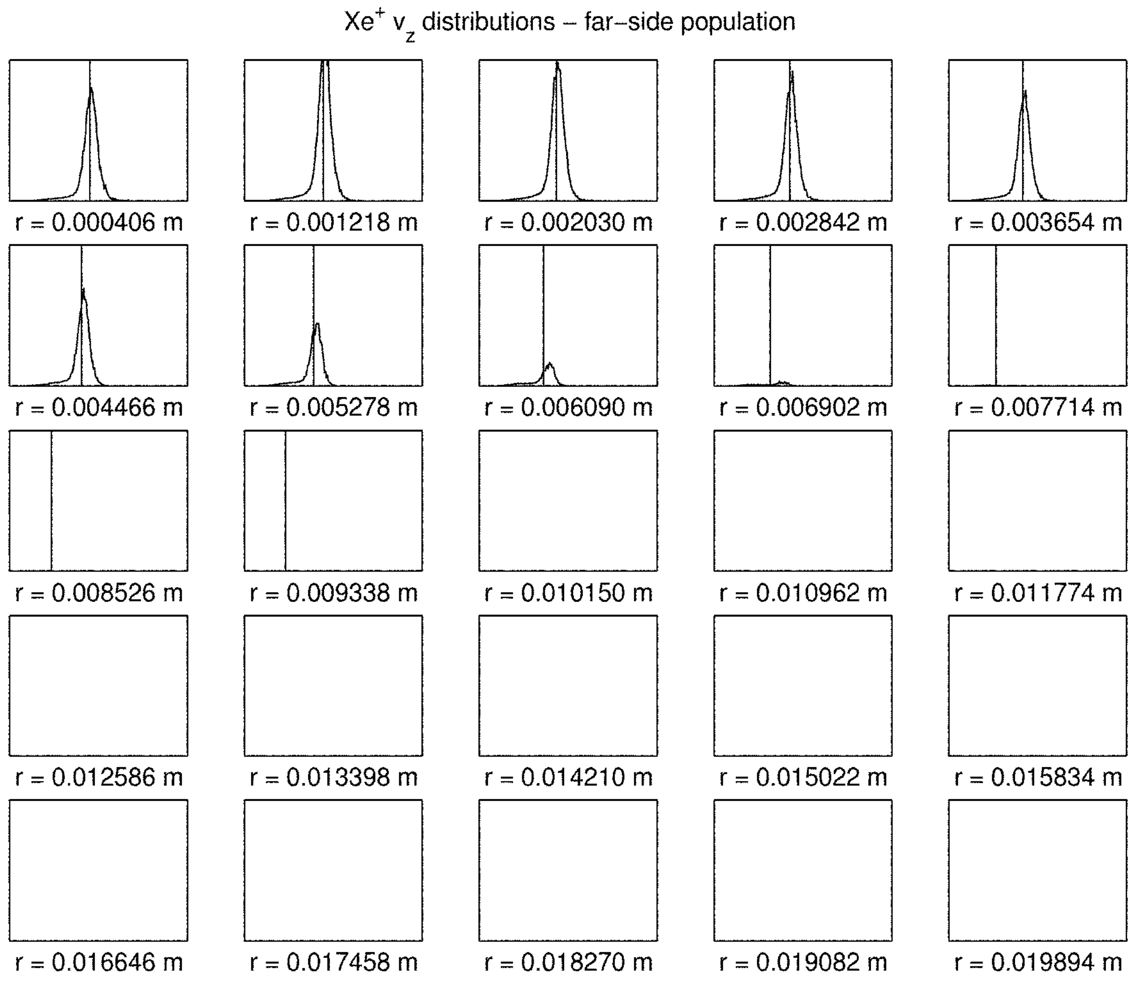

Figure 3-20: Far-side population single ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of far-side Xe$^+$ ranging from 0 to 1×10^{12}.

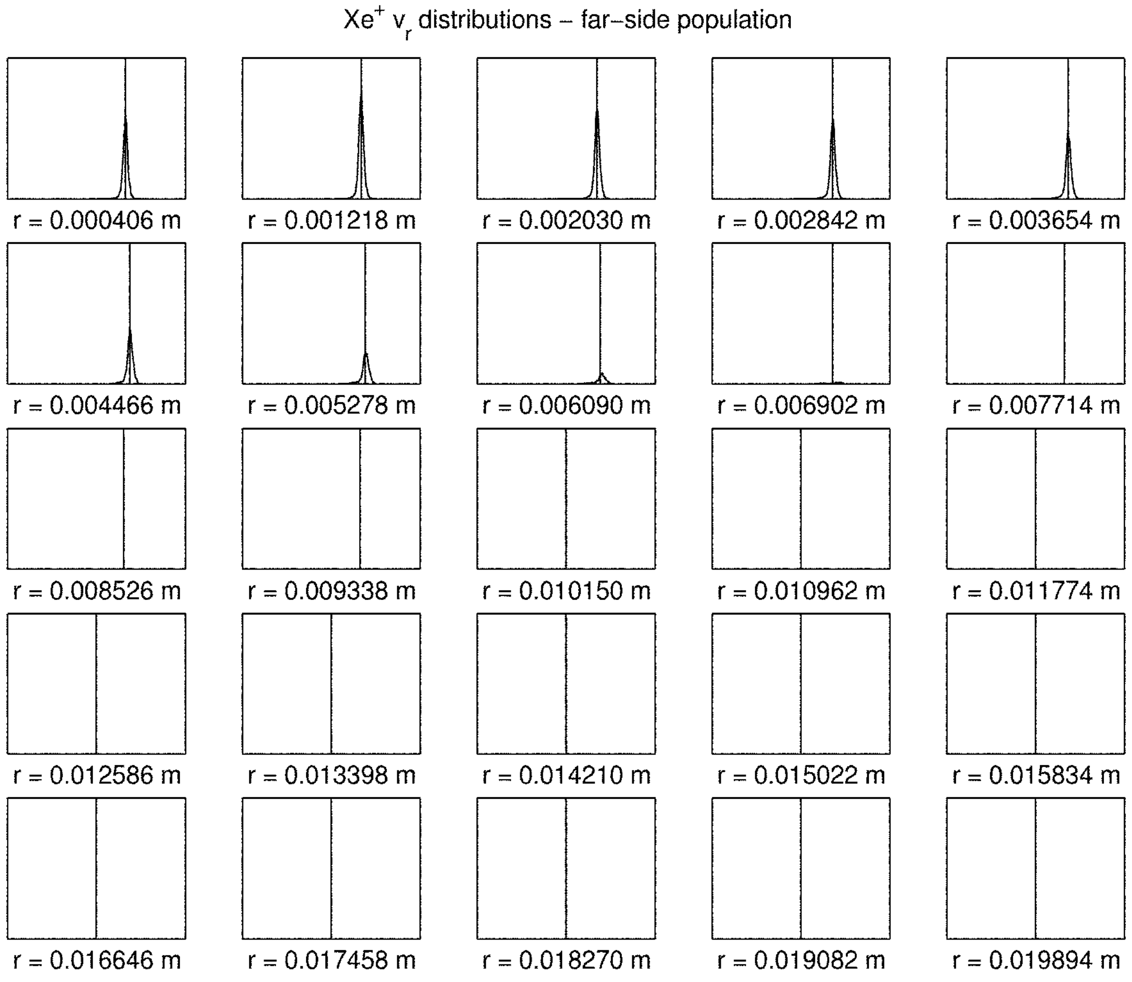

Figure 3-21: Far-side population single ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of far-side Xe^+ ranging from 0 to 2×10^{12}.

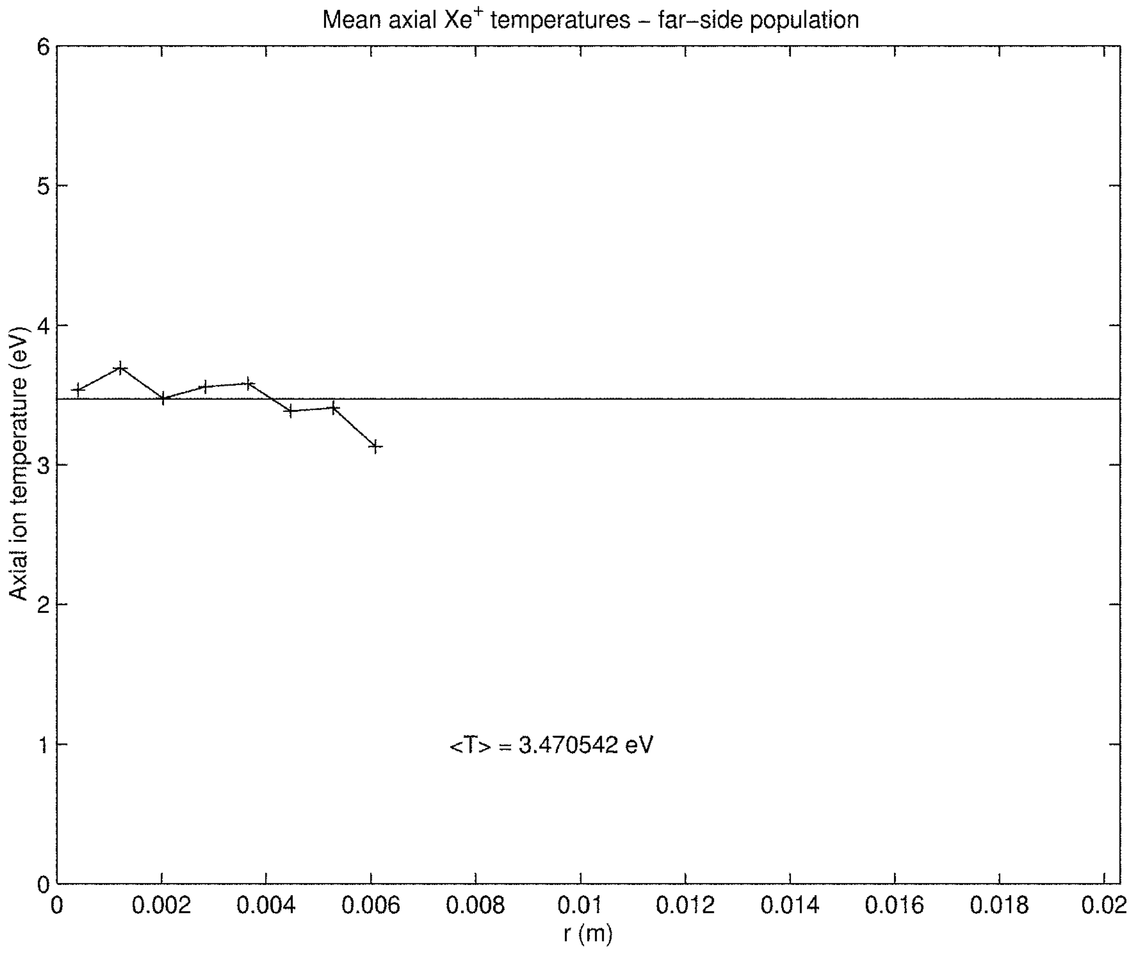

Figure 3-22: Axial far-side ion temperature.

Double ions

Since information about double ions is collected separately from that of the single ions, the doubly-charged species has its own set of exit plane distributions. Figures 3-23 through 3-32 and Equations 3.17 through 3.22 show the analysis for the double ions. Items of note are that the velocities of the double ions are indeed higher than those of the single ions. The extra charge causes the double ions to gain more energy when falling through the thruster's electric field. The double ions also have a larger spread in velocities, corresponding to the higher axial ion temperatures of 7.29 eV for the near-side population and 3.46 eV for the far-side population. This is believed to be a result of double ions being formed at different locations of the acceleration channel and falling through a range of potentials.

$$P_{Xe^{++}} = 70.1681r + 681.9905r^2 - 8.4330 \times 10^4 r^3 \tag{3.17}$$

$$f_{far-sideXe^{++}} = \frac{n_{far-sideXe^{++}}}{n_{near-sideXe^{++}} + n_{far-sideXe^{++}}} = 620.7026r^2 - 66.1980r + 0.4772 \tag{3.18}$$

$$\begin{aligned} v_z(r)_{near-sideXe^{++}} &= 1.6049 \times 10^{13} r^5 - 8.1792 \times 10^{11} r^4 + 1.2617 \times 10^{10} r^3 \\ &\quad - 1.0211 \times 10^8 r^2 + 8.6508 \times 10^5 r + 2.1926 \times 10^4 \end{aligned} \tag{3.19}$$

$$\begin{aligned} v_r(r)_{near-sideXe^{++}} &= -3.8977 \times 10^{13} r^5 + 1.5517 \times 10^{12} r^4 - 2.0763 \times 10^{10} r^3 \\ &\quad + 1.5344 \times 10^8 r^2 + 7.0821 \times 10^5 r - 1.5960 \times 10^4 \end{aligned} \tag{3.20}$$

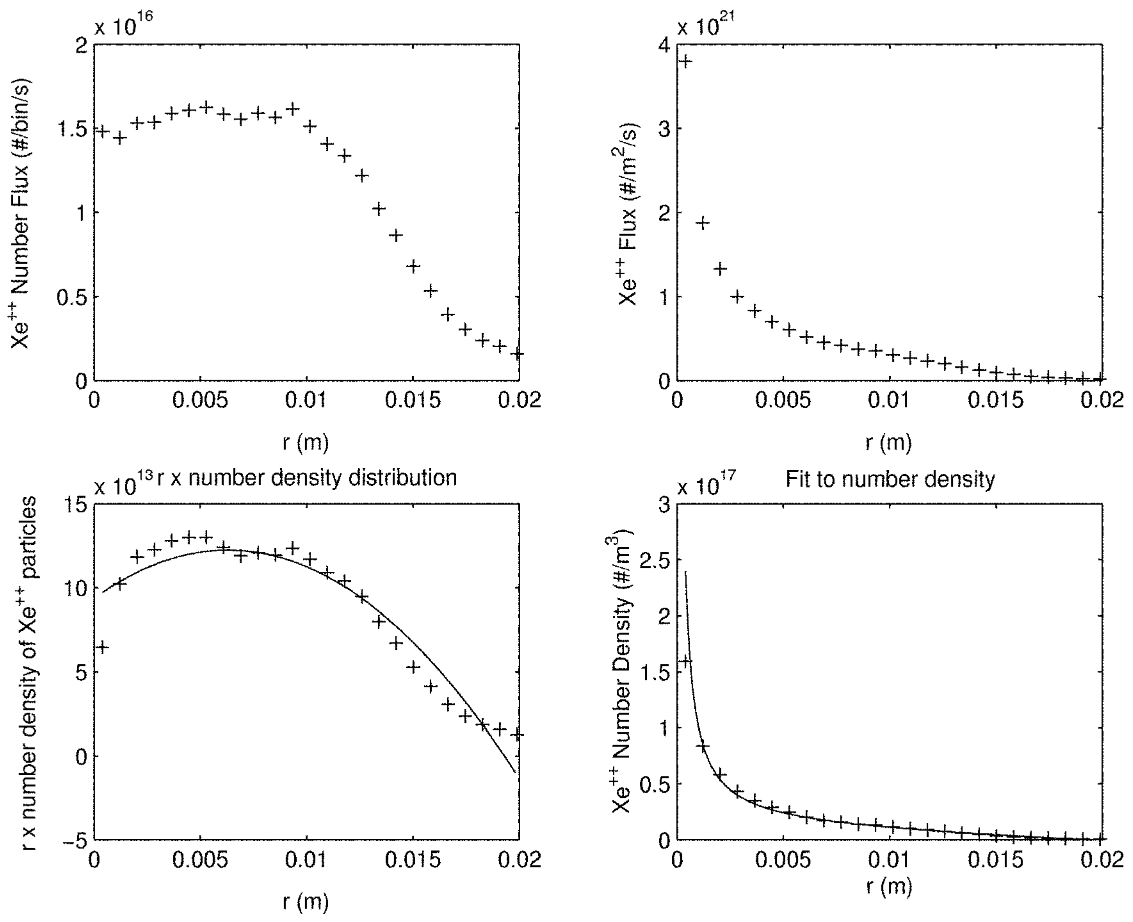

Figure 3-23: Derivation of Xe^{++} number density distribution.

$$\begin{aligned} v_z(r)_{far-sideXe^{++}} &= -3.9674 \times 10^{12} r^4 + 5.2944 \times 10^{10} r^3 - 2.8130 \times 10^8 r^2 \\ &\quad + 3.7329 \times 10^4 r + 2.1585 \times 10^4 \end{aligned} \tag{3.21}$$

$$\begin{aligned} v_r(r)_{far-sideXe^{++}} &= -3.8926 \times 10^{20} r^7 + 1.0983 \times 10^{19} r^6 - 1.2323 \times 10^{17} r^5 \\ &\quad + 7.0001 \times 10^{14} r^4 - 2.1190 \times 10^{12} r^3 + 3.2412 \times 10^9 r^2 \\ &\quad - 1.4654 \times 10^6 r + 1.6299 \times 10^4 \end{aligned} \tag{3.22}$$

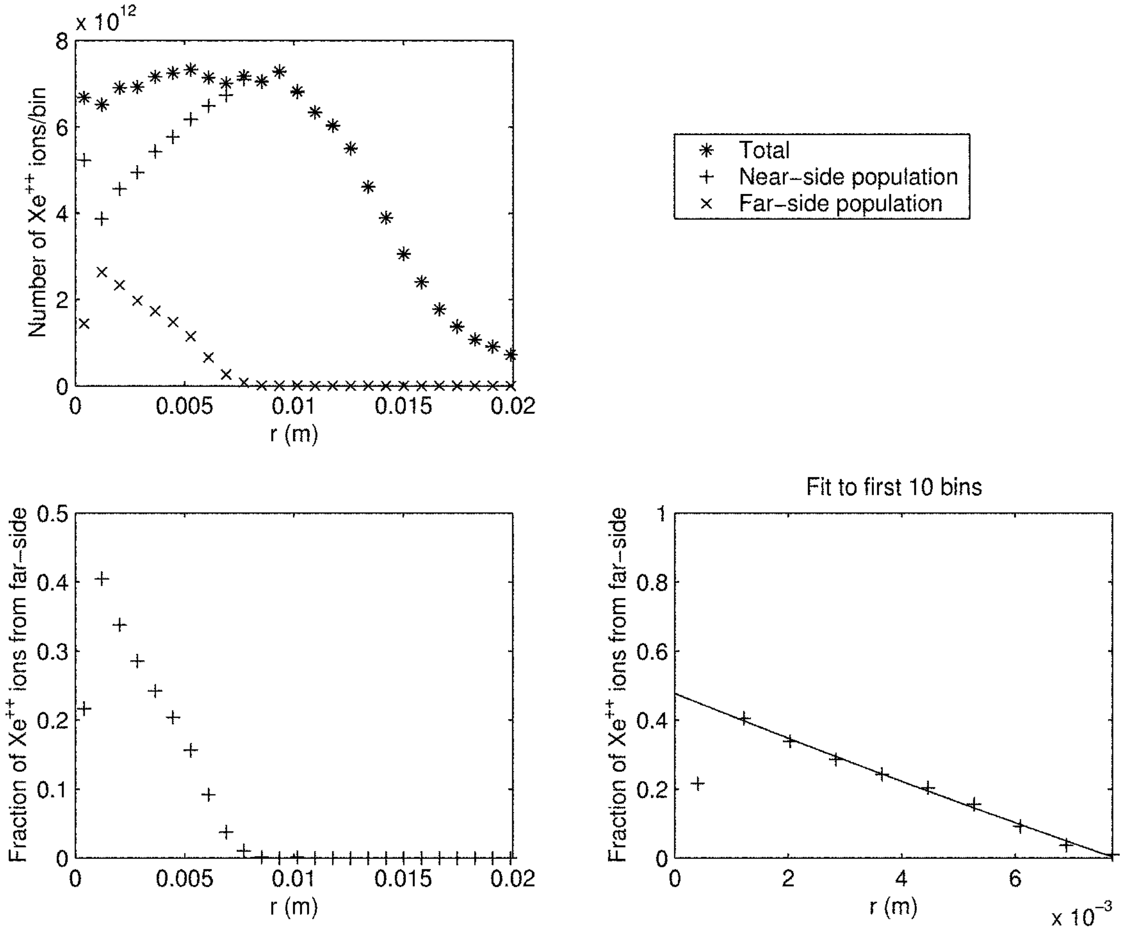

Figure 3-24: Split of double ion populations.

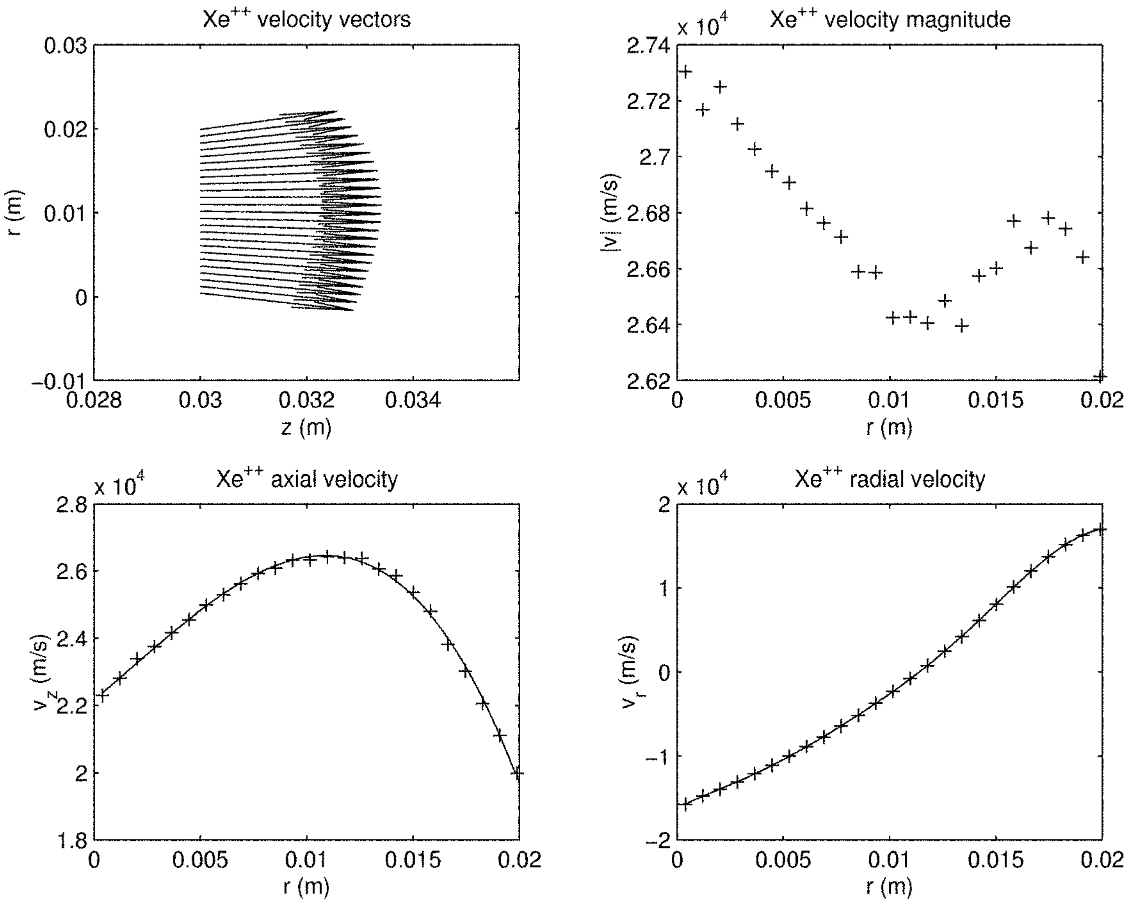

Figure 3-25: Double ion near-side population velocity distributions.

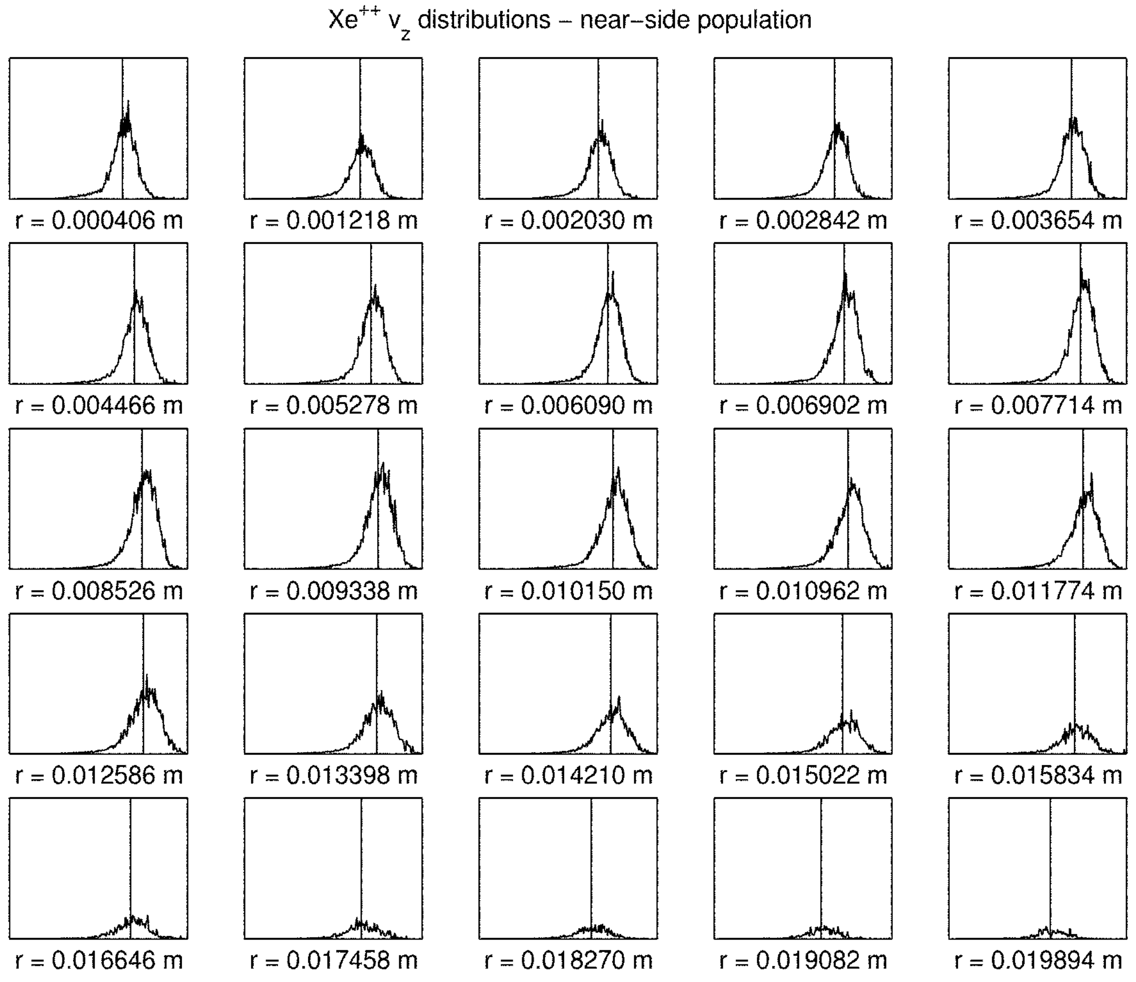

Figure 3-26: Near-side population double ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of near-side Xe^{++} ranging from 0 to 3×10^{11}.

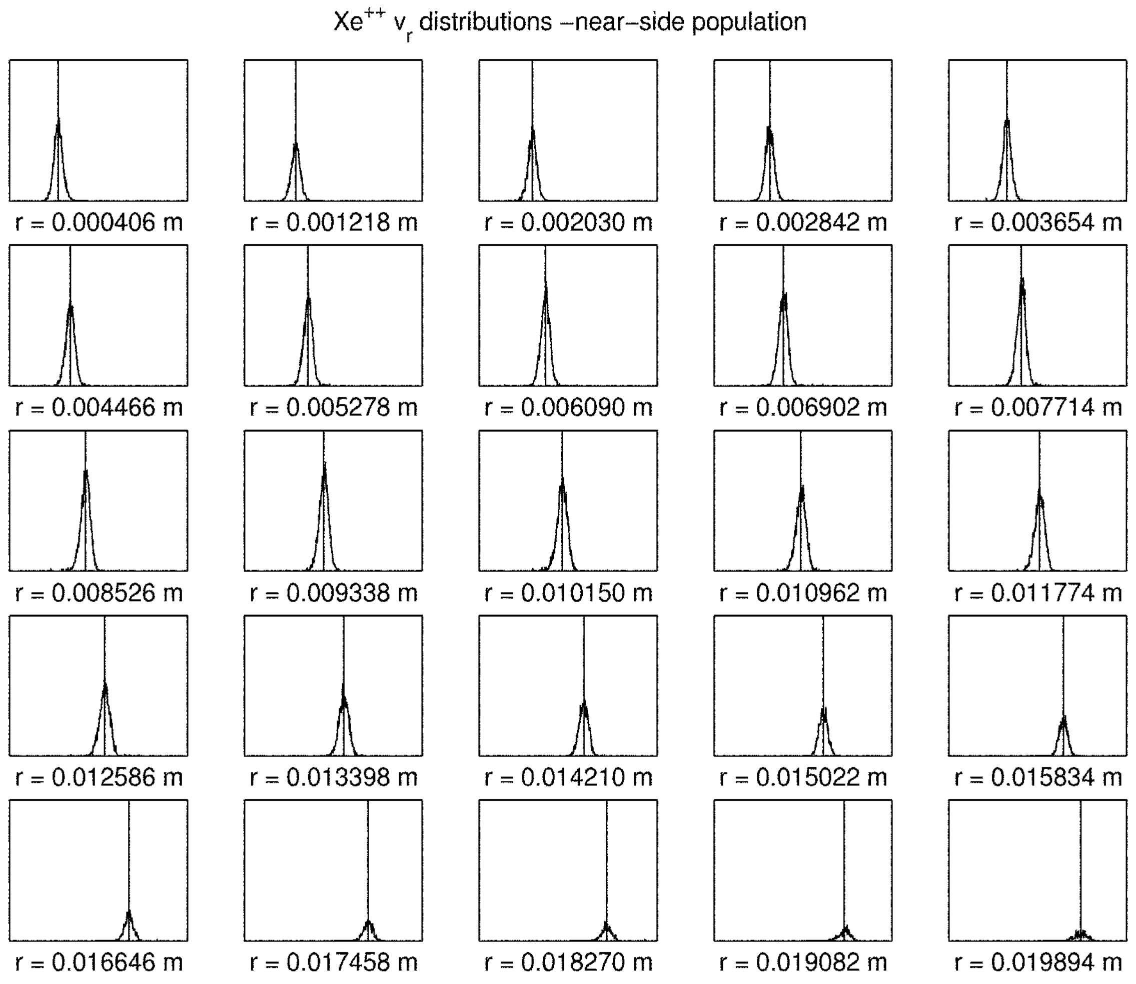

Figure 3-27: Near-side population double ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of near-side Xe^{++} ranging from 0 to 4×10^{11}.

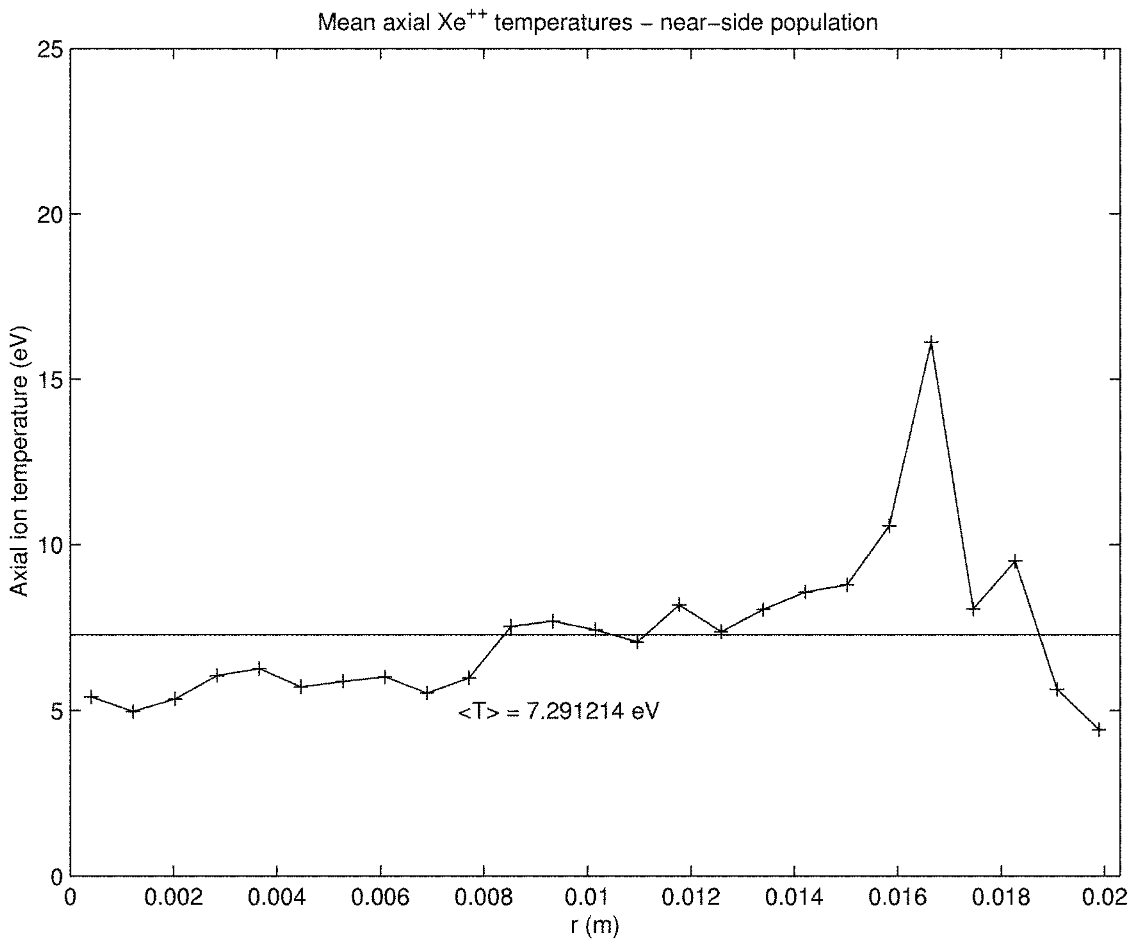

Figure 3-28: Axial near-side double ion temperature.

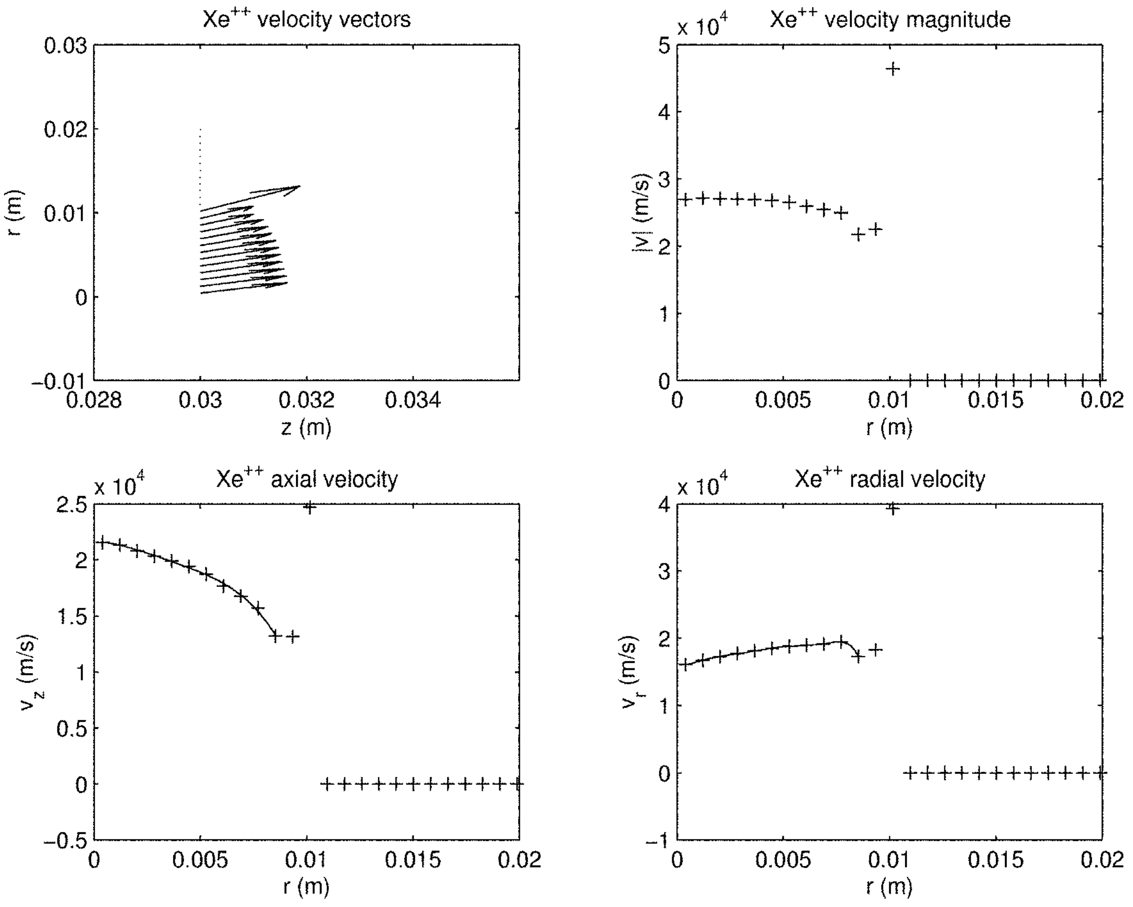

Figure 3-29: Double ion far-side population velocity distributions.

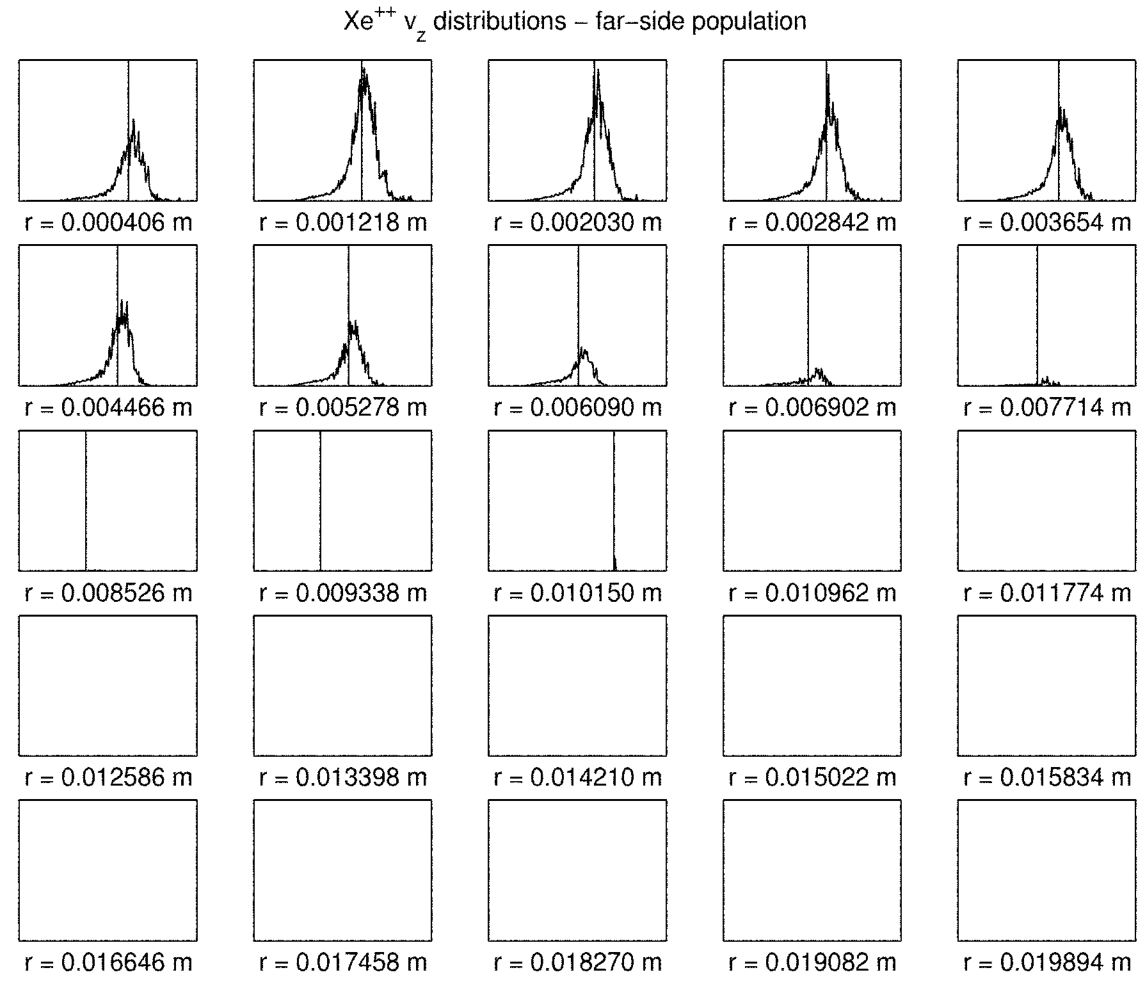

Figure 3-30: Far-side population double ion axial velocity distributions. Horizontal axes are axial velocity, v_z, ranging from 0 to 35000 m/s. Vertical axes are number of far-side Xe^{++} ranging from 0 to 1×10^{11}.

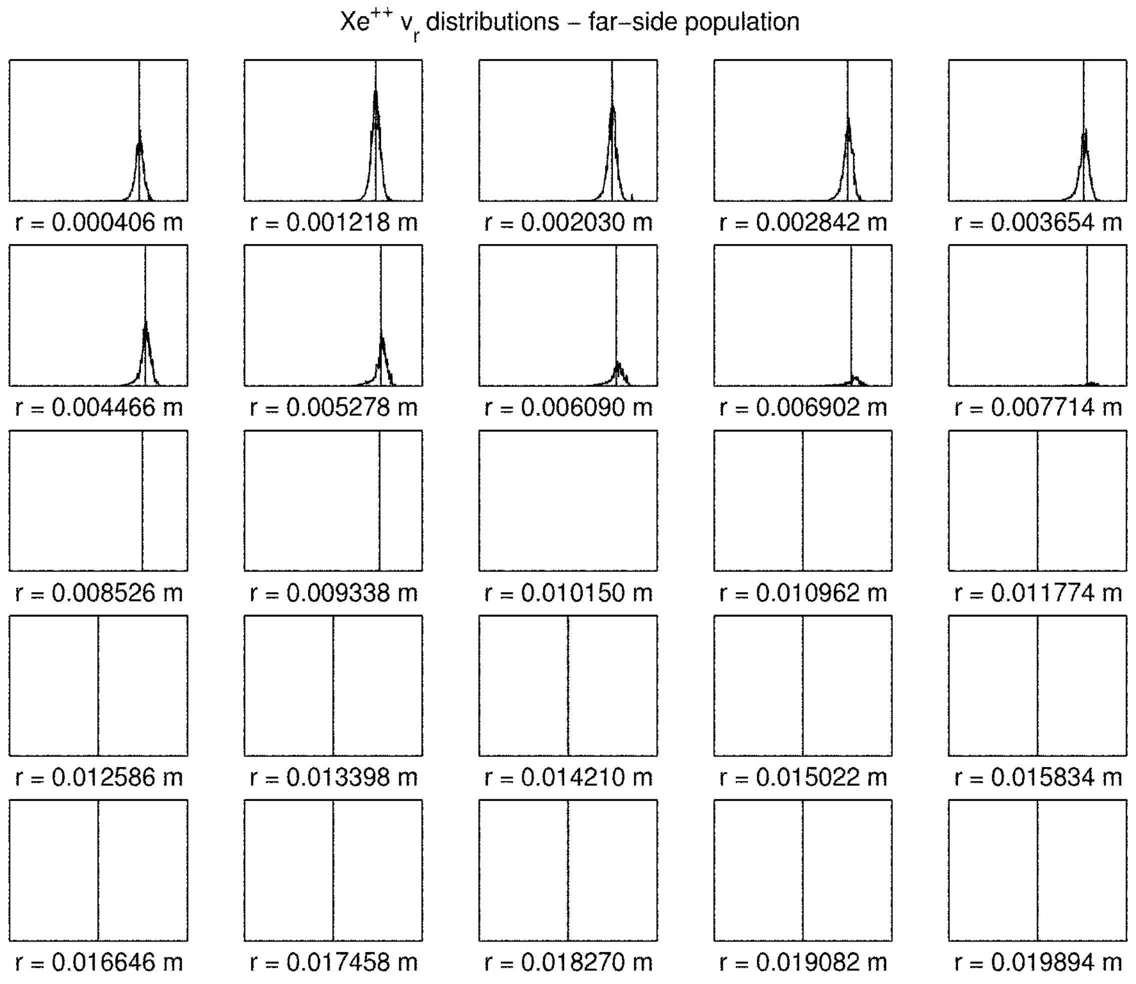

Figure 3-31: Far-side population double ion radial velocity distributions. Horizontal axes are radial velocity, v_r, ranging from -35000 to 35000 m/s. Vertical axes are number of far-side Xe^{++} ranging from 0 to 15×10^{10}.

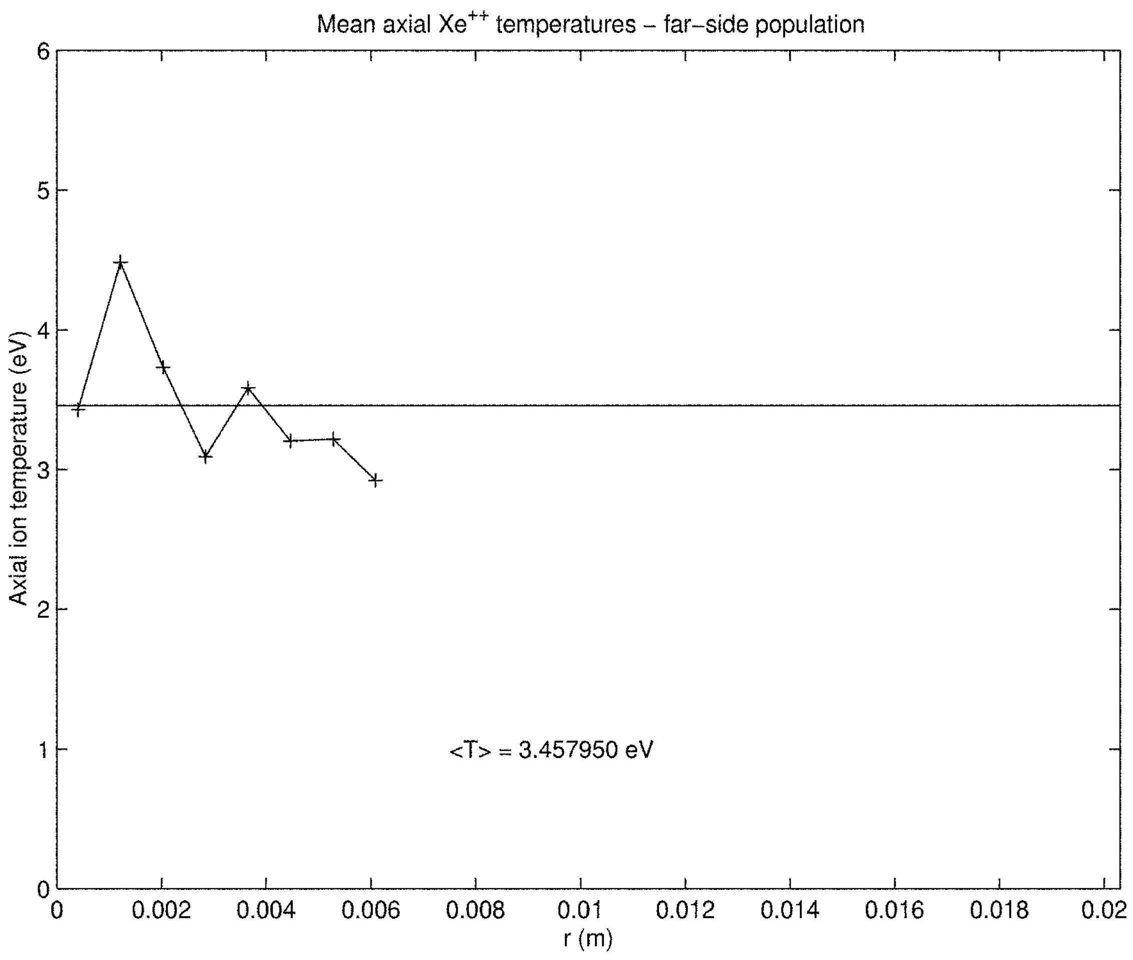

Figure 3-32: Axial far-side double ion temperature.

Exit Plane Outer Radius, r_1	0.0203 m
Exit Plane Inner Radius, r_2	0.0000 m
Anode Propellant Fraction, f_a	0.875
Anode Utilization Fraction, η_u	0.6957
Xe^+ Azimuthal Drift Velocity	221.4 m/s
Xe^+ Axial Near-side Ion Temperature	2.96 eV
Xe^{++} Axial Near-side Ion Temperature	3.47 eV
Xe^+ Axial Far-side Ion Temperature	7.29 eV
Xe^{++} Axial Far-side Ion Temperature	3.46 eV
Xe^+ Azimuthal Temperature	0.068975517 eV
Xe Temperature	0.0603448276 eV
Cathode Orifice Radius	0.0037338 m
Cathode Axial Offset	0.0094 m
Cathode Radial Offset	0.0472 m
Anode Double Ion Fraction	0.125

Table 3.5: Improved BHT-200 source model parameters.

3.3.2 Additional parameters

Most additional parameters remain the same from the old source model with the exception of the calculated axial ion temperatures, the anode ionization fraction, and the fraction of ions that are doubly charged. Anode ionization fraction is calculated as before using Equation 3.4 and *HPHall* results. However, instead of assuming only singly-charged species when converting beam current to an ion mass flow, an average charge per particle is found to account for the existence of double ions. From the exit plane information, the anode double ion fraction is found to be 12.5% by number, making the average particle charge 1.8×10^{-19} C. Table 3.5 summarizes the new source model parameters.

3.4 Comparison of Source Models

Figures 3-33 and 3-34 show the ions for the two source models after the source particles have reached steady-state. It is clear that the plume of the old source model is more beam-like while the new source model exhibits more divergence. This can be explained since the old source model assumes the same ion drift velocity across the exit plane.

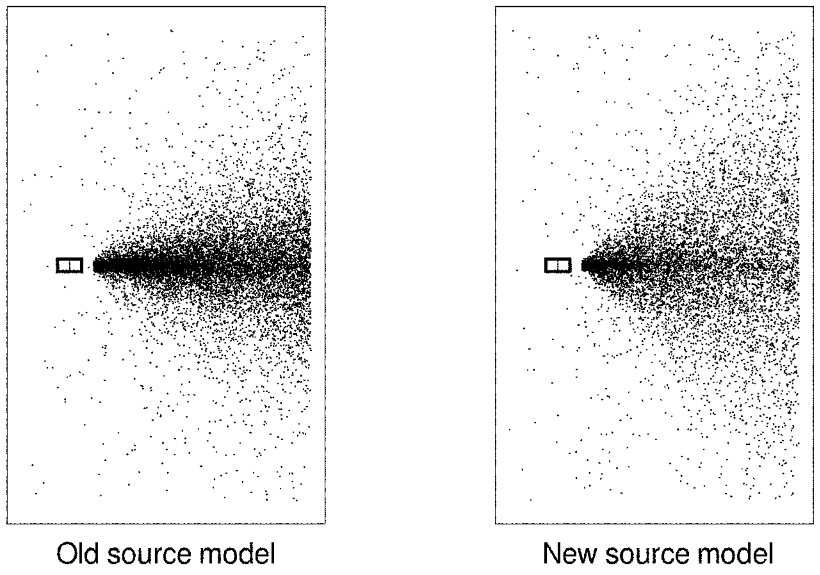

Figure 3-33: Comparison of source models - single ions.

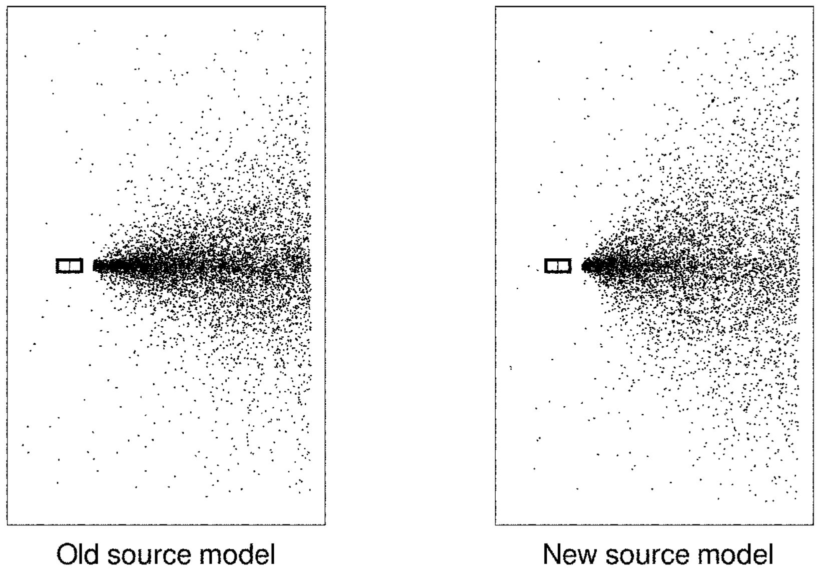

Figure 3-34: Comparison of source models - double ions.

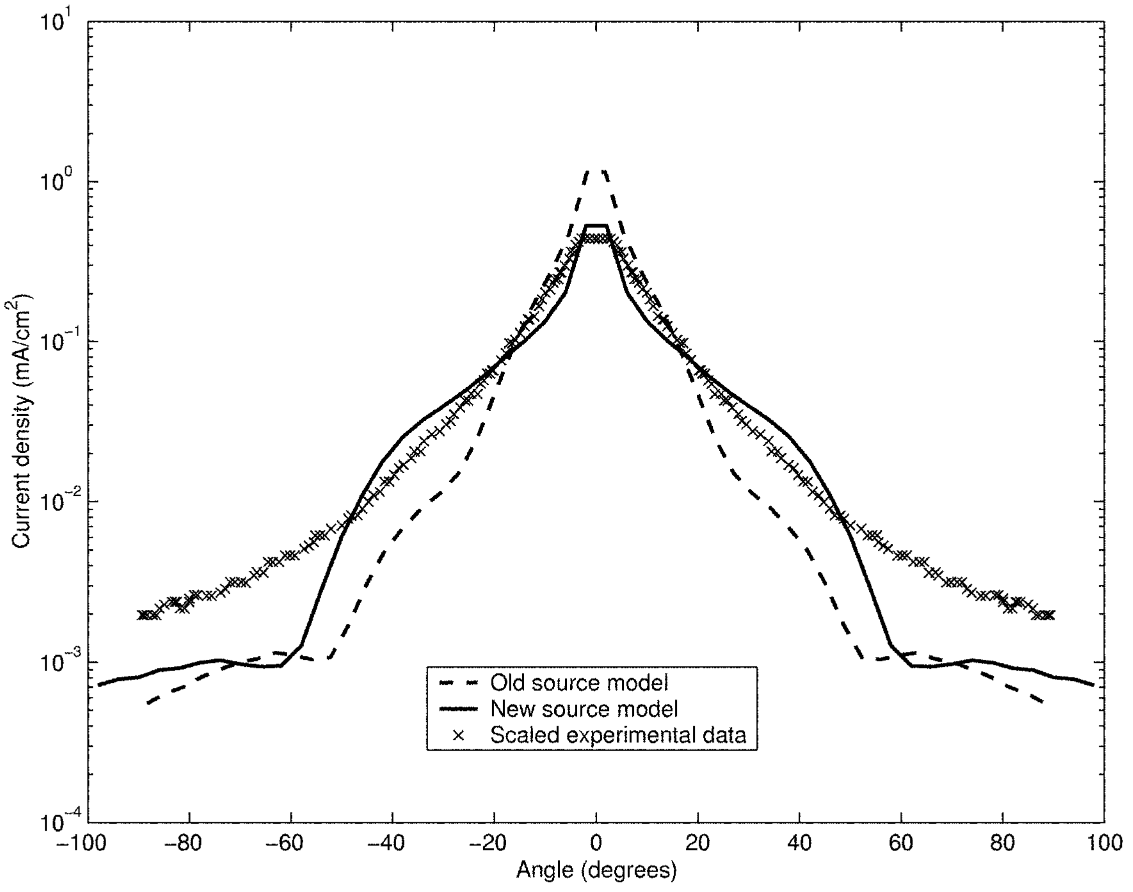

Figure 3-35: Comparison of source models.

In contrast, the new source model incorporates velocity distributions across the exit plane, modeling slower particles at the edge of the plume that will be affected more by the radial potential and cause more divergence. This behavior can also be observed in the current density distributions of the two source models as seen in Figure 3-35. The new source model has higher values of current density at larger angles, reflecting the presence of the sideways-pushed ions.

The simulated current densities of the two source models are also compared to scaled experimental data. As there is currently no experimental data available for the BHT-200, data taken from a SPT-100 by Pencil et. al. [12] is scaled to provide a basis for comparison. Figure 3-36 shows this data — for comparison to the BHT-200, scalings to account for the difference in power and measuring distance are performed. Current density is assumed to scale linearly with power and inversely with the square

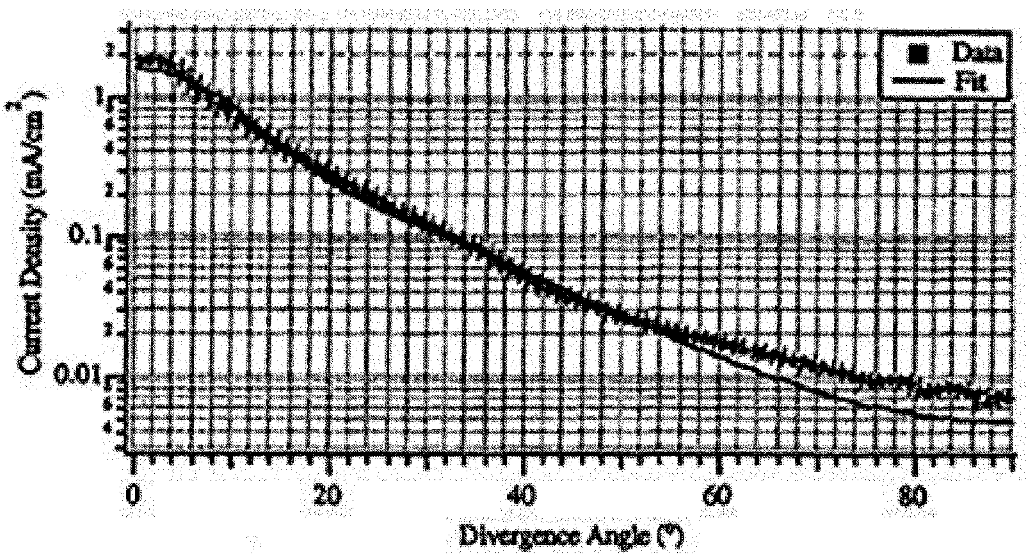

Figure 3-36: Ion current density versus divergence angle, measurements scaled to 1 m radius from the thruster exit plane. [12]

of the measurement distance. After these scalings, the resultant distribution is integrated to find the beam current which is 0.6127 A, greater than the .4022 A of the source model. This discrepancy may arise from instrument errors meaning the original unscaled plume data also does not integrate to the expected SPT-100 beam current or from a difference in the ratio of beam to anode current, $\frac{I_b}{I_a}$, between the two thrusters. As a result, an additional scaling is added. The final scaling formula is given by,

$$j_{BHT} = \alpha \, j_{SPT} \left(\frac{P_{BHT}}{P_{SPT}}\right) \left(\frac{r_{SPT}}{r_{BHT}}\right)^2 , \tag{3.23}$$

where the extra scaling factor is,

$$\alpha = \frac{(I_B)_{BHTsimulated}}{(I_B)_{BHTscaled}}. \tag{3.24}$$

Because of its greater detail in modeling the exit plane velocity distributions, the improved source model shows better agreement of current density magnitude, especially at small angles, with experimental data.

Chapter 4

Collisons

4.1 Overview

As previously discussed, collisions play an important role in affecting the plume's behavior, particularly under vacuum tank conditions. Several improvements to the *Quasi3* collision method have been implemented.

4.2 Charge Exchange Cross-section

The dominant collision phenomena in the plume region is the resonant charge exchange process where an ion gives its charge to a neutral.

$$\mathrm{Xe} + \mathrm{Xe}^{+} \rightarrow \mathrm{Xe}^{+} + \mathrm{Xe}$$
$$\mathrm{Xe} + \mathrm{Xe}^{++} \rightarrow \mathrm{Xe}^{++} + \mathrm{Xe}$$

These collisions are of potential concern in a Hall thruster plume as they typically involve a fast-moving ion giving its charge to a slow-moving neutral, resulting in a slow-moving ion and a fast-moving neutral. These slow-moving ions then do not have enough kinetic energy to overcome the potential hump of the thruster and may be accelerated at high angles or even backwards, coming in contact with the spacecraft, solar arrays, or other components in their path. While having less energy than the main beam ions, CEX ions still have enough energy to cause harmful damage through sputtering. Thus, their accurate modeling is important to predict surface interactions properly.

In modeling Xe - Xe^+ CEX collisions, *Quasi3* uses the collision cross-section of Rapp and Francis [14] given by,

$$\sigma_{RF} = (-0.8821 \ln c_r + 15.1262)^2 \times 10^{-20}\, m^2, \tag{4.1}$$

where c_r is the relative velocity in m/s.

However, Szabo [16] argues that better accuracy is achieved with the Sakabe and Izawa [15] cross-section. This cross-section,

$$\sigma_{SI} = [1.81 \times 10^{-14} - 2.12 \times 10^{-15} \log c_r](I/I_o)^{-1.5}\, cm^2 \tag{4.2}$$

where c_r is in cm/s, I is the ionization potential of xenon (12.1 eV) and I_o is the ionization potential of hydrogen (13.6 eV), agrees better with experimental data and results in cross-sections about 30% larger than those of Rapp and Francis. Thus, the Sakabe and Izawa cross-section has replaced that of Rapp and Francis in *Quasi3*.

4.3 Elastic collisions

As pointed out by Katz et. al. [7], elastic scattering may play a part in the relatively high-energy ions observed at large angles in the plume. These particles are not attributed to CEX collisions which should result in lower energy ions. The ion-neutral elastic collision method is slightly modified as it previously assumed final trajectories for particles, defying quantum mechanics in that the identity of each particle was known a priori. The method now randomly assigns final trajectories to each of the particles involved. The result is that more source ions may be scattered off neutrals at large angles while retaining their high energy levels, explaining the high energy peak measured in the plume wings. However, this modification did not have much effect on simulation results as seen in Figure 4-1.

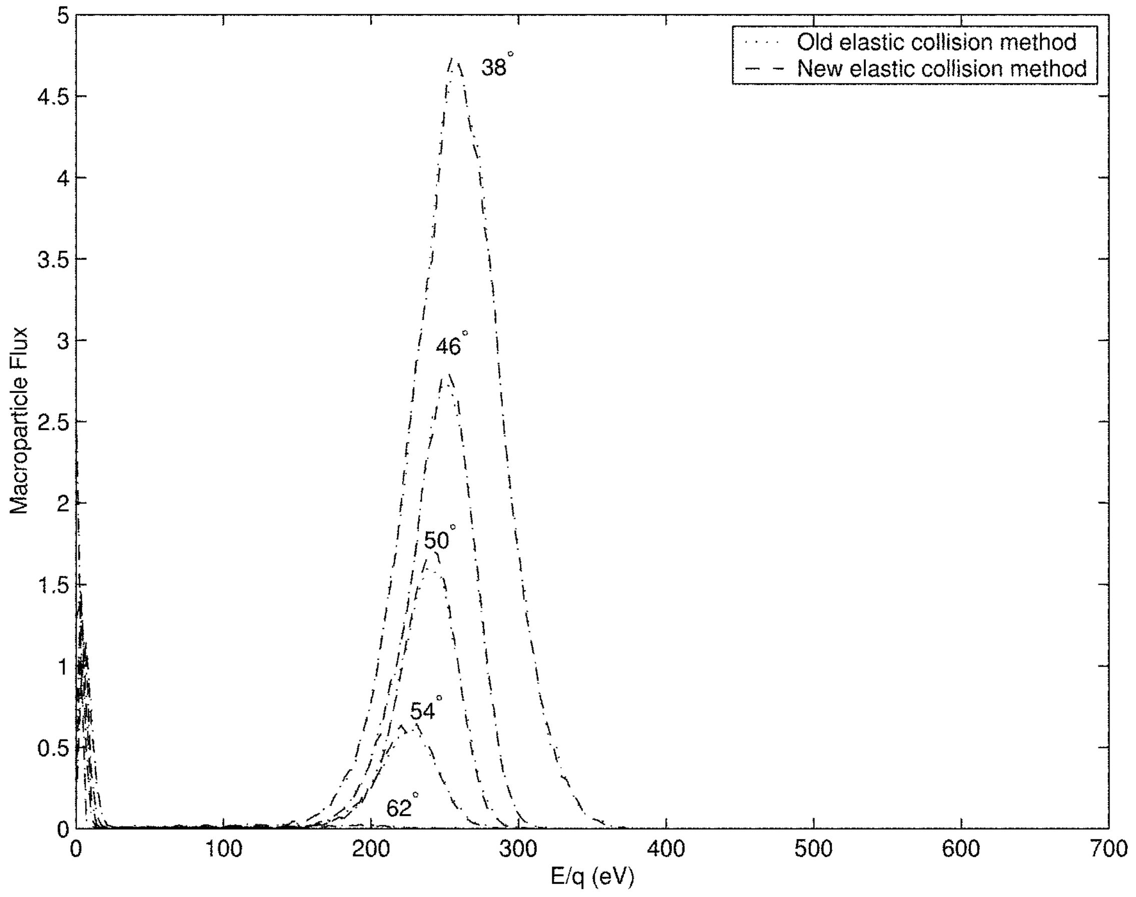

Figure 4-1: Simulated retarding potential analyzer (RPA) measurements for P = 5×10^{-6} *Torr*.

4.4 Neutral weighting selection

As discussed in Chapter 2, a PIC method is used to model the ions and neutrals of the plume. Weighting factors for each species are chosen so that enough macroparticles are present in the simulation to maintain statistical accuracy. Because of the implementation method of the collision module, particles of the same species are restricted to having the same macroparticle weight. As explained in Section 2.1.2, after each collision, a cell's local time counter is incremented by a collision-dependent time step. This time increment is given by the multi-species time counter,

$$\Delta t = \frac{W_L V}{N_o W_o (2N_1 W_1 + 2N_2 W_2 + \frac{N_o W_o}{2}) \sigma_{ij} c_r}, \tag{4.3}$$

where the subscripts 0, 1 and 2 represent neutrals, single ions and double ions respectively. N is the number of macroparticles of the species in the cell, W is the species weighting factor, W_L is the lower weighting factor of the two collision partners, V is the cell volume, and c_r is the relative velocity of the two particles. σ_{ij} is the cross-section of the collision being modeled, σ_{00}, σ_{01}, or σ_{02}.

In his thesis, Oh outlines some limitations to using this multi-species time counter. Since it is linearly dependent on the smaller of the two particle weightings and Xe - Xe elastic collisions are modeled, the neutral weighting should be on the same order as those of the ion species. If the neutral weighting is much higher than the ion species weightings, when a Xe - Xe elastic collision occurs, W_L will be a large value, as will Δt. Thus, the local time counter may be incremented by a value larger than the global time step, effectively removing the cell from consideration for collisions over several time steps. Setting the neutral weighting generally becomes a concern only when trying to model a plume with a background density, for example, in a vacuum tank. If the plume is being modeled in a perfect vacuum, the combined neutral flux from the thruster and its cathode is small enough that a small neutral weighting does not slow the simulation down. However, when a background neutral density is included, many more neutral particles must be simulated. If the neutral weighting is kept on the same order as the ion species weightings, a significant amount of the computation time and memory is spent on background neutrals. To avoid this issue, Oh proposes that when modeling a background density, Xe - Xe elastic collisions should not be included, removing the concern of large collision time steps and permitting a larger neutral weighting.

Another issue that arises due to the neutral weighting occurs when trying to achieve good resolution around the thruster exit by placing embedded grids in this region. Since the weighting of every neutral macroparticle in the simulation must be identical, cells in finer meshes may only contain a few macroparticles due to their small volume. In order to preserve good statistics, *Quasi3* requires a minimum of four neutral macroparticles in a cell before collisions occur. Hence, if the neutral weighting is set too high, embedded grid cells near the thruster exit may not have enough simulated particles and this critical region will have no collisions.

Chapter 5

Sputtering

5.1 Overview

Sputtering is the process in which atoms of a target material are removed by impinging particles. Sputtering yields, the number of target atoms removed per incident particle, are dependent upon incident particle energy, incident angle, and target surface composition. Particles found in Hall thruster plumes typically have energies above sputtering thresholds and concerns about possible damage to spacecraft components have been raised. Understanding sputtering in a tank environment is also important in order to distinguish between sputtered material from the tank and from the engine.

As mentioned in Chapter 2, the version of *Quasi3* containing the sputtering method improvements of Qarnain and Asare could not be run properly, so a version prior to their changes is the starting point for this research. The angular dependence of sputtering yield used by Qarnain and the normal sputtering yield used by Asare are re-implemented. Asare's addition of inserting sputtered particles into the simulation is also redone. However, instead of requiring all materials to be composed of aluminum, the user now has the option in *Mesh3* to specify object materials. Currently, sputtering off of aluminum, iron, silver, gold, silicon, and molybdenum has been implemented. It is fairly simple to add more materials and this procedure is described in more detail in B.5.

5.2 Normal Sputtering Yield

The method of Yamamura et. al. [20] for angular dependence of sputtering yield recommends using the third Matsunami formula for the normal incidence sputtering yield. This is given by,

$$Y(E) = P\frac{s_n(\epsilon)}{1+0.35U_s s_e(\epsilon)}\left[1-\left(\frac{E_{th\_norm}}{E}\right)^{\frac{1}{2}}\right]^{2.8}. \tag{5.1}$$

Values of P are tabulated in [20] and,

$$s_n(\epsilon) = \frac{3.441\sqrt{\epsilon}\ln(\epsilon+2.718)}{1+6.355\sqrt{\epsilon}+\epsilon(-1.708+6.882\sqrt{\epsilon})}, \tag{5.2}$$

$$s_e(\epsilon) = k\epsilon^{1/2}, \tag{5.3}$$

$$E_{th\_norm} = U_s\left[1.9+3.8\left(\frac{M_1}{M_2}\right)+0.314\left(\frac{M_2}{M_1}\right)^{1.24}\right]. \tag{5.4}$$

ϵ for Equations 5.2 and 5.3 is,

$$\epsilon = \frac{E}{E_L}, \tag{5.5}$$

where,

$$E_L = \frac{M_1+M_2}{M_2}\frac{Z_1Z_2e^2}{a} = 14.39\frac{M_1+M_2}{M_2}\frac{Z_1Z_2}{a}. \tag{5.6}$$

e is elementary charge and is divided by $4\pi\epsilon_o$ to convert it to SI units. Combining this with other factors needed to convert E_L to eV results in the 14.39 factor. k for Equation 5.3 is,

$$k = 0.0793Z_1^{1/6}\frac{(Z_1Z_2)^{1/2}(M_1+M_2)^{3/2}}{\left(Z_1^{2/3}+Z_2^{2/3}\right)^{3/4}M_1(M_1M_2)^{1/2}}, \tag{5.7}$$

and a, in Angstroms, for Equation 5.6 is,

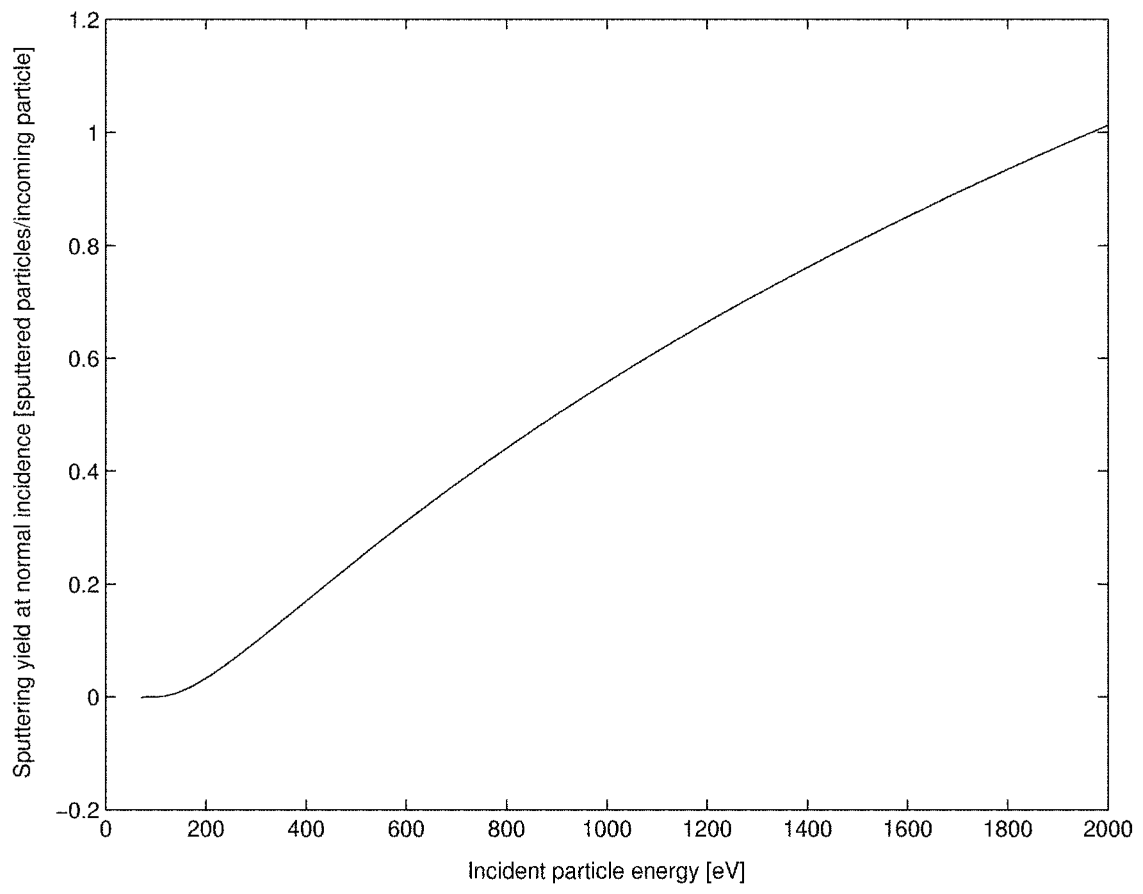

Figure 5-1: Normal incidence sputtering yield for silicon by xenon.

$$a = 0.4685 \left(\frac{1}{Z_1^{2/3} + Z_2^{2/3}} \right)^{1/2} . \tag{5.8}$$

Table 5.1 summarizes parameters used in this calculation and Figure 5-1 shows the normal incidence sputtering yield for silicon by xenon which increases steadily with incident particle energy. The yield will eventually plateau at higher energies since particles are so energetic that they burrow deep into the target material without causing any sputtering. However, these energies lie beyond the range of interest for Hall thrusters.

5.3 Angular Dependence of Sputtering Yield

A correction factor to the normal sputtering yield is calculated with,

$$\frac{Y(\theta)}{Y(0)} = \cos^{-f}\theta \exp -\Sigma(\cos^{-1}\theta - 1), \tag{5.9}$$

where,

$$f = f_s\left(1 + 2.5\frac{1-\zeta}{\zeta}\right), \tag{5.10}$$

$$\Sigma = f\cos\theta_{opt}. \tag{5.11}$$

f_s in Equation 5.10 is the Sigmund f and is tabulated in [20] while,

$$\zeta = 1 - \sqrt{\frac{E_{th\_ang}}{E}}, \tag{5.12}$$

$$\begin{aligned} E_{th\_ang} &= 1.5\frac{U_s}{\gamma}\left[1 + 1.38\left(\frac{M_1}{M_2}\right)^h\right]^2, \\ h &= 0.834, M_2 > M_1 \\ h &= 0.18, M_2 < M_1 \end{aligned} \tag{5.13}$$

$$\gamma = \frac{4M_1M_2}{(M_1+M_2)^2}. \tag{5.14}$$

Parameters from Equation 5.11 are as follows:

$$\theta_{opt} = 90^\circ - 286(\psi)^{0.45}, \tag{5.15}$$

$$\psi = \left(\frac{a}{R_o}\right)^{3/2}\left[\frac{Z_1Z_2}{\left(Z_1^{2/3}+Z_2^{2/3}\right)^{1/2}}\frac{1}{E}\right]^{1/2}, \tag{5.16}$$

Parameter	Al	Si	Ag	Au	Fe	Mo
M_1, projectile mass $[amu]$	131.29	131.29	131.29	131.29	131.29	131.29
Z_1, projectile atomic number	54	54	54	54	54	54
M_2, target mass $[amu]$	26.982	28.086	107.868	196.967	55.845	95.94
Z_2, target atomic number	13	14	47	79	26	42
U_s, sublimation energy $[eV]$	3.39	4.63	2.95	3.81	4.28	6.82
a, screening radius $[\mathring{A}]$	0.1052	0.1045	0.0896	0.0819	0.0976	0.0912
k	0.11	0.112	0.153	0.207	0.125	0.146
E_L $[keV]$	563.5	591.3	904.3	1250	694.5	847.9
E_{th_norm} [eV]	69.2	91.16	19.57	17.74	46.57	49.05
P	25.54	13.66	88.31	81.26	38.47	28.41
f_s	1.8	1.8	1.84	1.73	1.84	1.85
R_o, average lattice constant	2.56	2.72	1.13	0.98	1.12	0.91
γ	0.5657	0.5807	0.9904	0.9600	0.8375	0.9758
E_{th_ang} $[eV]$	72.25	95.24	26.38	23.44	52.2	63.45

Table 5.1: Sputtering yield calculation parameters with Xe as the projectile.

where E is the incident energy in eV. Table 5.1 summarizes parameters used in this calculation.

Figure 5-2 shows the sputtering yield of silicon by incident xenon particles with an energy of 200 eV as a function of incident angle. A maximum for the yield occurs around 60° which represents a minimum in the threshold energy. Sputtering due to heavy particles such as xenon is mainly due to two processes. At angles near normal incidence, the primary mechanism for sputtering is due to the projectile atom penetrating into the target surface, backscattering off interior atoms and into surface atoms which are subsequently ejected from the material as seen in Figure 5-3. As incident angle increases, the threshold energy for this process increases and the dominant mechanism at oblique angles then becomes projectile atoms directly knocking out target atoms near the surface, shown in Figure 5-4. This process starts out with a high threshold energy at normal incidence which steadily drops with increasing incidence angle. The combination of these two mechanisms results in a minimum in the sputtering threshold energy which is then reflected in the sputtering yield maximum.

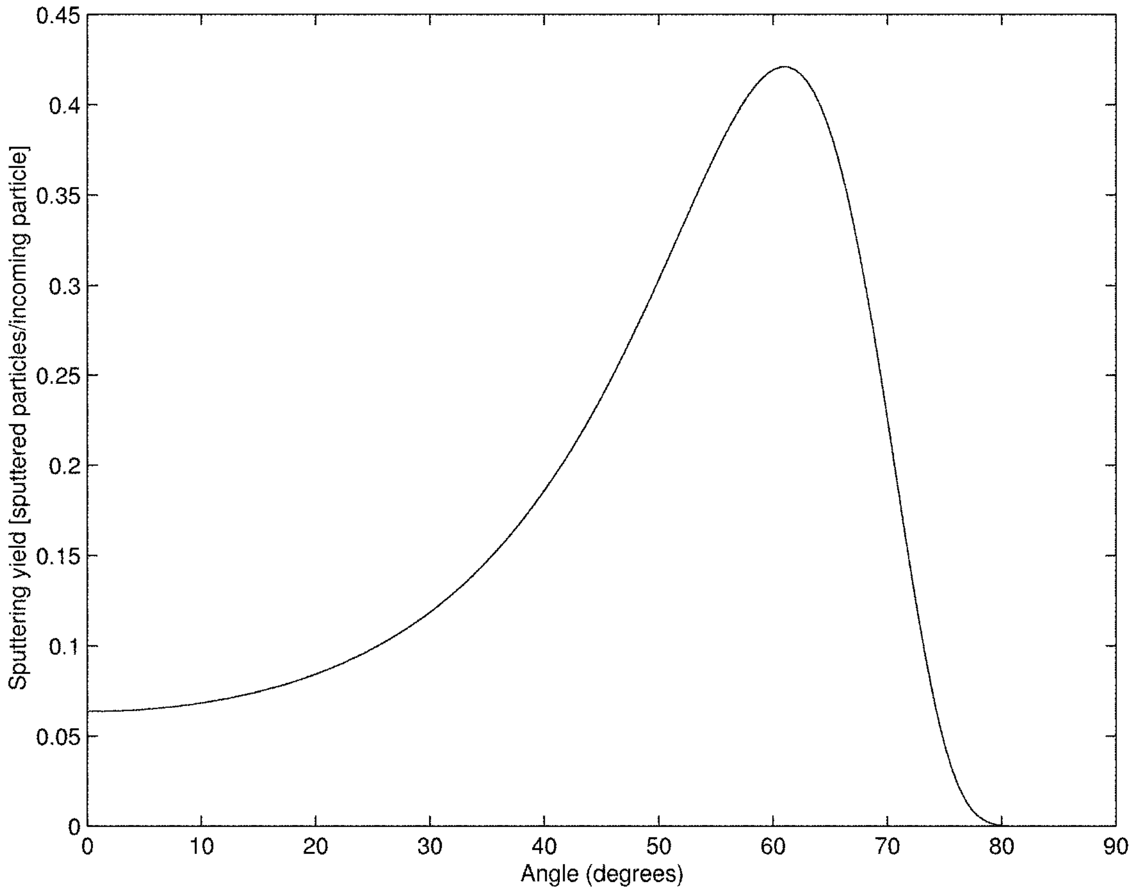

Figure 5-2: Sputtering yield of silicon by incident xenon at 200 eV.

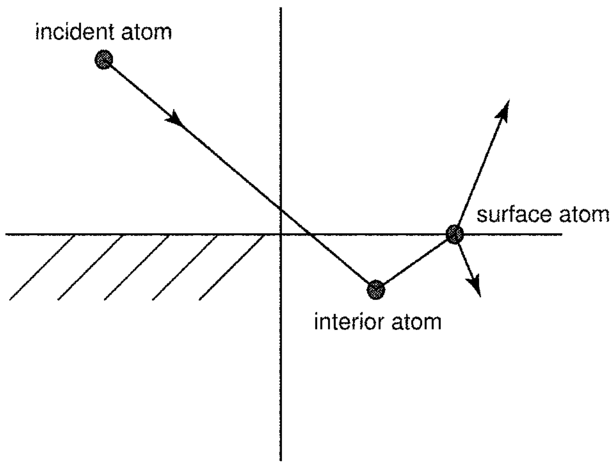

Figure 5-3: Dominant sputtering mechanism near normal incidence.

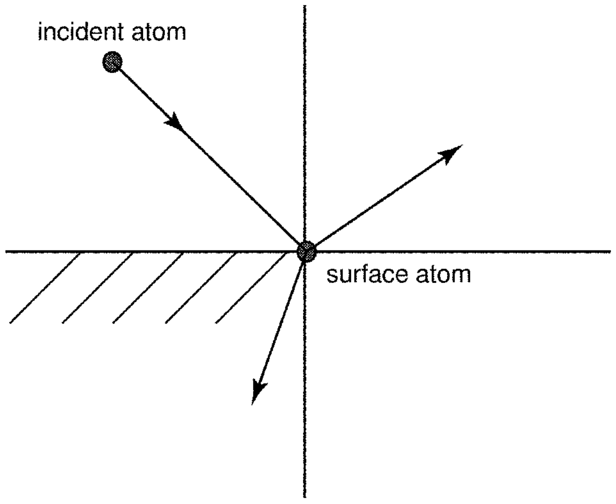

Figure 5-4: Dominant sputtering mechanism at oblique incidence.

Element	Absolute E_{th} $[eV]$	Pseudo E_{th} $[eV]$
Aluminum	69.3	100
Silicon	91.3	150
Silver	20	40
Gold	18.9	50
Iron	46.9	70
Molybdenum	49.9	80

Table 5.2: Sputtering yield thresholds.

5.4 Low-energy Sputtering Method

Figure 5-5 shows the sputtering yield of silicon for various incident angles as a function of energy. At energies near threshold, the sputtering yield formulation of Yamamura for non-normal incidence angles breaks down as the yield approaches infinity. Hence, an alternate method for this energy regime has been implemented as illustrated in Figure 5-6. An absolute energy threshold is taken where the sputtering yield at normal incidence goes to zero. Below this energy, it is assumed that no sputtering occurs, regardless of incident angle. Additionally, a pseudo energy threshold is determined for each material by selecting an energy at which the yield curves described by Yamamura still provide realistic values. For each incident angle, a linear extrapolation between the yield at the pseudo energy threshold and zero yield at the absolute threshold is performed to model low-energy sputtering. In this manner, sputtering yields at all energies can be calculated. Table 5.2 tabulates these threshold energies for the materials of interest.

5.5 Angular Distribution of Sputtered Material

Although equations for the outgoing angular distribution of the sputtered particles are described by Asare [1, 8], the formulation was never implemented in *Quasi3*. Instead, ejected particles were assumed to leave the surface at 45° in-plane. In order to better understand where sputtering deposition occurs, this theory has now been incorporated into the simulation.

Once the yield has been calculated, a corresponding number of sputtered macropar-

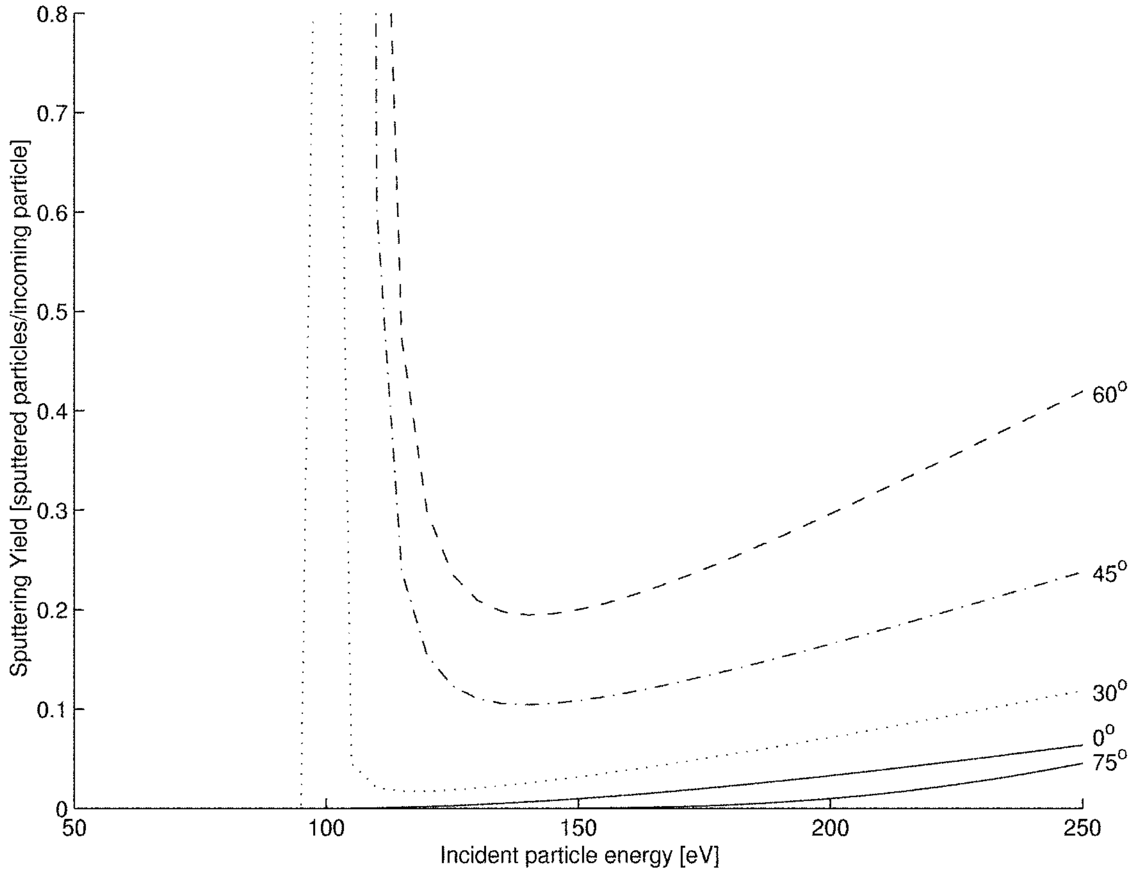

Figure 5-5: Sputtering yield of silicon by xenon for various incident angles.

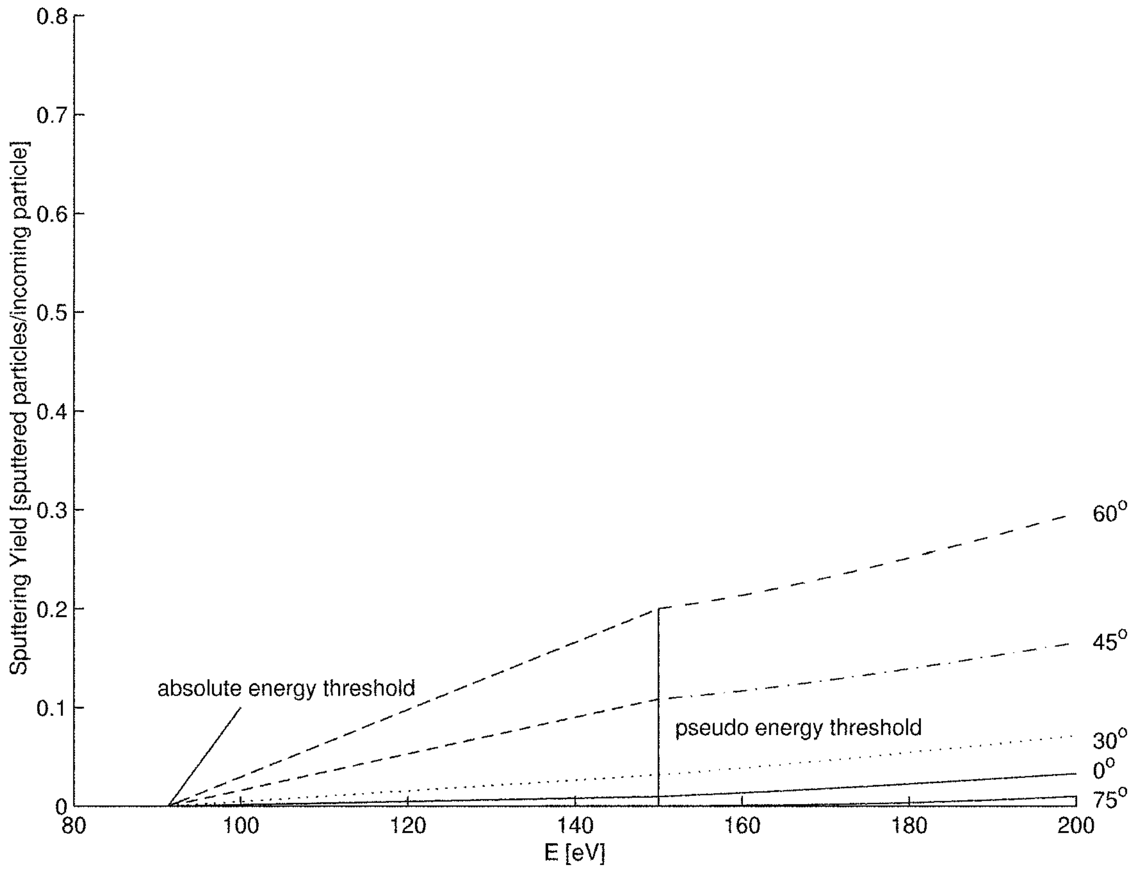

Figure 5-6: Near-threshold energy sputtering curves for xenon striking silicon.

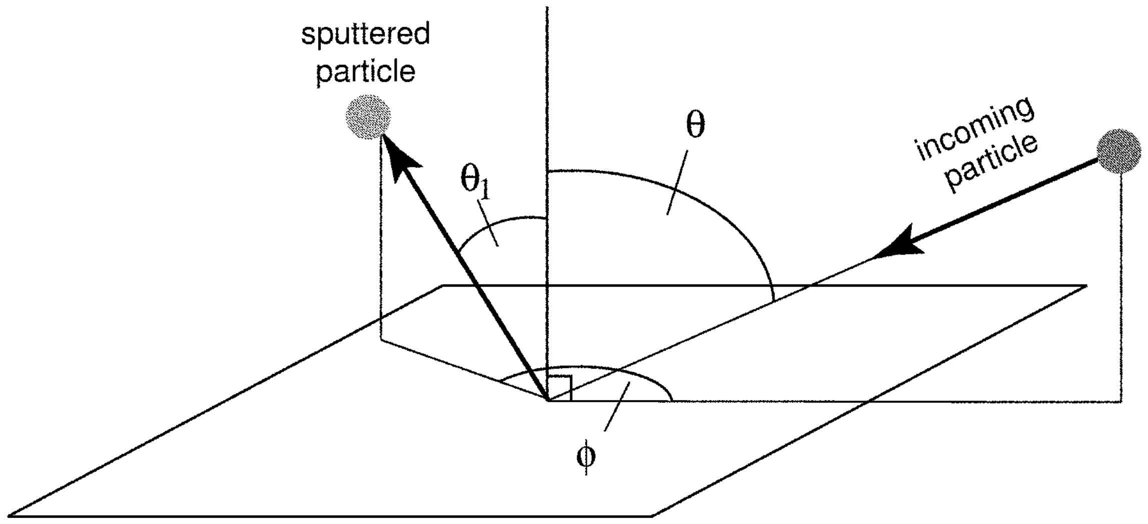

Figure 5-7: Sputtering angles.

ticles are inserted into the simulation. The sputtered particle's outgoing angles are selected from a distribution that depends on incident angle and energy. Figure 5-7 shows the nomenclature for the angles involved.

A differential sputtering yield is defined such that its integral over all possible outgoing angles, θ_1 and ϕ, results in the yield for incident energy, E, and angle, θ as follows:

$$Y(E,\theta) = \int_0^{2\pi} d\phi \int_0^{\pi/2} S(E,\theta;\theta_1,\phi)\sin\theta_1 d\theta_1. \tag{5.17}$$

The differential sputtering yield is given by,

$$S(E,\theta;\theta_1,\phi) = \frac{0.042}{\pi}\frac{\alpha S_n}{U_s}\cos\theta_1\left\{1-\frac{1}{2}\sqrt{\frac{E_{th}}{E}}\left[\cos\theta\gamma(\theta_1)+\frac{3\pi}{4}\sin\theta\cos\phi\sin\theta_1\right]\right\}, \tag{5.18}$$

where,

$$\gamma(\theta_1) = \frac{1}{2}\left[\frac{3\sin^2\theta_1 - 1}{\sin^2\theta_1} + \frac{\cos^2\theta_1(3\sin^2\theta_1+1)}{2\sin^3\theta_1}\ln\frac{1+\sin\theta_1}{1-\sin\theta_1}\right]. \tag{5.19}$$

The integrand of Equation 5.17 is an unnormalized probability distribution function for the outgoing angles. After integrating over all θ_1 and ϕ, the normalized probability distribution,

$$f(\theta_1, \phi)\, d\theta_1\, d\phi = P(\theta_1, \phi, d\theta_1, d\phi), \tag{5.20}$$

$$f(\theta_1, \phi) = \frac{1}{\pi(1-\epsilon\cos\theta)} \sin\theta_1 \cos\theta_1 \left\{1 - \frac{1}{2}\epsilon \left[\cos\theta\gamma(\theta_1) + \frac{3\pi}{4}\sin\theta\cos\phi\sin\theta_1\right]\right\},$$

is achieved, where $\epsilon = \sqrt{\frac{E_{th}}{E}}$. From this, cumulative distribution functions for θ_1 and ϕ are derived as shown below:

$$F_1(\theta_1) = \int_0^{\theta_1} d\theta_1{}' \int_0^{2\pi} f(\theta_1{}', \phi)\, d\phi,$$

$$F_1(\theta_1) = \frac{2}{1-\epsilon\cos\theta}\left[\frac{1}{2}x^2 - \frac{1}{4}\epsilon\cos\theta\left\{x^2 + \left(\frac{1}{2}x^3 - x + \frac{1}{2x}\right)\ln\left[\frac{1-x}{1+x}\right]\right\}\right], \tag{5.21}$$

where $x = \sin\theta_1$, and

$$F_2(\phi, \theta_1) = \frac{\int_0^{\phi} f(\theta_1, \phi')d\phi'}{\int_0^{2\pi} f(\theta_1, \phi')d\phi'},$$

$$F_2(\phi) = \frac{1}{2\pi}\left[\phi - \frac{\frac{3\pi}{8}\epsilon\sin\theta\sin\theta_1\sin\phi}{1 - \frac{1}{2}\epsilon\cos\theta\gamma(\theta_1)}\right]. \tag{5.22}$$

These equations are used to statistically select the outgoing angles of the sputtered paticles. Figure 5-8 gives a sense of the angular distribution shape for a few incident angles.

5.6 Implementation of Sputtering Method

When a simulation particle crosses an interior boundary, *Quasi3* checks if it has enough energy to make it through the sheath. Particles repelled by the sheath are specularly reflected while those that make it through the sheath access material-

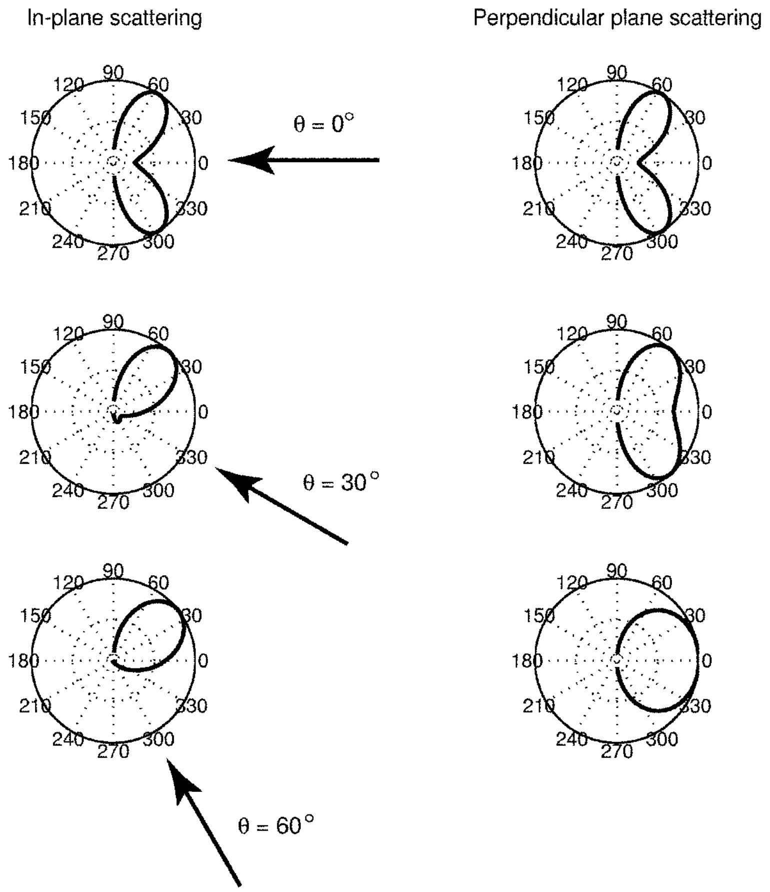

Figure 5-8: Angular distribution of sputtered material for $E/E_{th} = 2$.

dependent sputtering information. If the particle's energy is below the absolute threshold of the material, it is reflected and neutralized (if originally charged) with an energy loss by reflecting the normal velocity and multiplying all velocity components by an accommodation factor. The wall is assumed to be at a temperature of 300 K and particles coming in contact with it are assumed to gain energy if their incident energy is smaller and lose energy if their incident energy is higher as given by,

$$\frac{T_{out} - T_w}{T_{in} - T_w} = \alpha \approx 0.5. \tag{5.23}$$

After many collisions with the walls, particles should eventually thermalise and approach the same temperature as the wall. Particles above the absolute energy threshold cause sputtering and the yield is calculated as described in Sections 5.2 through 5.4. An appropriate number of sputtered macroparticles is then inserted into the simulation. The outgoing angles are chosen using the equations outlined in Section 5.5 — a random number between 0 and 1 is chosen and used with Equation 5.21 to pick the particle's θ_1. This result is then substituted into Equation 5.22 to find ϕ by comparison with another random number. Sputtered particles are assumed to follow straight-line trajectories until they are either deposited on another surface or exit the simulation via an exterior boundary. Measurements by Wehner [19] found velocities of sputtered atoms from metal surfaces by mercury ions corresponding to energies on the order of 10 to 30 eV. Mercury, like xenon, falls in the category of heavy-ion sputtering, so sputtered particles caused by xenon impingement are likely to fall in the same range. This finding supports the straight-line trajectory assumption since the velocities of sputtered atoms are high enough that they do not remain in the plume region long enough to collide with other particles and can be considered a second-order effect. The velocities are also low enough that the particles pose no threat in causing sputtering damage to surfaces themselves. In order to speed the simulation, sputtered particles are given an artificially high velocity to decrease their transit time through the domain.

Chapter 6

Results and Conclusion

6.1 Overview

Various tests are run with the final model to investigate plume behavior. Simulations of current density distribution serve to verify the source model and delve into the effects of background pressure on the plume. More detailed analysis into the causes of these effects provides much insight. Finally, cases in a simulated vacuum tank are run to synthesize all the improvements and to qualitatively verify the sputtering model.

6.2 Current Density Distribution

Runs of the simulation to measure current density distribution are done with the grid pictured in Figures 6-1 and 6-2. The simulation domain shown in Figure 6-1 is large enough to allow measurement of the current density at 0.6 m from the thruster exit. Samples of particles crossing an imaginary hemisphere of radius 0.6 m are taken to generate the simulated distributions. In order to achieve better resolution around the engine exit plane, three successively smaller embedded meshes are placed inside the parent mesh as shown in Figure 6-2. Each embedded mesh has a cell width half as big as its parent. For cases with a background density, the six exterior boundaries are set to have a background neutral flux in order to replenish neutrals

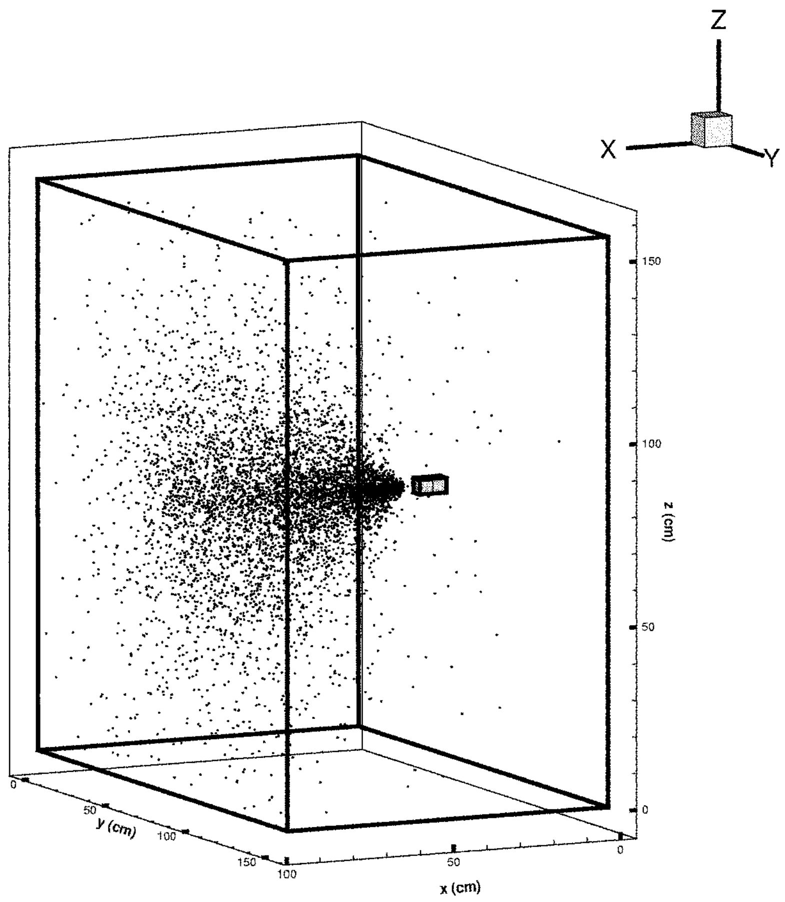

Figure 6-1: 3D view of current density simulation domain.

leaving the simulation and to maintain pressure. Furthermore, for reasons described in Section 4.4, background density runs only use one layer of embedded mesh and do not model neutral-neutral elastic collisions. Simulation runs are for 5,000 iterations.

Typically, since neutral weighting is not an issue in vacuum cases, all collisions, including neutral-neutral elastic collisions, are modeled. However, as seen in Figure 6-3, a broader current density distribution results. The Xe-Xe collisions serve to diffuse the neutrals which in turn leads to the wider distribution of ions. The exclusion of these collisions results in a difference in the plume structure. For more consistent comparisons to the background density cases, vacuum cases are also run without the neutral-neutral elastic collisons.

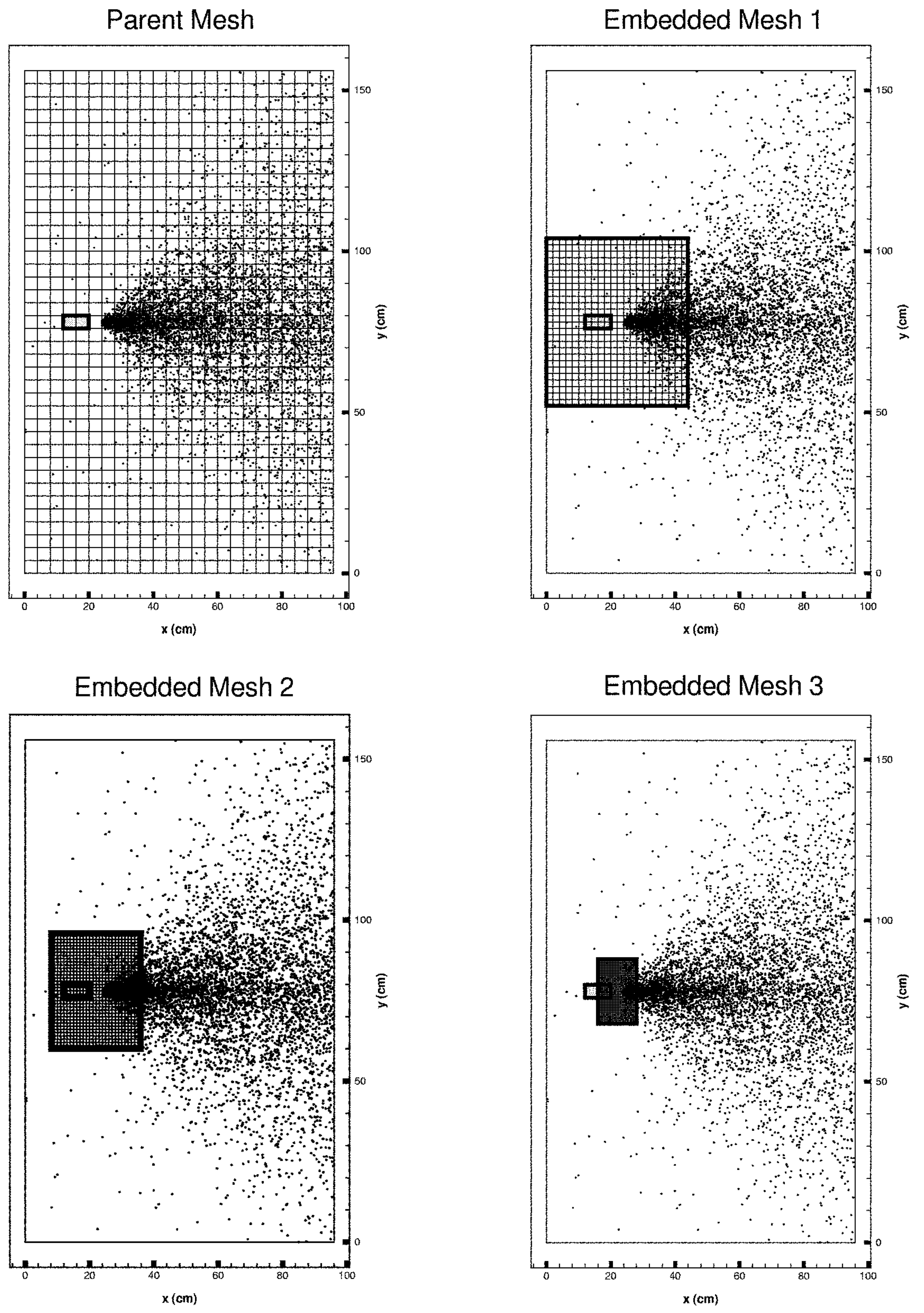

Figure 6-2: Embedded meshes for current density simulation domain.

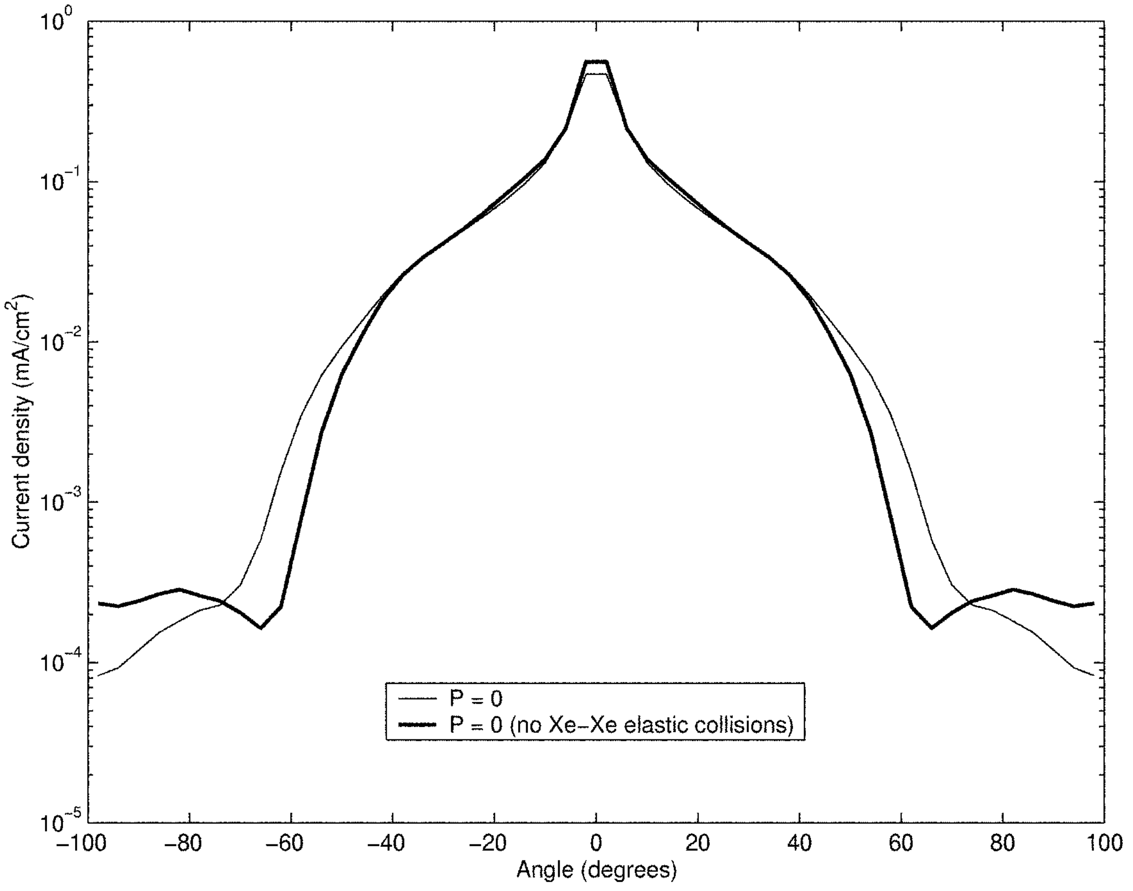

Figure 6-3: Comparison of current density distributions in vacuum.

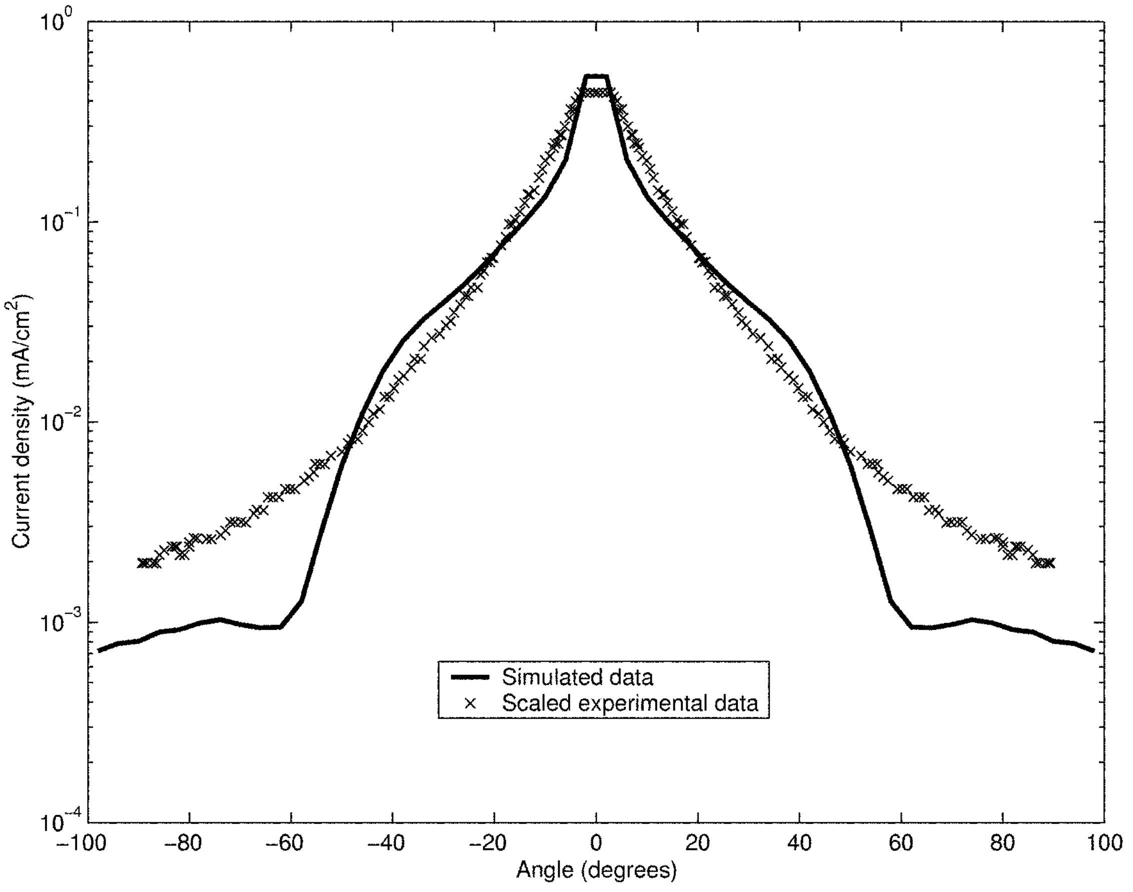

Figure 6-4: Comparison of scaled and simulated data at 60 *cm*.

6.2.1 Comparison to Experimental Data

Figure 6-4 is the comparison of the simulated current density distribution against experimental data scaled as described in Section 3.4 at a pressure of 5×10^{-6} *Torr*. The simulated distribution matches the experimental data peak well. However, the shape of the distribution for the inner angles does not correlate well and the amount of current found in the wings is under-predicted. Section 6.4.1 discusses possible reasons for these incongruities.

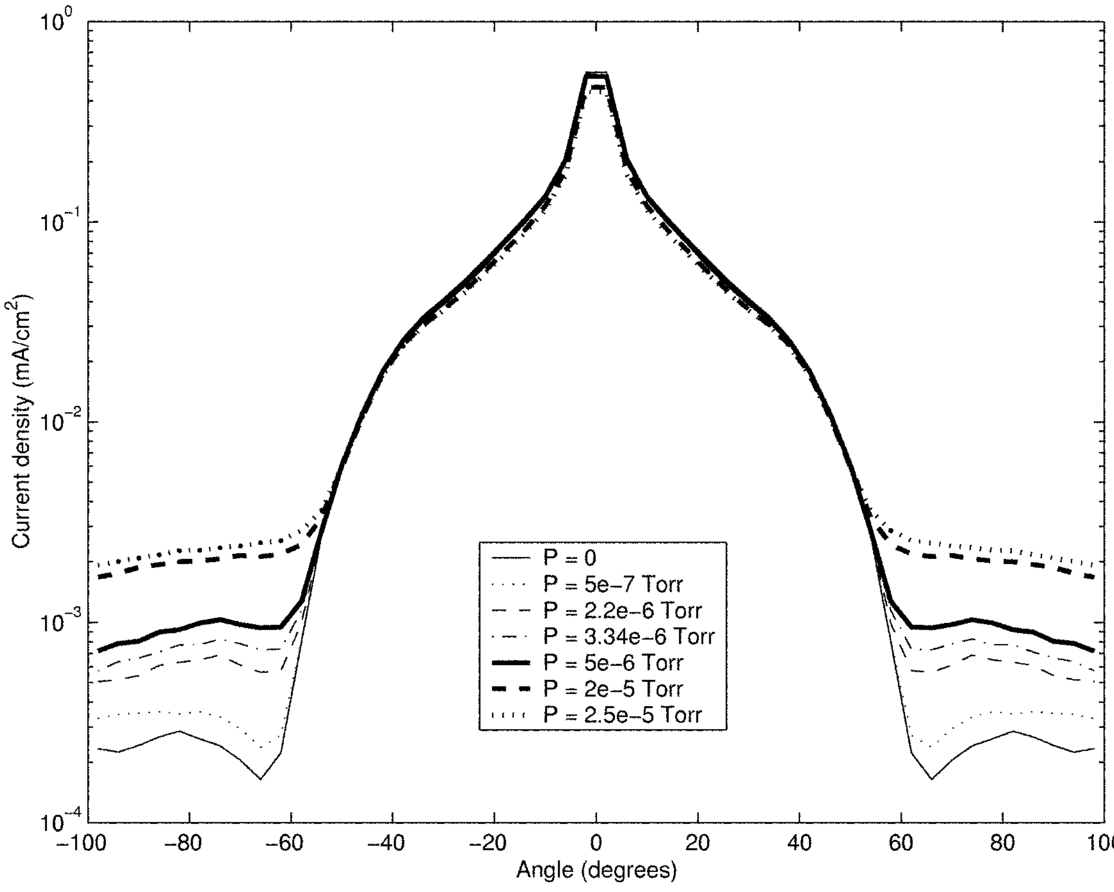

Figure 6-5: Current density distributions for various background pressures at 60 *cm*.

6.2.2 Effect of Background Pressure

Current densities are simulated at various background pressures and results are shown in Figure 6-5. There is a trend of increasing current density in the wings with increasing background pressure which is explained by the larger amount of CEX collisions that occur with higher background densities. The ions produced in these collisions are of lower energy than the source ions and as a result are more affected by the potential hump of the plume and thus are pushed into the larger angle range.

Figure 6-6 shows that the amount of particles with a low energy-to-charge ratio at 70° increases with background pressure. These low energy ions are believed to be the products of CEX collisions. Figures 6-7 and 6-8 are phase-space velocity plots for all single and double ions in the domain at the end of the simulation separated into source, elastic and CEX populations. It is apparent that the ions emanating from the

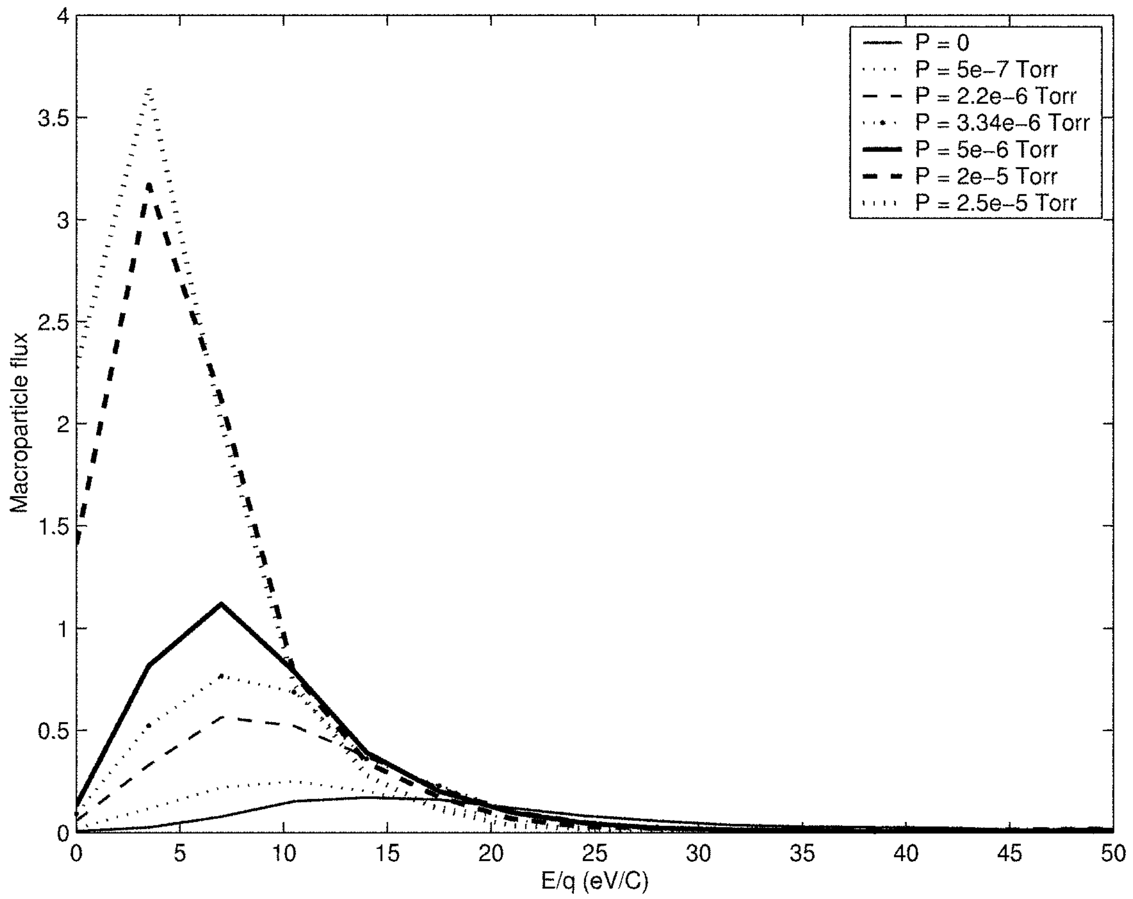

Figure 6-6: Simulated RPA data at 70° for various background pressures.

source are at higher energies than those produced from charge exchange, confirming the conclusion that the low-energy ions are from CEX processes. As expected, ions from elastic collisions fall in an energy range between those of the source and CEX ions. Figure 6-9 shows all charge exchange ions for two different pressures. Clearly, more charge exchange ions exist at the higher value of background pressure. Figure 6-10 shows all source ions for the same pressures for which little difference is observed. It is obvious that the charge exchange ions are found throughout the domain, including the high angles, whereas the source ions are mainly directed forward in the thrust direction. Hence, hypotheses about the effect of background pressure on CEX ion production and their location are justified.

A more detailed look at the effect of the background pressure on plume structure is done by running cases where the neutral flux from the source (that coming from the

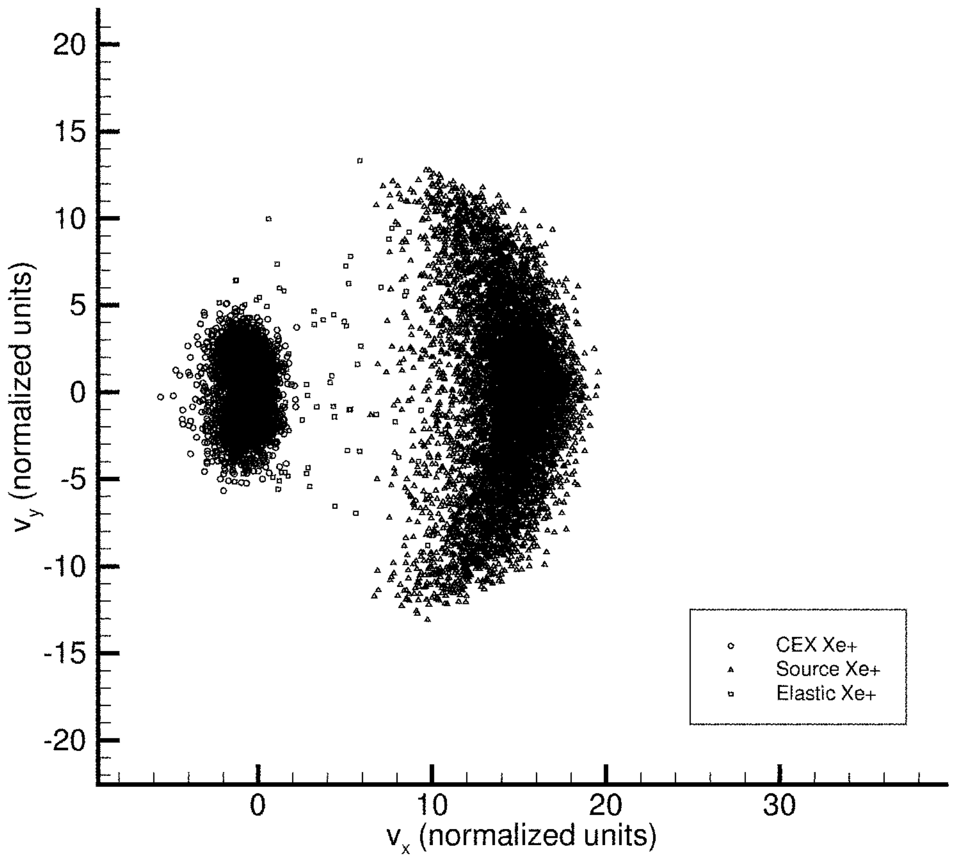

Figure 6-7: Velocity phase-space plot for single ions.

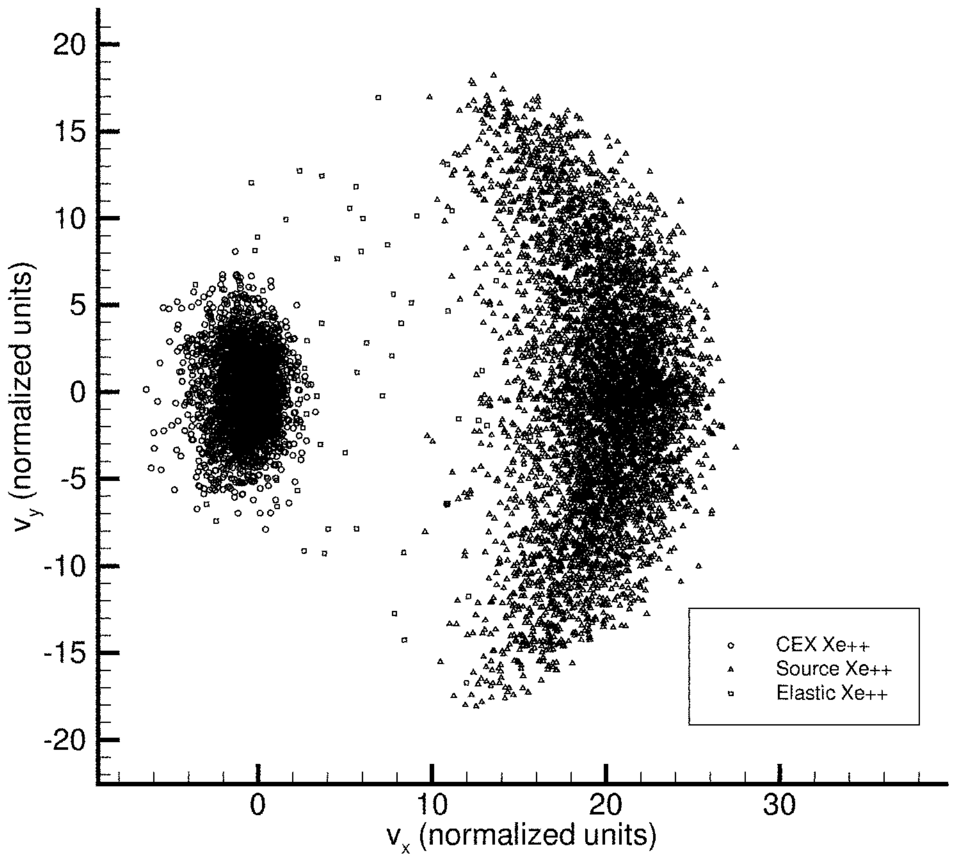

Figure 6-8: Velocity phase-space plot for double ions.

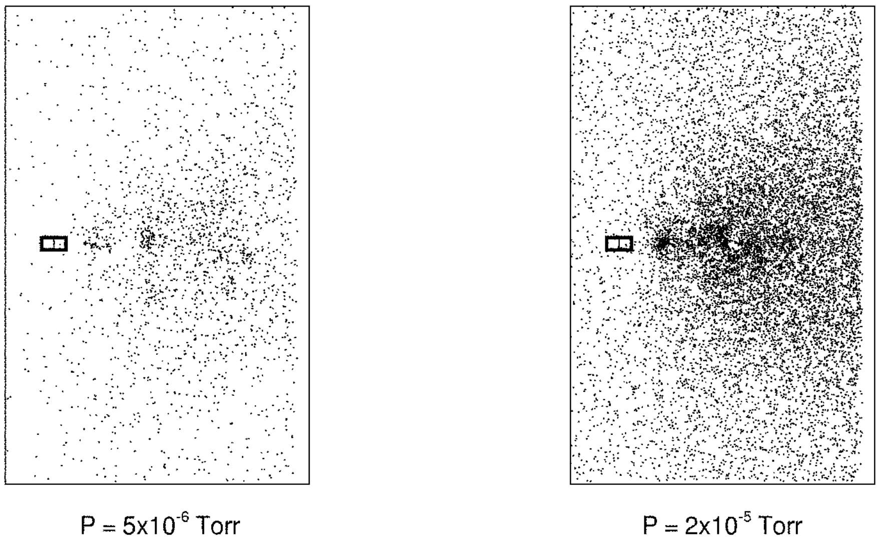

Figure 6-9: Charge exchange ions for different background pressures.

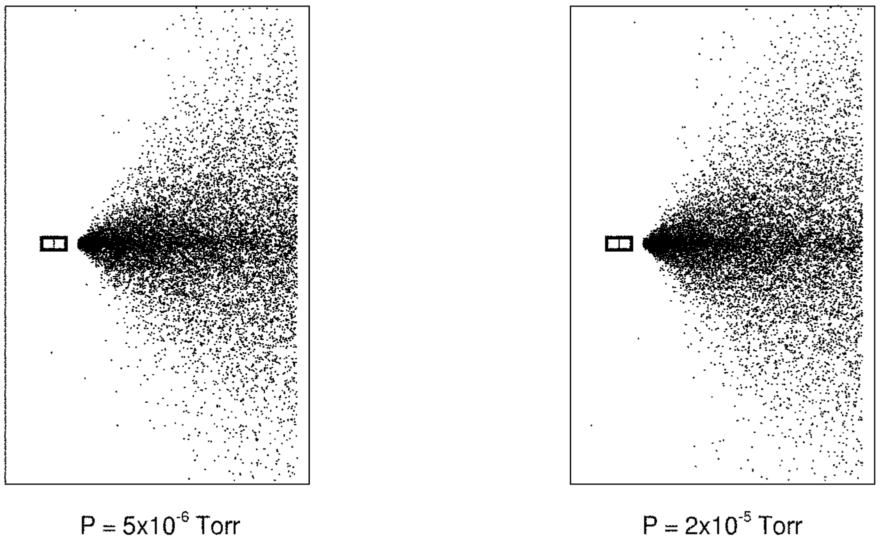

Figure 6-10: Source ions for different background pressures.

main engine and the cathode) is set to zero. The current densities are then solely a result of interaction with neutrals of the background density. These distributions, as well as those from the baseline cases, are depicted in Figure 6-11. It is observed that the current densities resulting from only the effect of the background neutrals follow the predicted pattern of higher wings with increasing pressure and that the presence of the background density becomes the dominant effect in the distribution's outlying angles. To better understand this, Figures 6-12 through 6-17 are generated. As seen in Figures 6-12 through 6-14, the current density in the central region of the plume decreases as background pressure increases since the presence of more background neutrals causes a greater number of source ions to be scattered out of the plume's core. The cases that exclude the source neutrals have higher current densities because there are less neutrals for ions to interact with. This decreasing trend continues until about 50° where growth of the current density with background pressure begins to emerge. The ions lost from the center of the distribution make their appearance in the larger angles which explains this result. The cases that include source neutrals typically have the higher current density values in this regime since more ions are scattered into the wings. Figures 6-15 through 6-17 confirm that source ions are being diverted from the smaller to the higher angles of the plume. Small angles are almost entirely dominated by source ions. However, as angles become larger, the fraction of CEX ions begins to climb, especially at higher background pressures. By the time the outer regions of the plume are reached, CEX ions have become the main contributor to the current density. An interesting observation is that the angle where crossover of the source and CEX ion fractions for a certain pressure occurs roughly corresponds to the angle where a "kink" in the current density distribution of Figure 6-11 for that pressure happens.

It is clear that the wings of the current density distribution become increasingly dominated by the presence of a background gas as pressure rises. If these large angles are the domain of interest in a vacuum tank experiment, a much lower pressure must be maintained to capture realistic plume behavior. At a critical pressure, when the background density becomes the overpowering effect, extrapolation of experimental

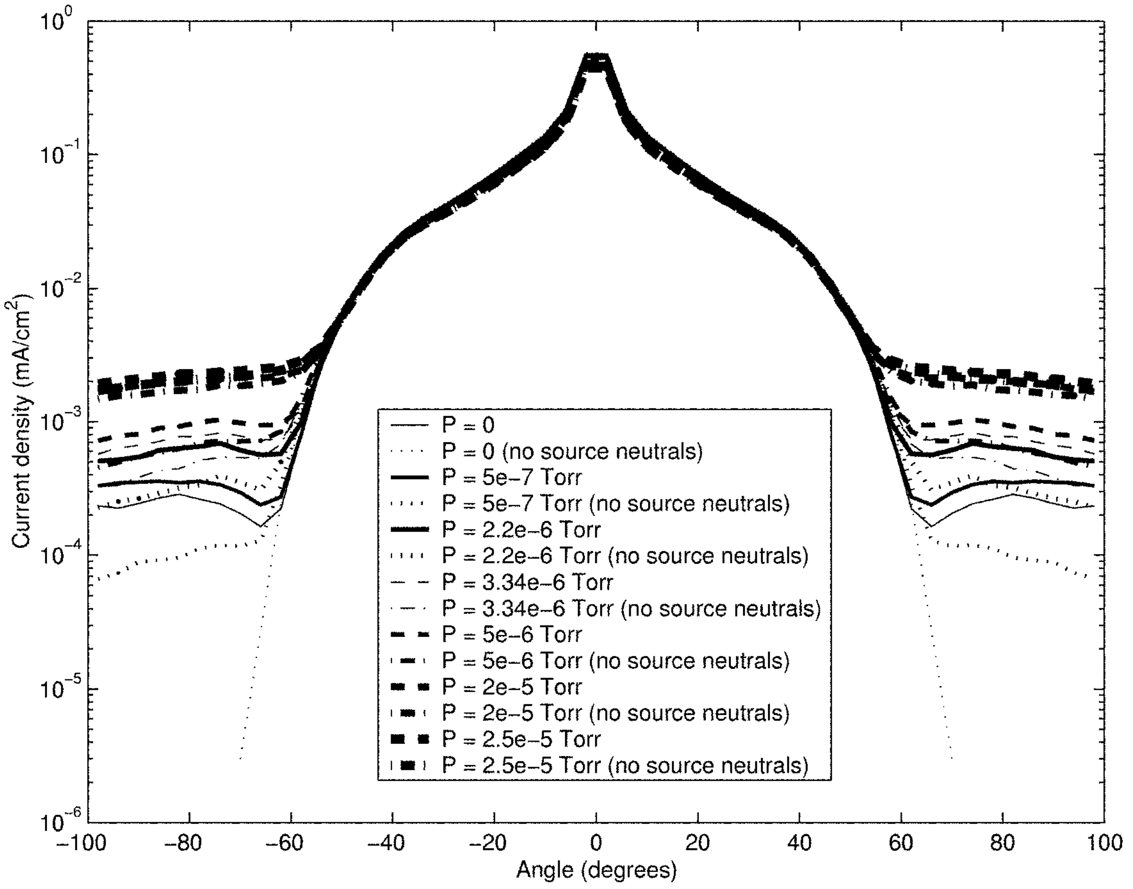

Figure 6-11: Current density distributions for various background pressures at 60 *cm* with inclusion and exclusion of source neutrals.

to actual in-space results is impossible. It is hypothesized that this critical pressure occurs when the corresponding background neutral density approximately equals the neutral density produced by the thruster in areas where CEX collisions primarily take place. For the BHT-200, a pressure of 5×10^{-6} *Torr* or below suffices. While low enough pressures may be possible to maintain for a small thruster such as the BHT-200, this may not be the case for a larger thruster. Since the source neutral density at the exit plane is smaller, maintaining an adequate vacuum pressure becomes increasingly difficult as the pumps need to work harder to both clear the larger amount of gas injected by the thruster and to maintain a low enough background neutral density that the source neutral density is not overpowered.

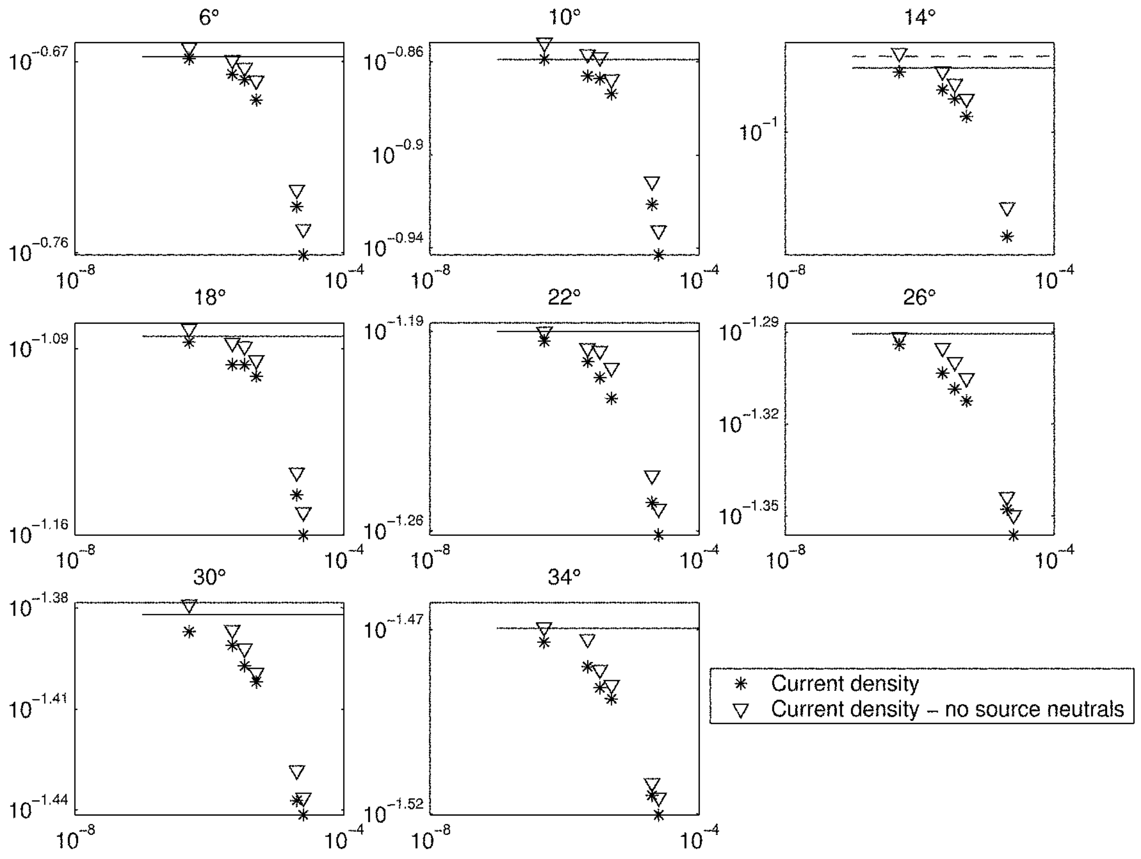

Figure 6-12: Current density $[mA/cm^2]$ versus background pressure $[Torr]$ at various angles. Solid line represents current density at P = 0 $Torr$. Dashed line represents current density without source neutrals at P = 0 $Torr$.

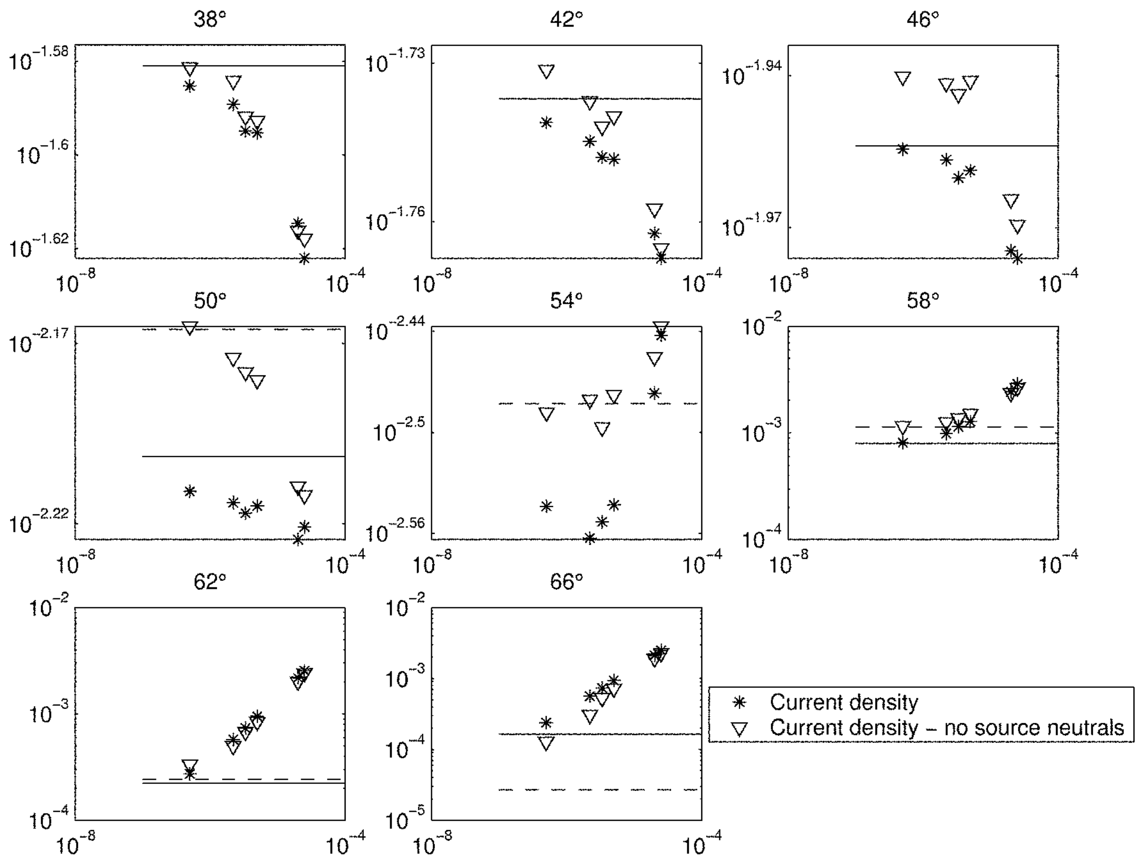

Figure 6-13: Current density $[mA/cm^2]$ versus background pressure $[Torr]$ at various angles. Solid line represents current density at P = 0 $Torr$. Dashed line represents current density without source neutrals at P = 0 $Torr$.

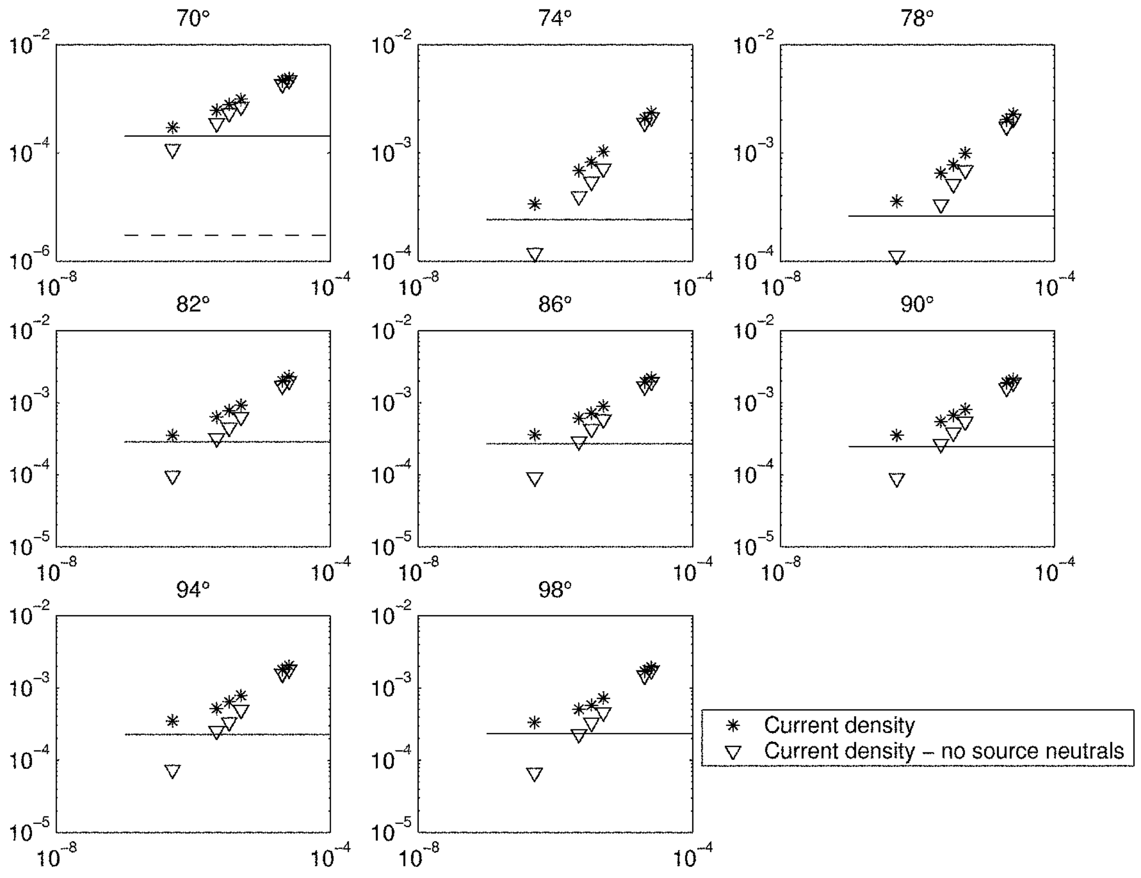

Figure 6-14: Current density $[mA/cm^2]$ versus background pressure $[Torr]$ at various angles. Solid line represents current density at P = 0 $Torr$. Dashed line represents current density without source neutrals at P = 0 $Torr$.

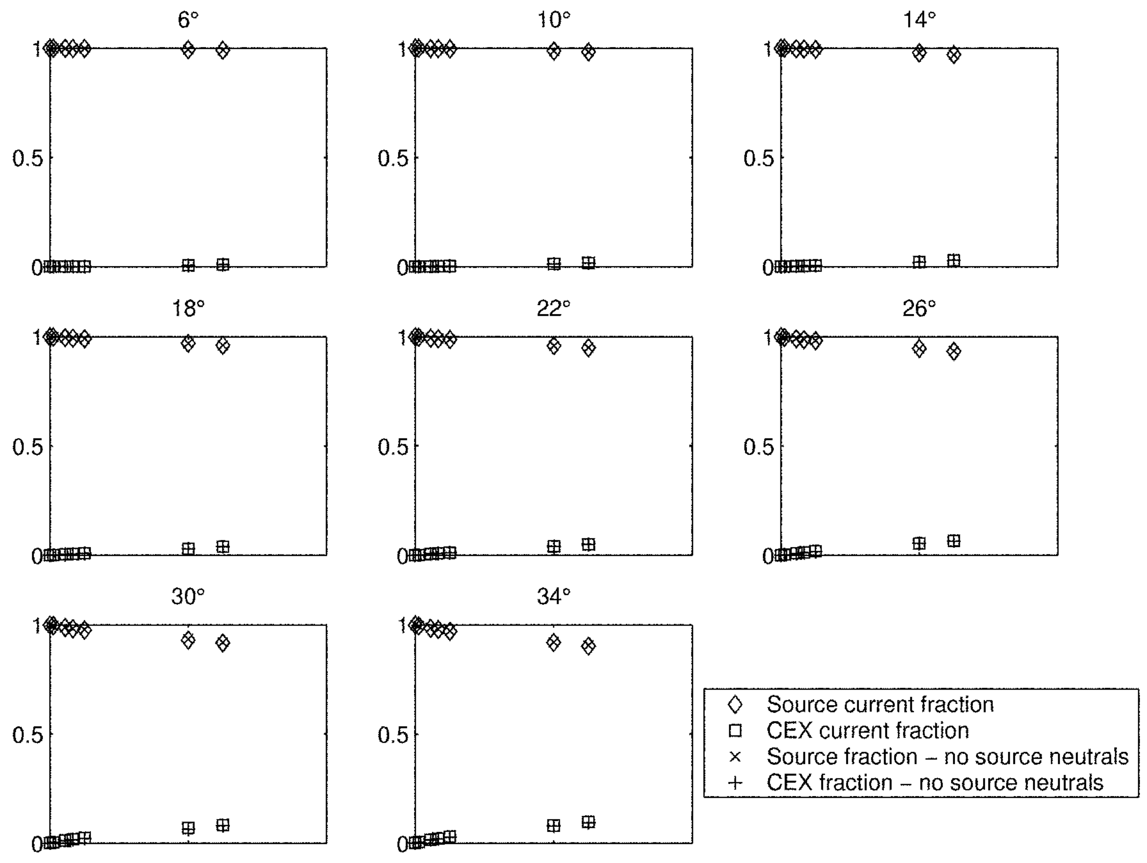

Figure 6-15: Fractional split of source and CEX ions versus background pressure [*Torr*] for various angles. Horizontal axis is pressure from 0 to $4{\times}10^{-5}$ *Torr*.

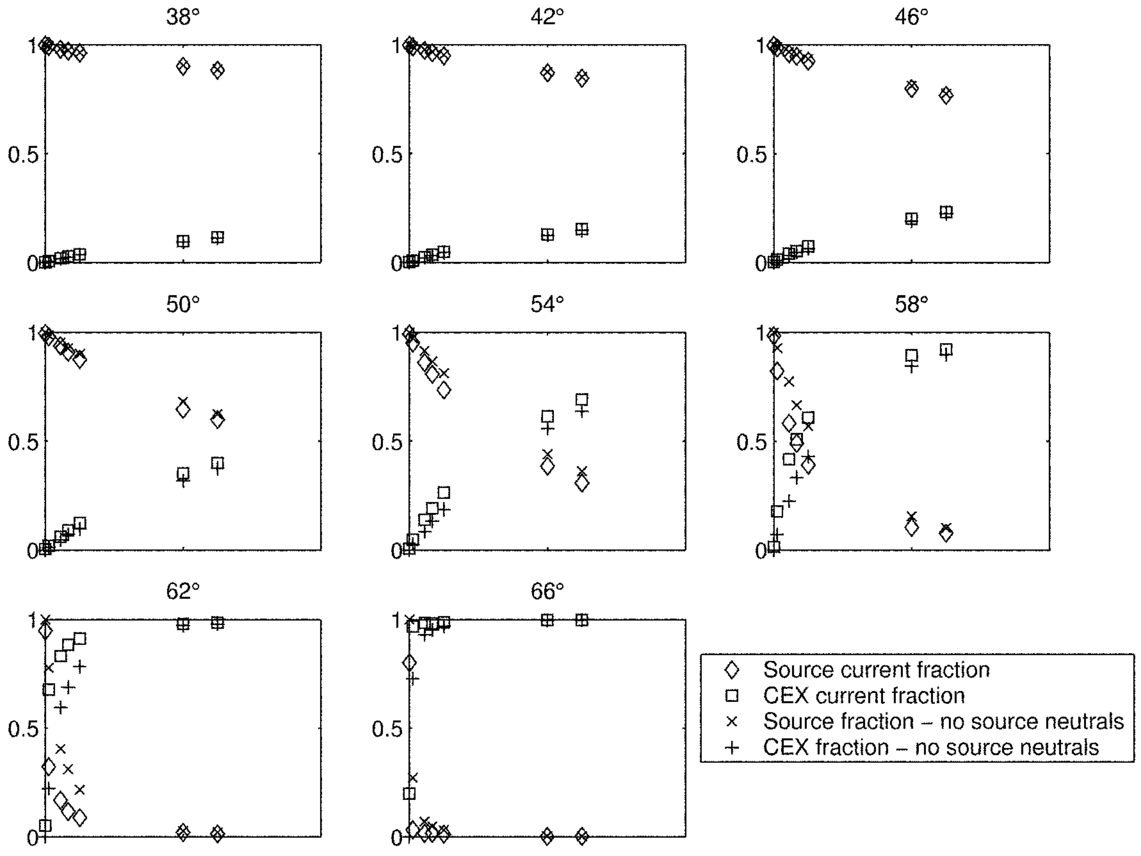

Figure 6-16: Fractional split of source and CEX ions versus background pressure [$Torr$] for various angles. Horizontal axis is pressure from 0 to 4×10^{-5} $Torr$.

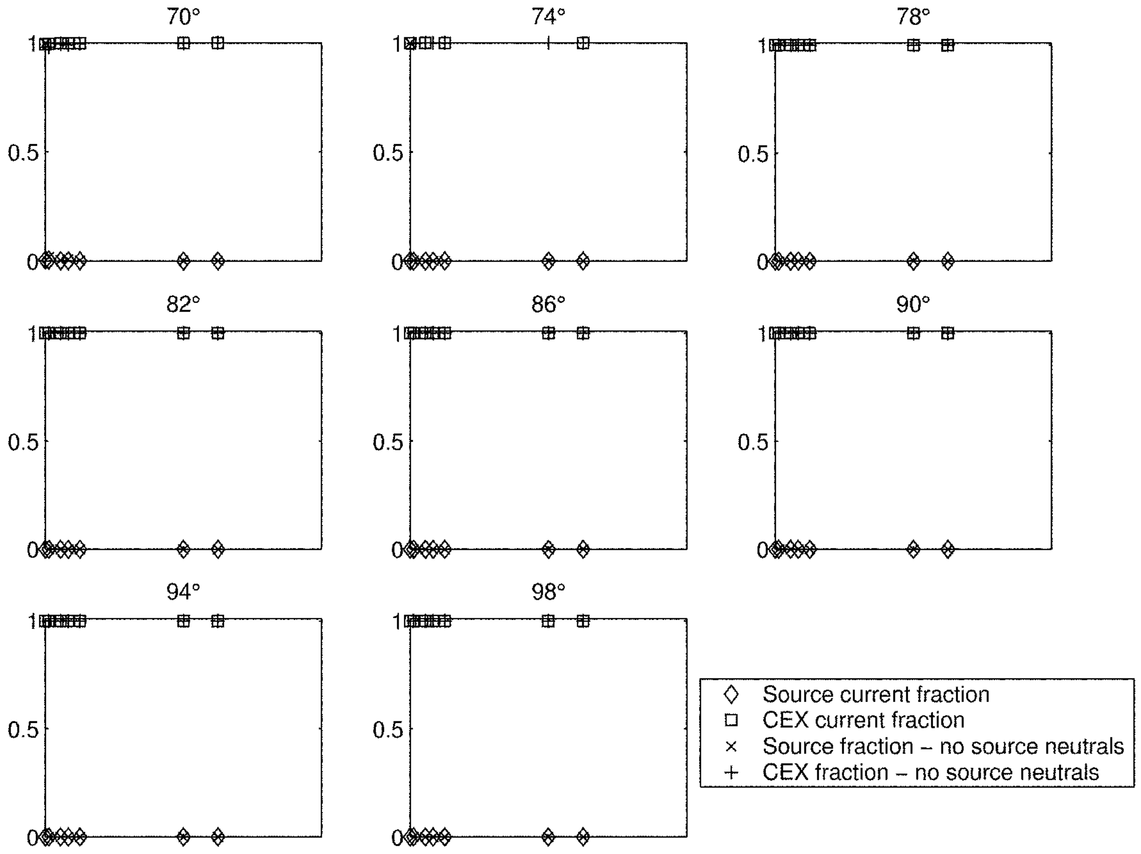

Figure 6-17: Fractional split of source and CEX ions versus background pressure [$Torr$] for various angles. Horizontal axis is pressure from 0 to 4×10^{-5} $Torr$.

6.3 Simulation in Vacuum Tank

In order to bring together all the modifications made to *Quasi3*, a simulation of the thruster in a vacuum tank is run. This run synthesizes the improvements made to the code's source model and sputtering methodology.

6.3.1 Simulation Geometry

The tank modeled is the vacuum facility located at MIT being used to prepare the ETEEV experiment. Figure 6-18 shows the grid used to simulate the tank. Two walls have been removed for easier viewing of the tank interior. The geometry is quite crude with respect to the actual tank which is cylindrical in shape and has a complex back wall designed to catch sputtered material. Because *Quasi3* is limited to rectangular geometries, such features could not be captured in the model. The geometry includes walls modeling the tank as a box, the thruster, two witness plates to study the sputtering and its deposition in the tank, and a hole in the center of the bottom to model the vacuum pumps. The hole is scaled by balancing mass flow injected by the thruster to mass flow leaving the tank,

$$\dot{m}_{thruster} = m_{Xe}\frac{n\bar{c}}{4}A. \tag{6.1}$$

The pressure the tank stabilizes to during engine operation is $2\times10^{-5}\ Torr$. Assuming room temperature of 300 K in the tank, the density and neutral thermal velocity can be calculated with,

$$n = \frac{P}{kT}, \tag{6.2}$$

$$\bar{c} = \sqrt{\frac{8}{\pi}\frac{kT}{m}}. \tag{6.3}$$

Equations 6.1 through 6.3 result in an expression for the area of the hole,

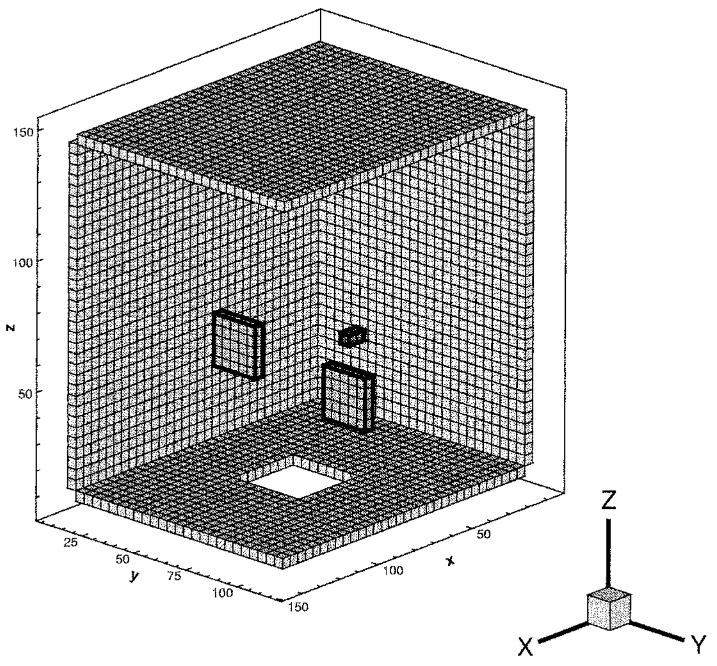

Figure 6-18: Simulated tank geometry.

$$A = \frac{\dot{m}}{P}\sqrt{\frac{2\pi kT}{m}}. \tag{6.4}$$

The two witness plates are modeled as silver and are oriented with respect to the thruster as shown in Figures 6-19 and 6-20. The tank walls are made of steel and hence are modeled as iron. Similar embedded meshes to those in Figure 6-2 are placed around the thruster exit plane. The potential reference point is placed at a node in the plume's path and walls in contact with the plasma are modeled as conductors as described in Section A.1. The tank simulation runs are 45,000 iterations.

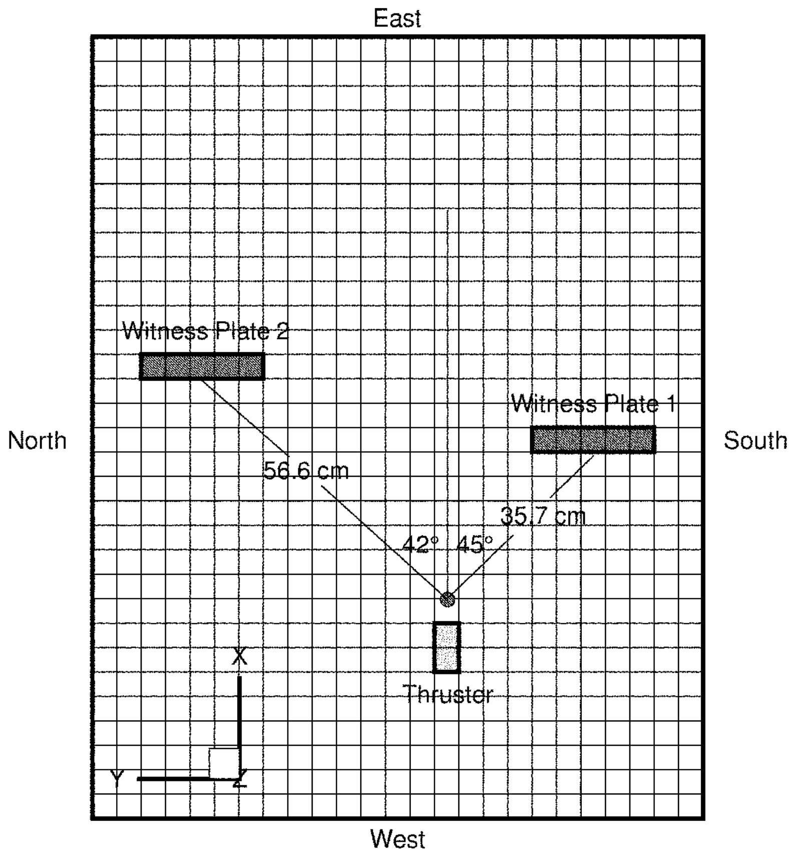

Figure 6-19: Witness plate orientation (top view).

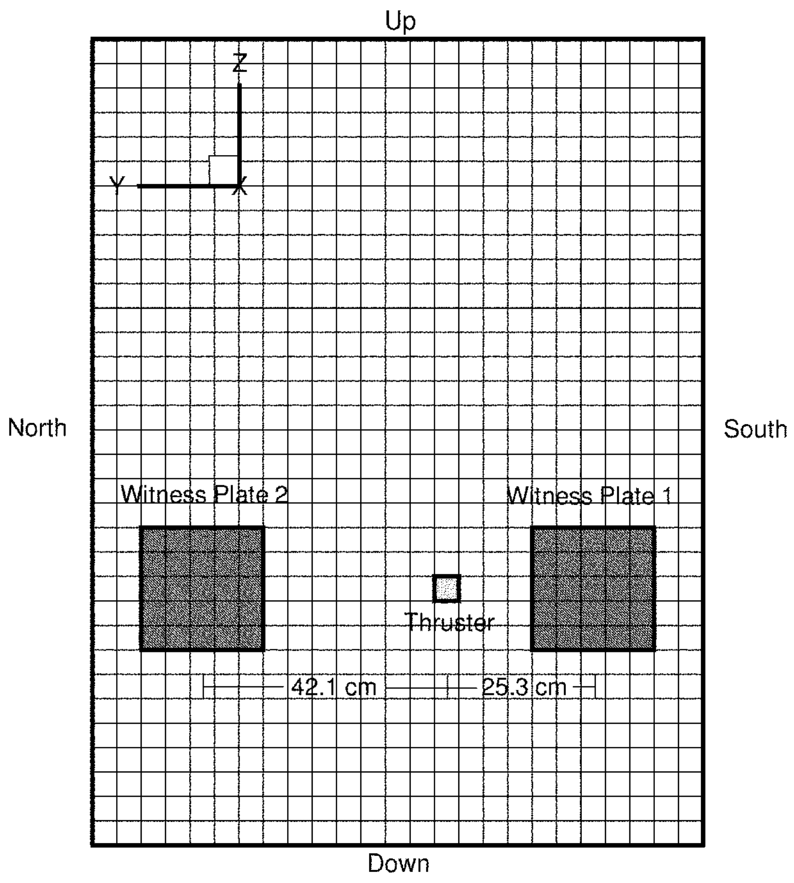

Figure 6-20: Witness plate orientation (back view).

6.3.2 Tank Simulation Results

Cathode Asymmetry

An interesting aspect of the 3D simulation is that the flow from the cathode is modeled separately from that of the main engine, so the plume is not completely symmetric. For this simulation, the cathode is modeled as being directed in the negative y direction. All particles emitted from the cathode are assumed to be neutrals (the cathode plume may in reality be 3-10% ionized). Figures 6-21 and 6-22 demonstrate this asymmetry of the source neutrals. Figure 6-21 shows them without any obstructions in the domain, while Figure 6-22 shows the plume neutrals with the simulated witness plates in their path. Shadowing in the particle distribution due to the witness plate obstruction can be observed.

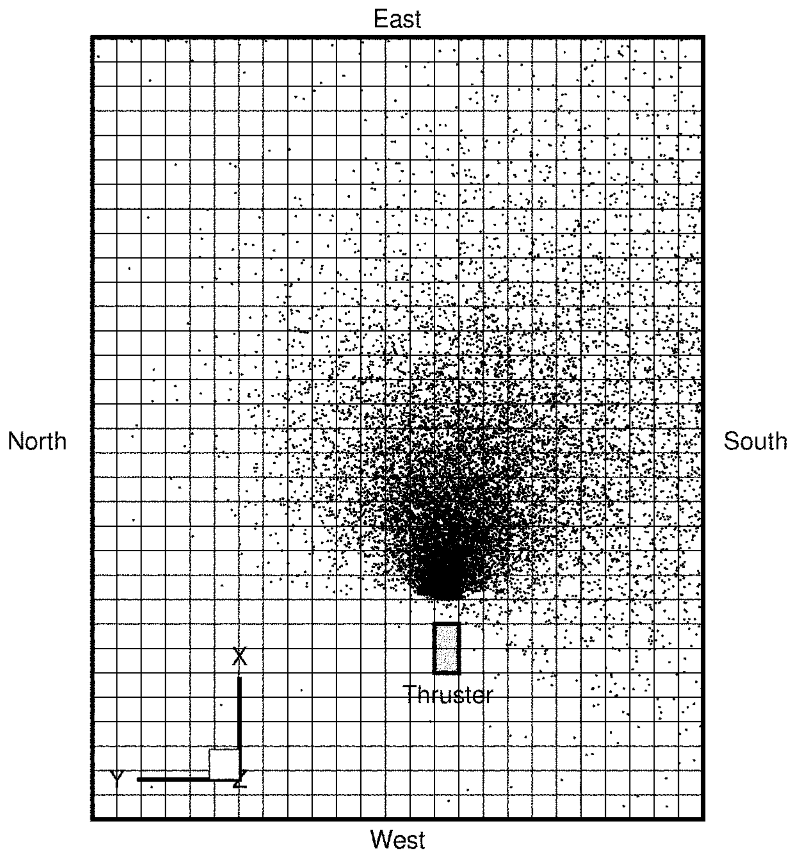

Figure 6-21: Xe particles after 1000 iterations.

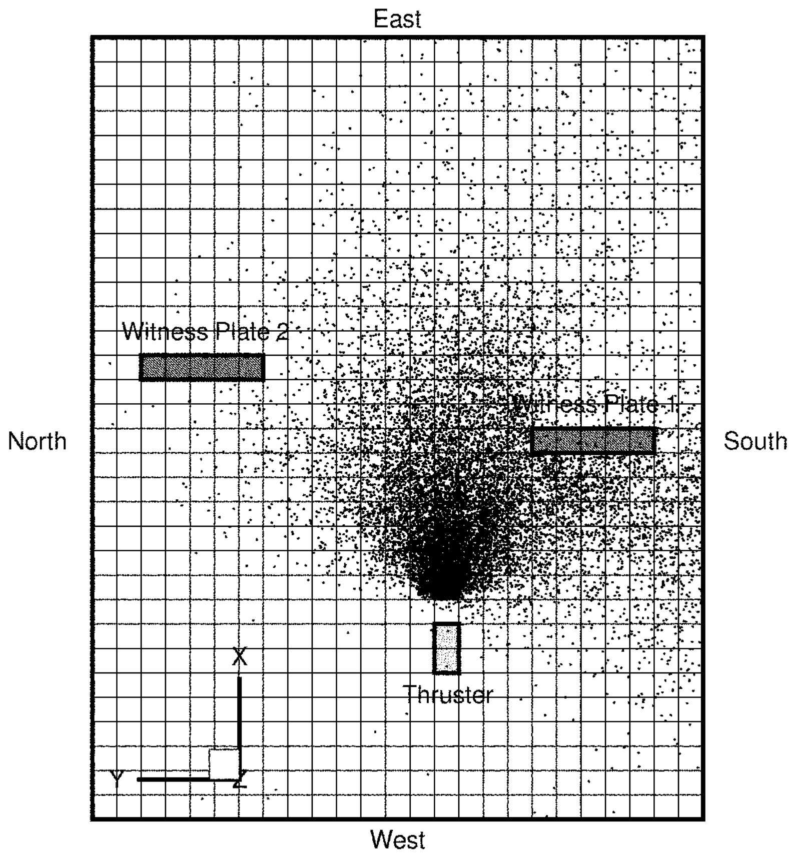

Figure 6-22: Xe particles after 1000 iterations.

Surface Interactions

Sputtering and deposition data are collected for all surfaces in the tank simulation as shown in Figures 6-23 through 6-32. It should be noted that these results are from runs with a minor error in the calculation of sputtered particle outgoing angles. An extra +1 term is included inside the curly braces of Equation 5.21. As illustrated in Figures 6-23 and 6-25, silver sputters from the west walls of the witness plates in large amounts as these are the sides directly exposed to the thruster plume. As expected, this material from the witness plates accumulates mostly on the west wall behind the thruster and on the north and south walls west of the plates (Figures 6-28 and 6-30). Figures 6-23 and 6-26 show no silver deposition on the west and east walls of witness plates 1 and 2 respectively, accounting for the fact that these surfaces have no line of sight from the other plate. Another point to note is that both the east wall of witness plate 1 and the west wall of witness plate 2 receive a finite amount of silver deposition. This phenomenon is explained because the witness plates have a finite width in the simulation. Hence, the north and south walls of the witness plates are capable of sputtering and subsequently depositing on the other plate.

Iron sputtering mostly occurs in the east side of the tank since the plume is aimed in this direction (Figures 6-27 through 6-29). In Figure 6-28, the shielding of the north wall from sputtering due to obstruction by witness plate 2 can be seen. Deposition of iron occurs throughout the tank, primarily on the east faces of the witness plates and the eastern parts of the tank walls. Since iron sputtering is predominantly located in this region, it follows that its deposition is as well.

Figure 6-32 depicts the flux of different types of Xe^+ on the tank's west wall. Throughout the course of the simulation, when charge exchange collisions occur, those particles involved are labeled as "CEX particles," allowing separation and comprehension of the differences between particles that have undergone the process and those that have not. It is clear that the ion flux striking the west wall is primarily due to CEX ions — to reach this surface, ions must have a velocity opposite to the thrust direction which most likely is the result of the CEX process. The occurrence of source ion flux on this wall is probably due to large angle elastic scattering of source ions — particles undergoing elastic collisions do not get relabeled and as such are grouped with the source ion population. It can be seen from Figures 6-31 and 6-32 that CEX ions generally strike walls close to the locale of their formation (typically around the engine exit plane). The side "wings" formed by the CEX ions which are pushed laterally by the plume's potential hump can also be observed.

Runs with an initial background density produce basically the same surface interaction data, but fluxes are altered by the introduction of background neutrals as shown in Figures 6-33 and 6-34. CEX ion flux is still the major contributor to the west wall and is also increased by an order of magnitude over the vacuum case. However, the shape of their distribution corresponding to the plume's potential profile is lost. The presence of the neutral background results in CEX collisions happening throughout the tank and not only near the thruster exit plane, where the source neutrals are localized. The preferential impingement of the upper part of the west wall occurs because the thruster is closer to the tank's bottom. Thus, CEX ions produced below the thruster are more likely to deposit on the bottom wall before having the opportunity to reach the west wall. The opposite is true for the upper region of the tank, explaining the focusing of CEX ions on the upper west wall. Remnants of the former CEX ion distribution can still be seen in Figure 6-33, but it is much broader as explained by the dispersed production of CEX ions.

Overall, the surface interaction results of the tank simulations seem to make sense qualitatively, giving confidence in the general sputtering methodology.

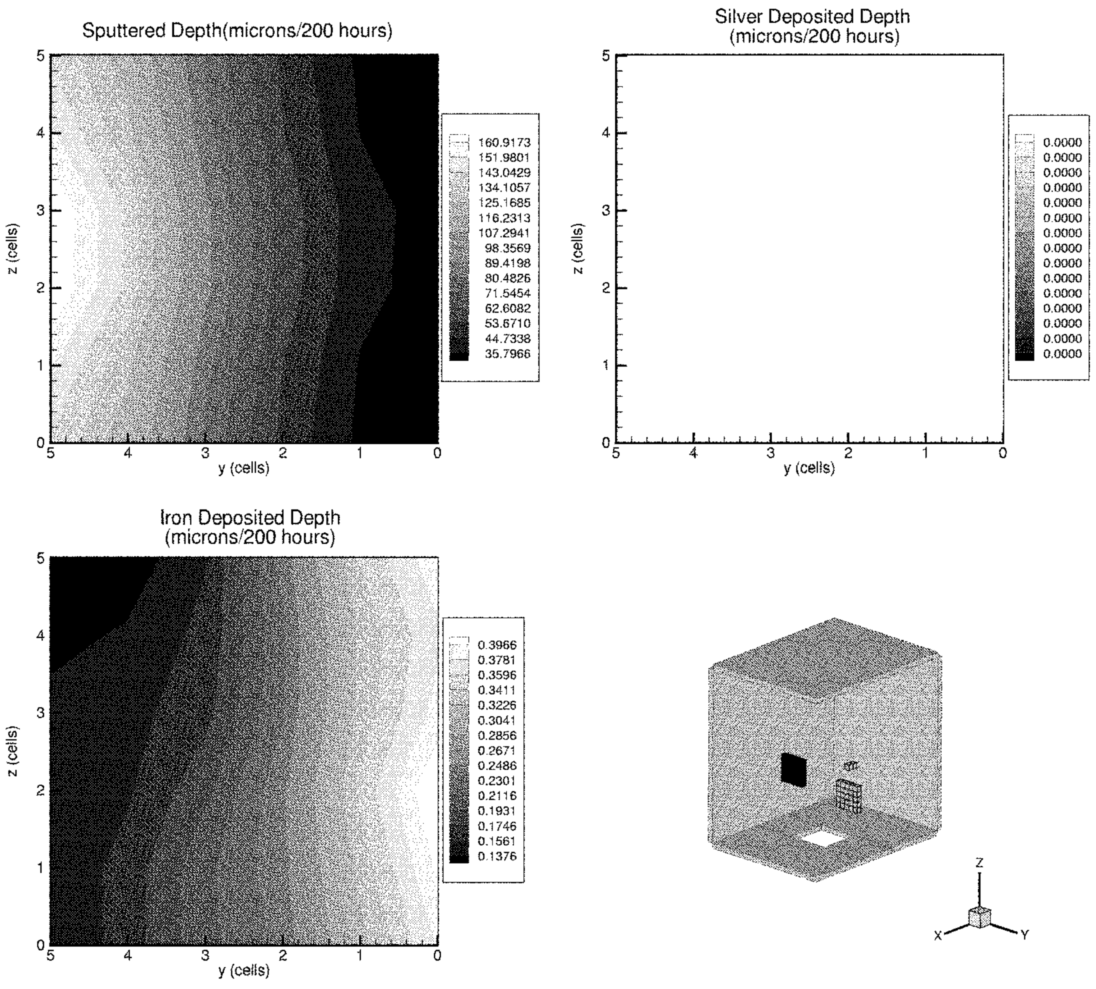

Figure 6-23: Witness plate 1, west wall, silver material.

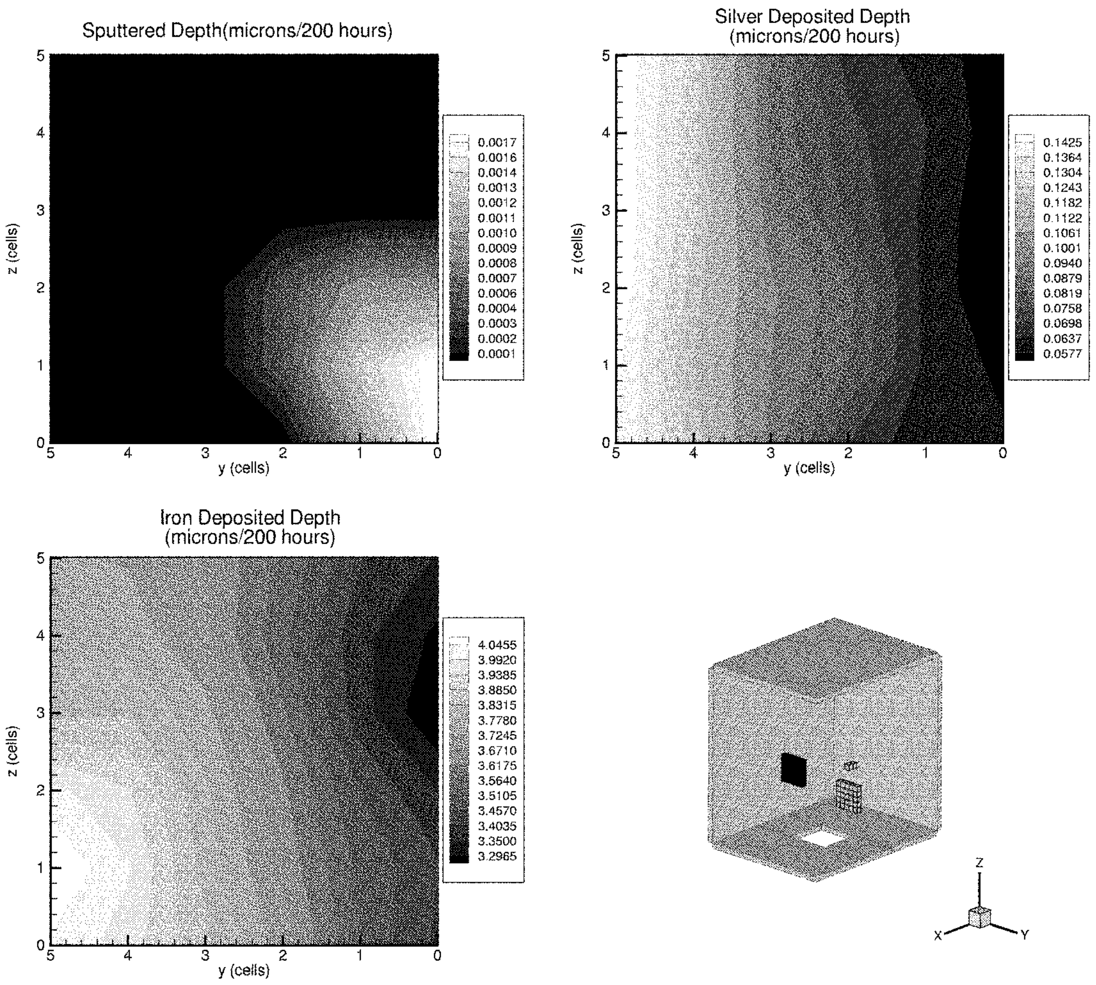

Figure 6-24: Witness plate 1, east wall, silver material.

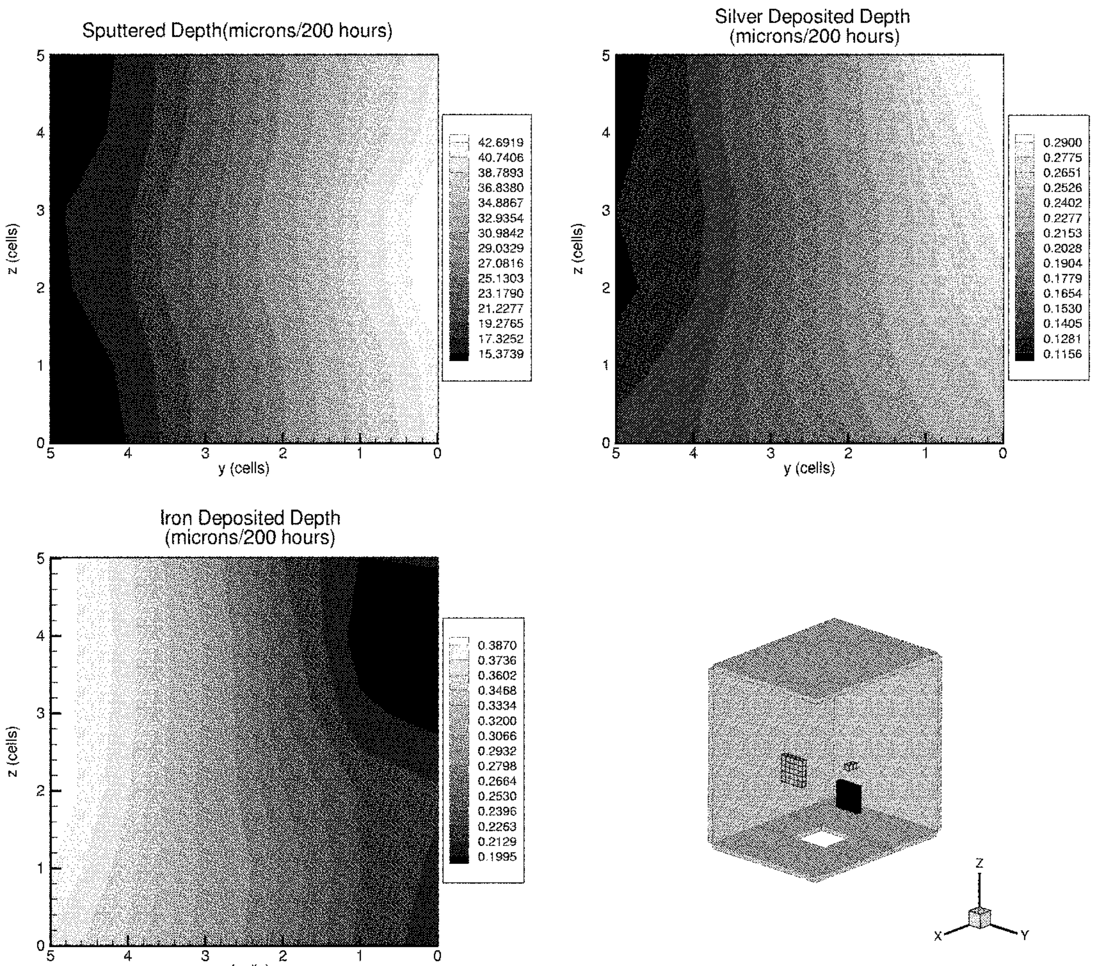

Figure 6-25: Witness plate 2, west wall, silver material.

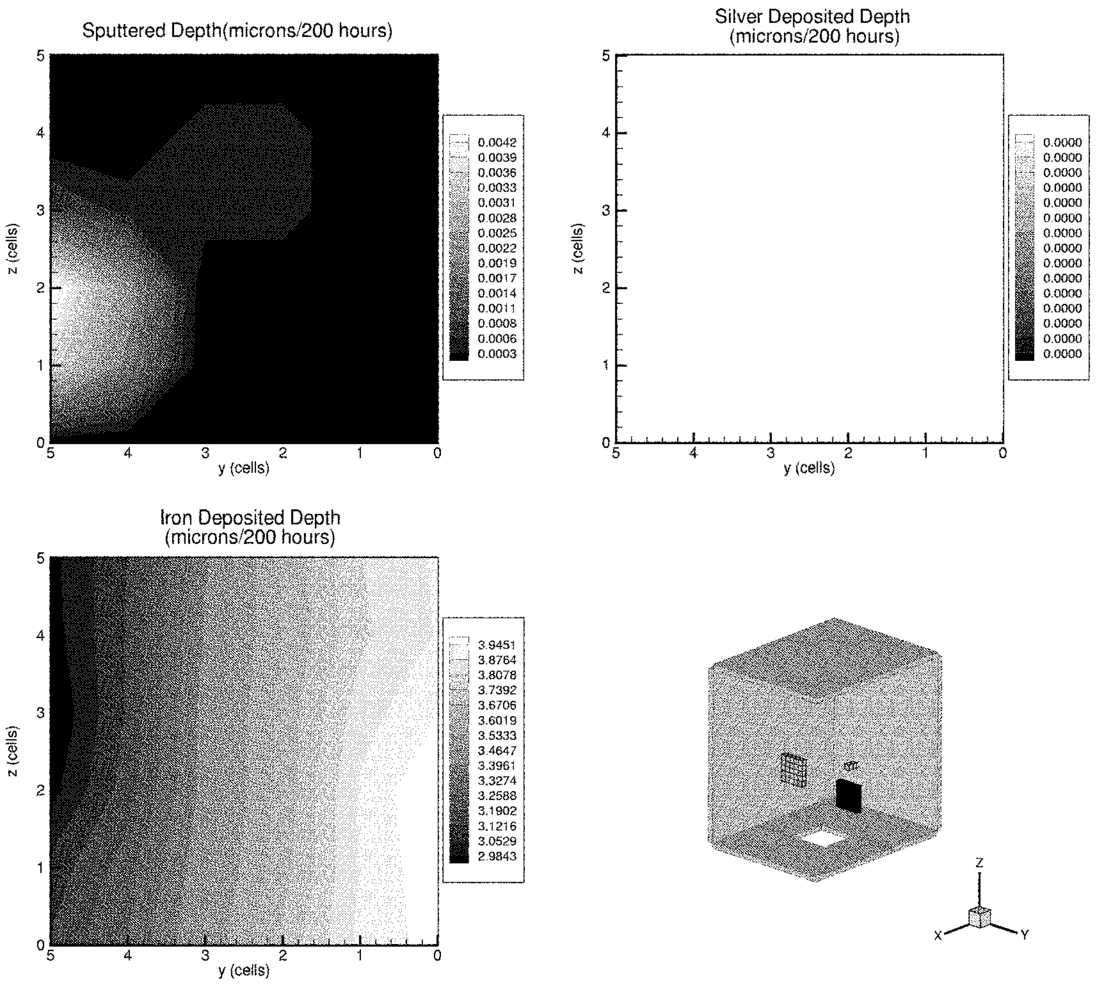

Figure 6-26: Witness plate 2, east wall, silver material.

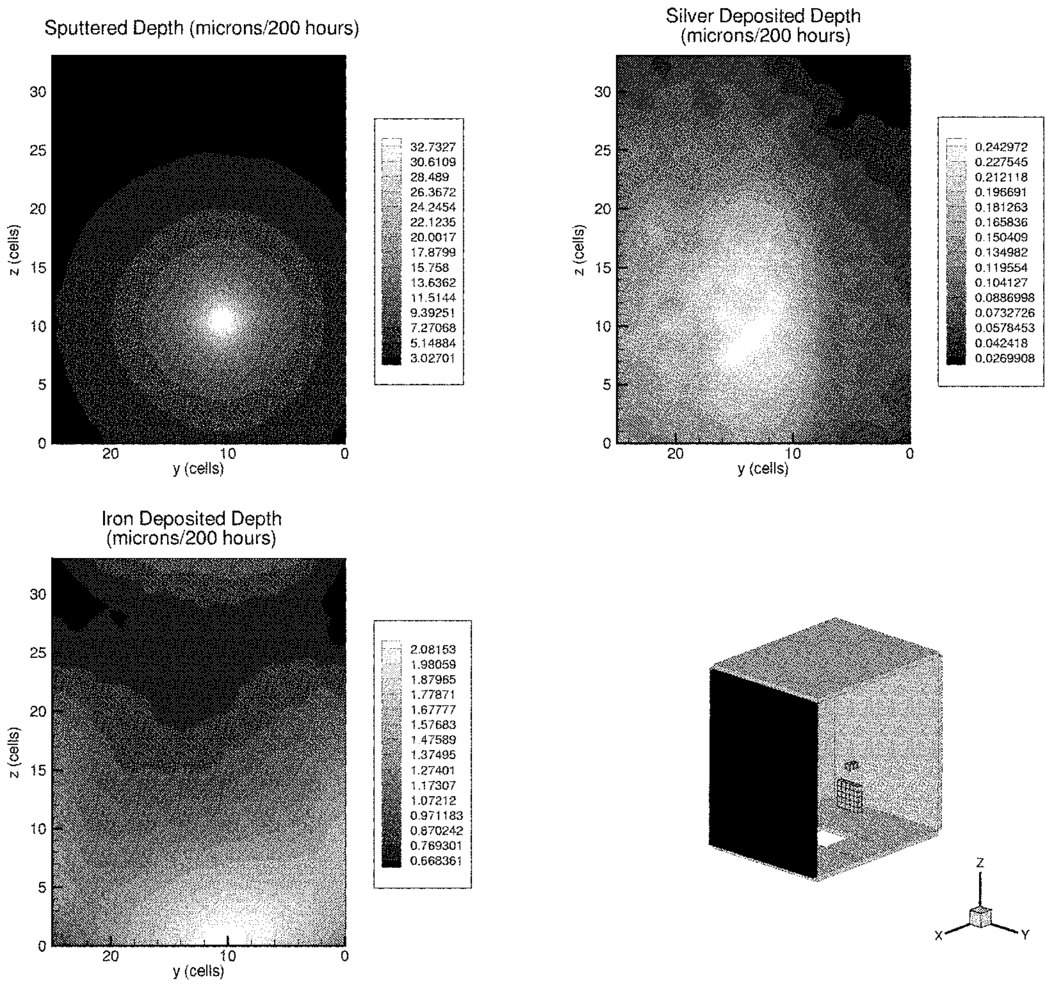

Figure 6-27: East wall, iron material.

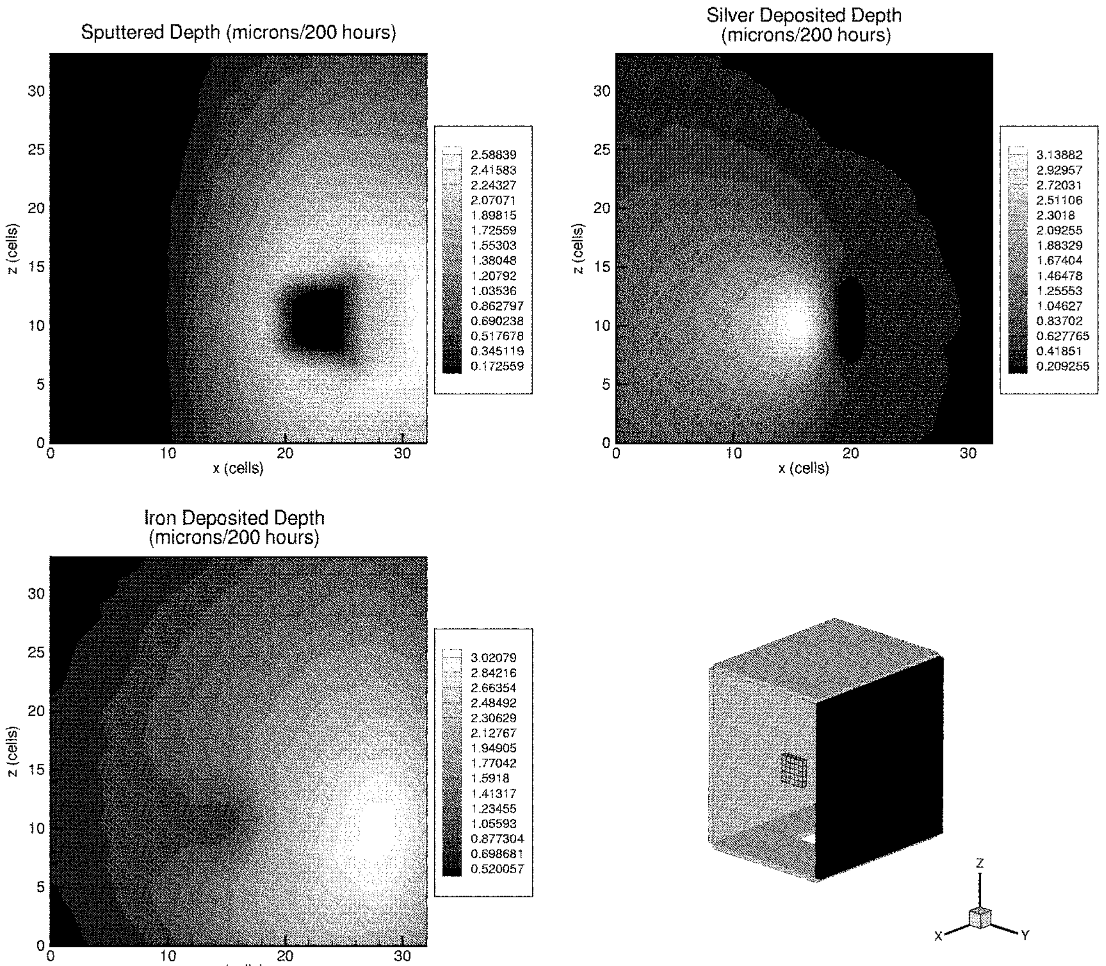

Figure 6-28: North wall, iron material.

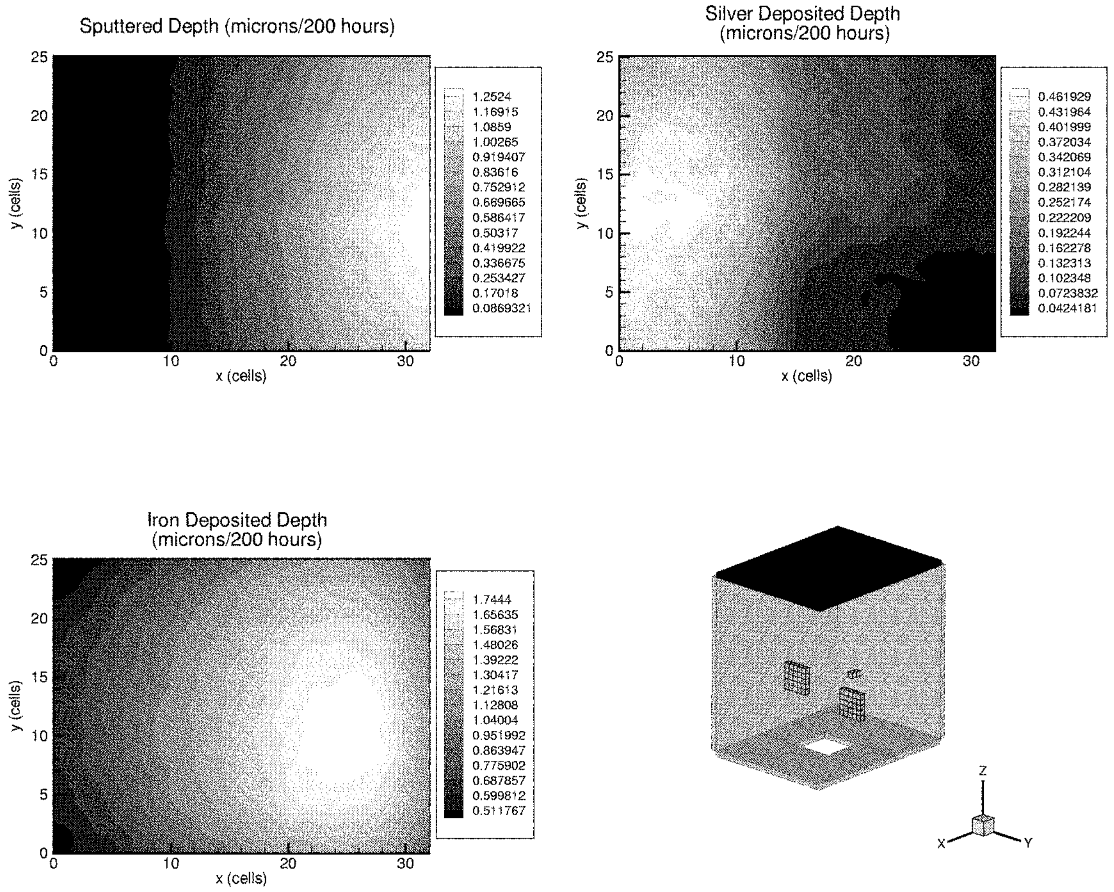

Figure 6-29: Up wall, iron material.

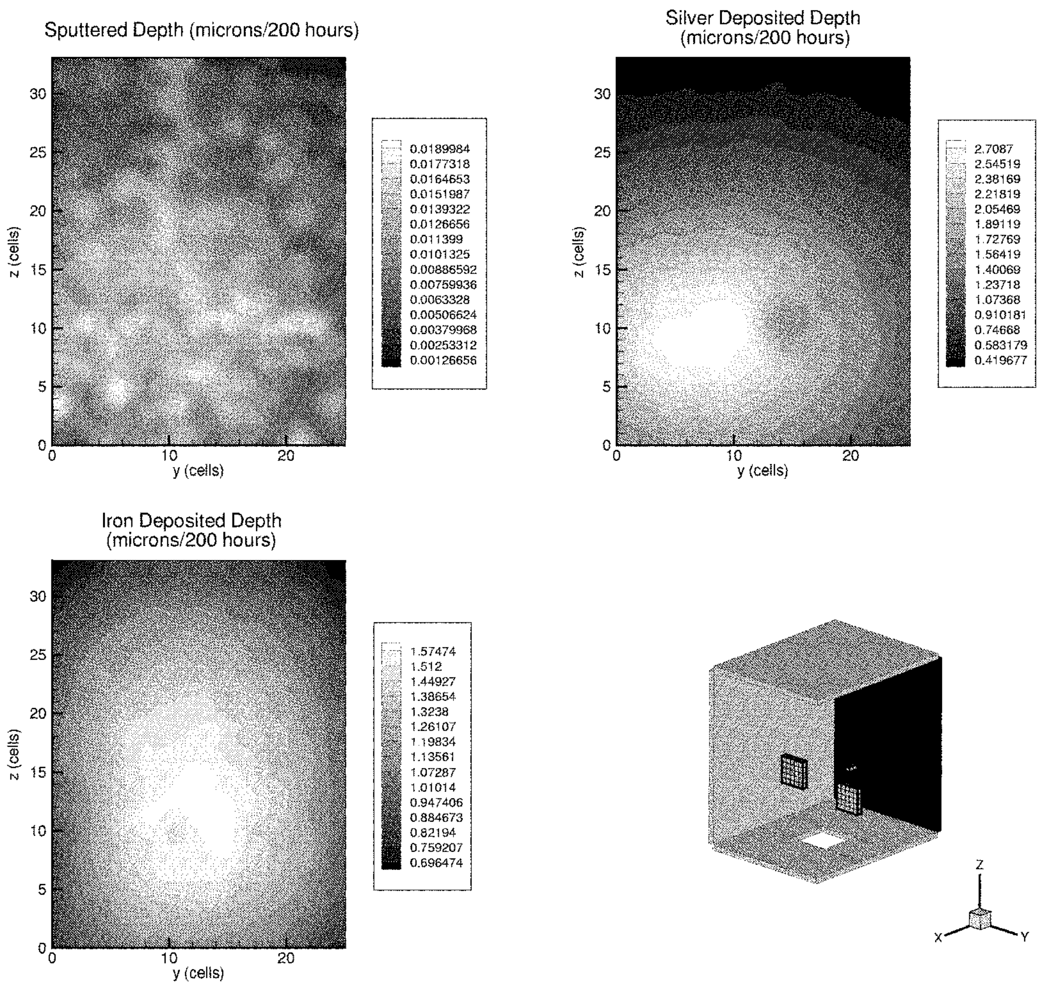

Figure 6-30: West wall, iron material.

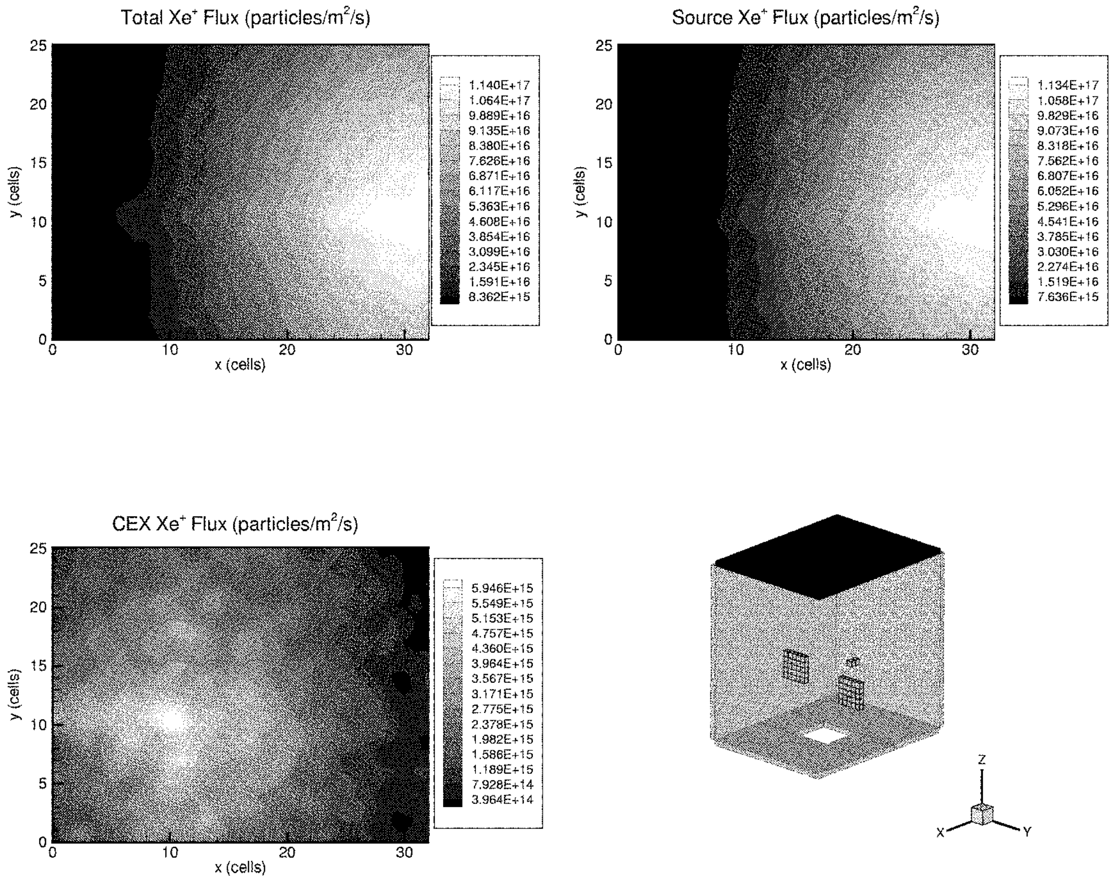

Figure 6-31: Up wall, iron material.

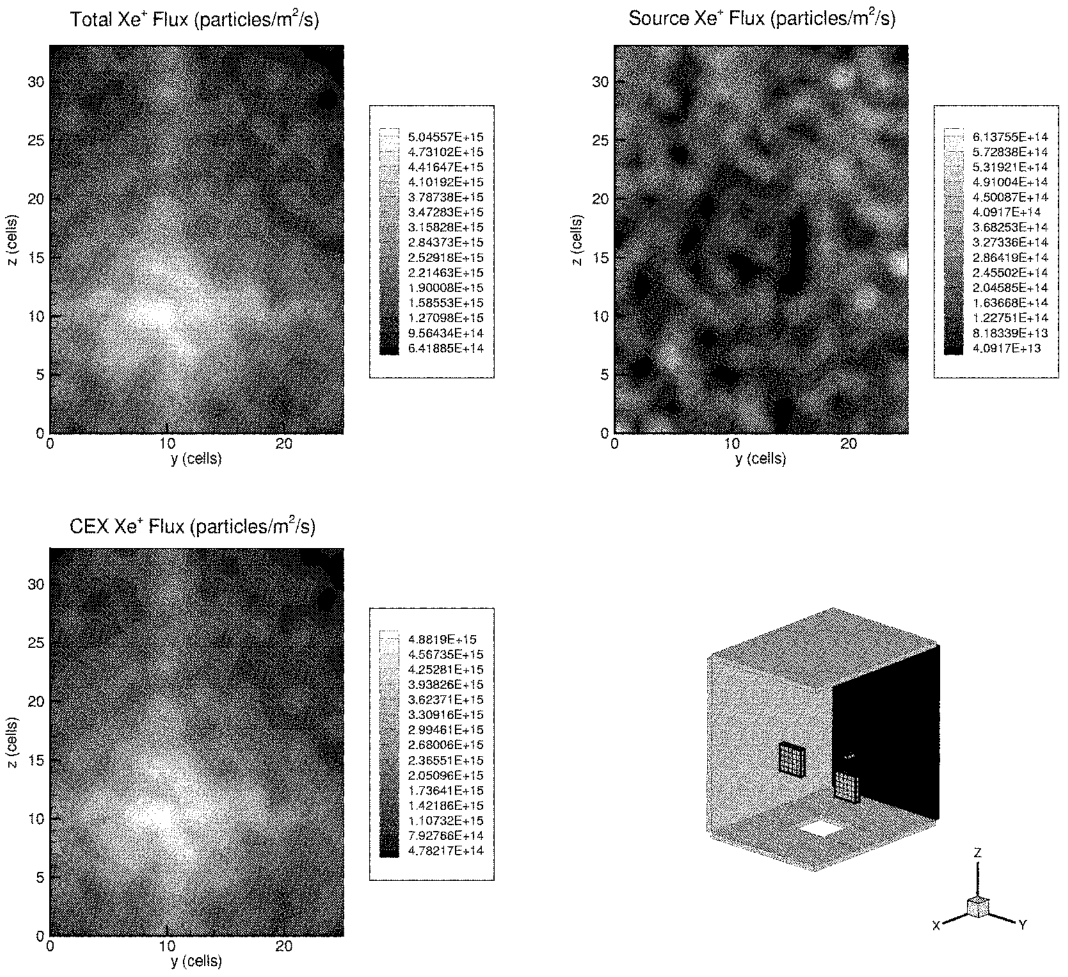

Figure 6-32: West wall, iron material.

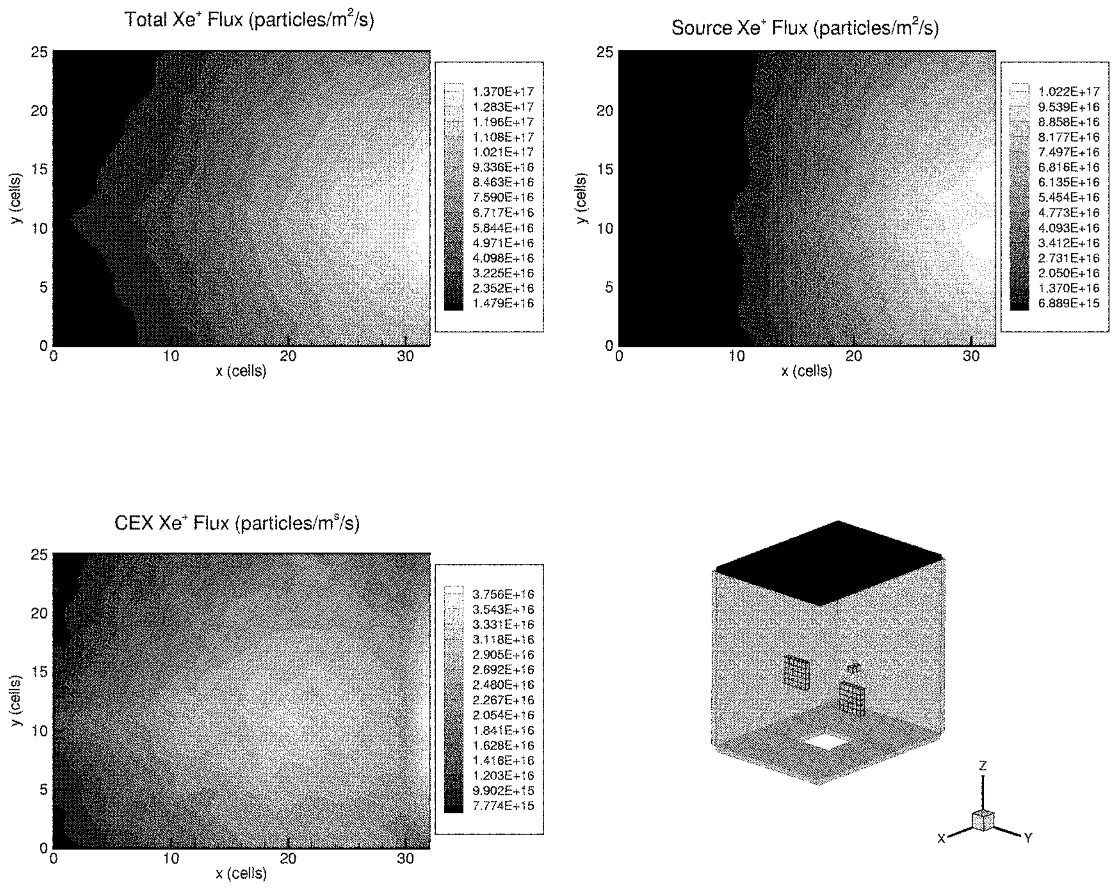

Figure 6-33: Up wall with background density, iron material.

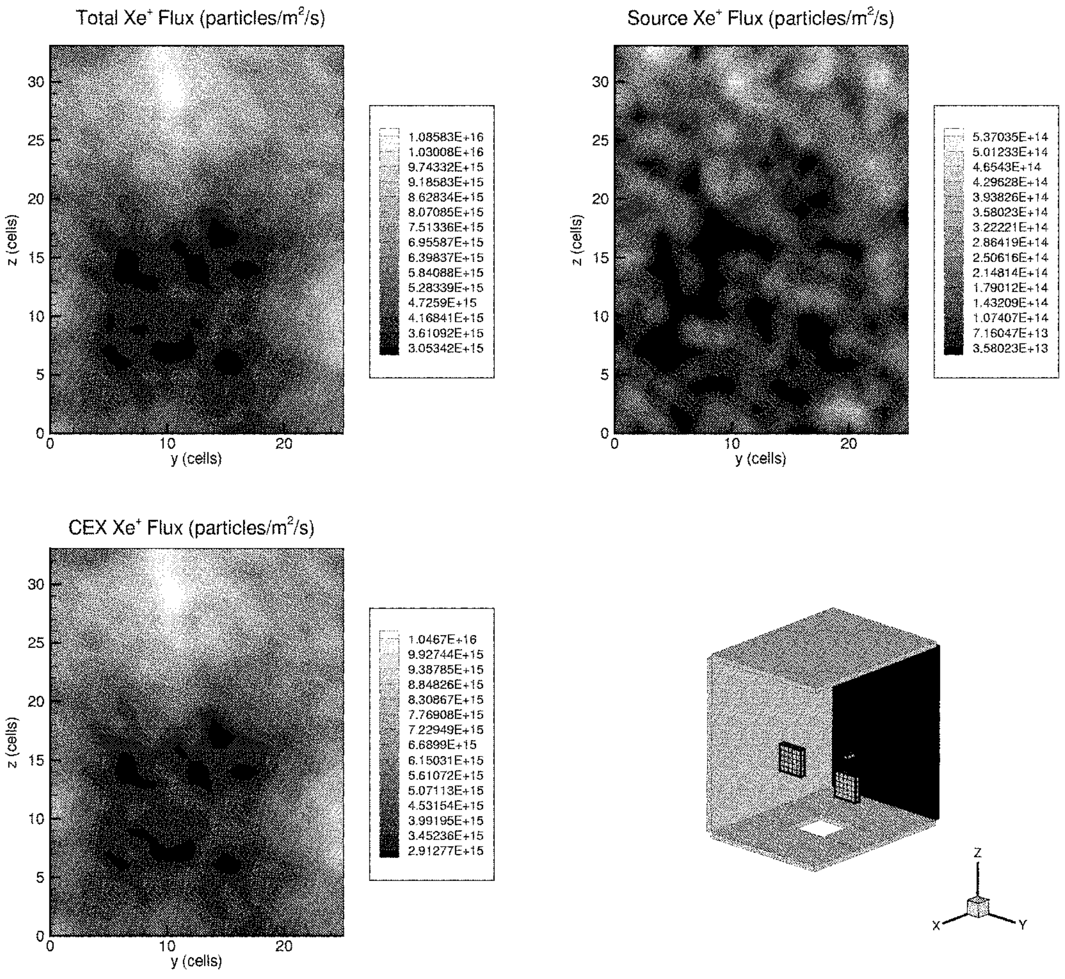

Figure 6-34: West wall with background density, iron material.

6.3.3 Tank Filling

When simulating the tank filling process, no initial background density is assumed, so neutrals only originate from the source model. The code is run until particles leaving are balanced by particles entering, resulting in a steady-state total. Because the number of neutrals dominates, a very large neutral weighting needs to be used to maintain a reasonable computation rate and memory usage. Figure 6-35 shows the convergence of a tank filling case run over 11 days. The ion species reach their steady-state totals almost immediately whereas the neutrals take much longer. Temperatures of each species are calculated from particle information saved every 250,000 iterations and plotted in Figure 6-36. The neutral temperature should asymptote to 300 K since accommodation to the tank wall temperature occurs after many collisions. However, the simulated temperature seems to be decreasing below the predicted value. In addition, the final pressure of 1.4×10^{-5} $Torr$ as seen in Figure 6-37 is below the expected 2×10^{-5} $Torr$. Although there are issues with the final temperature and pressure values of the tank filling simulation, it is promising that the number of neutrals converges, so further improvements to the model may result in completion of tank simulation runs over the course of a few days.

6.4 Sources of Error

Although improvements to *Quasi3* have been made, issues with the simulation still remain, largely evidenced by the mismatching of simulated and experimental current density data. Several probable sources of error are outlined below.

6.4.1 Charge Exchange

As seen in Figure 6-4, the current density in the wings of the plume is under-predicted. A possible reason for this is that CEX ions generated inside the engine before the exit plane are not currently modeled. These slow-moving ions could easily be pushed into the wings by the potential hump of the plume, thus filling out the current density.

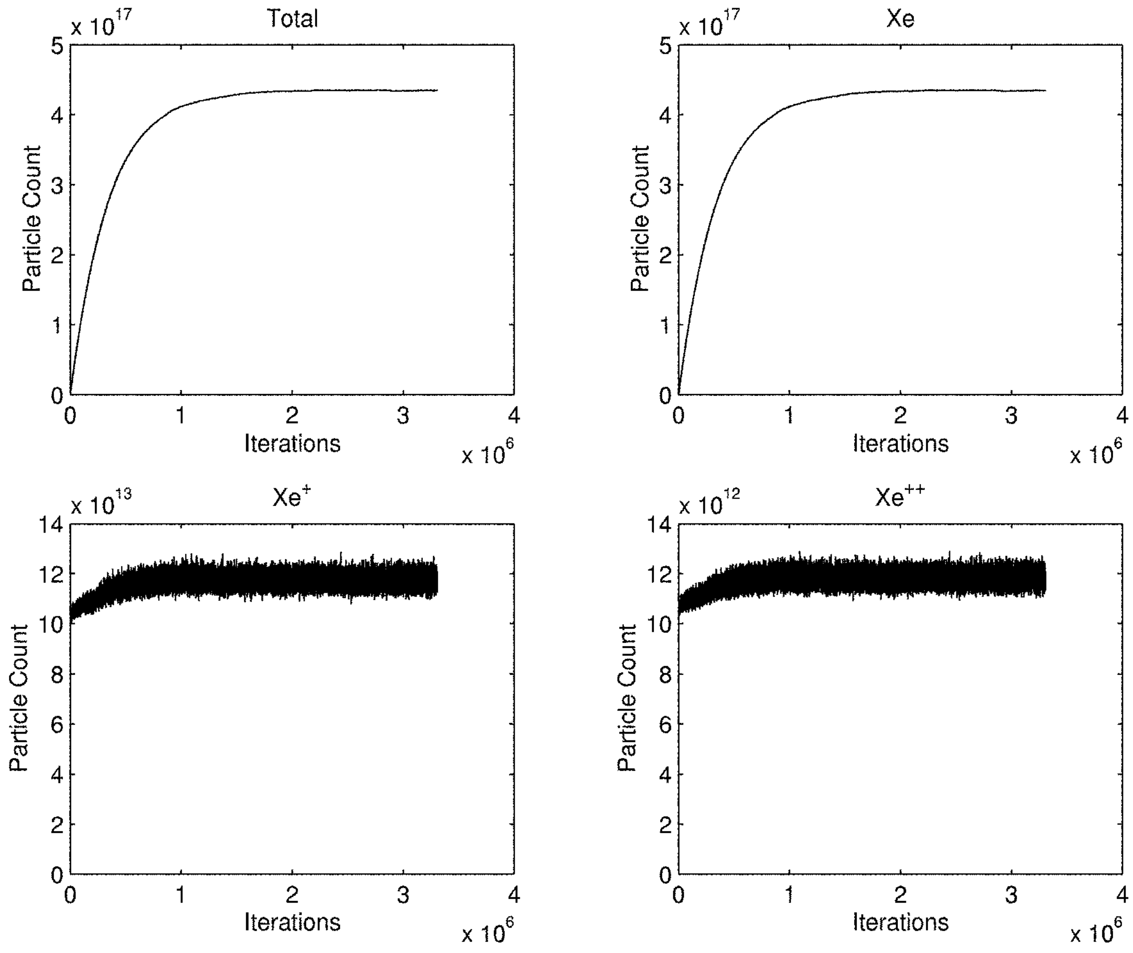

Figure 6-35: Particle totals for tank filling case.

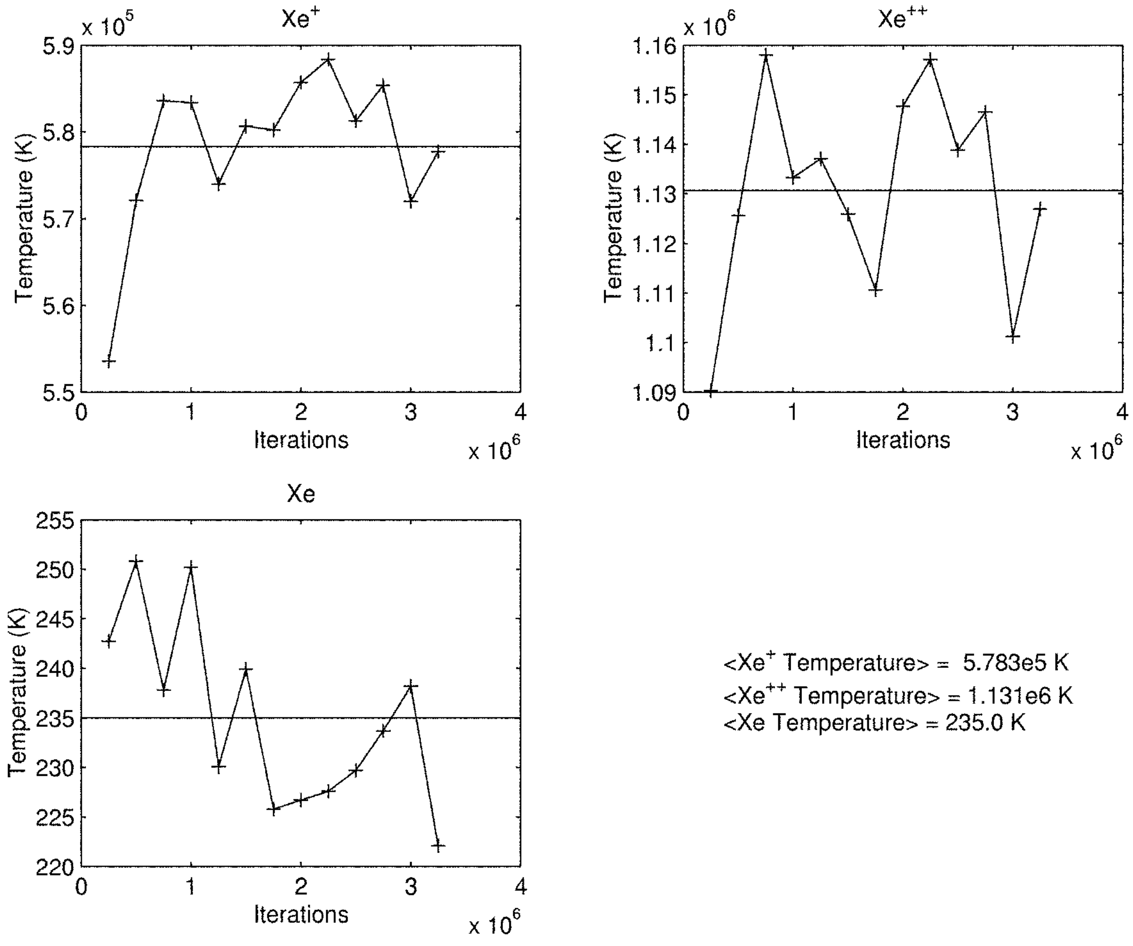

Figure 6-36: Species temperatures for tank filling case.

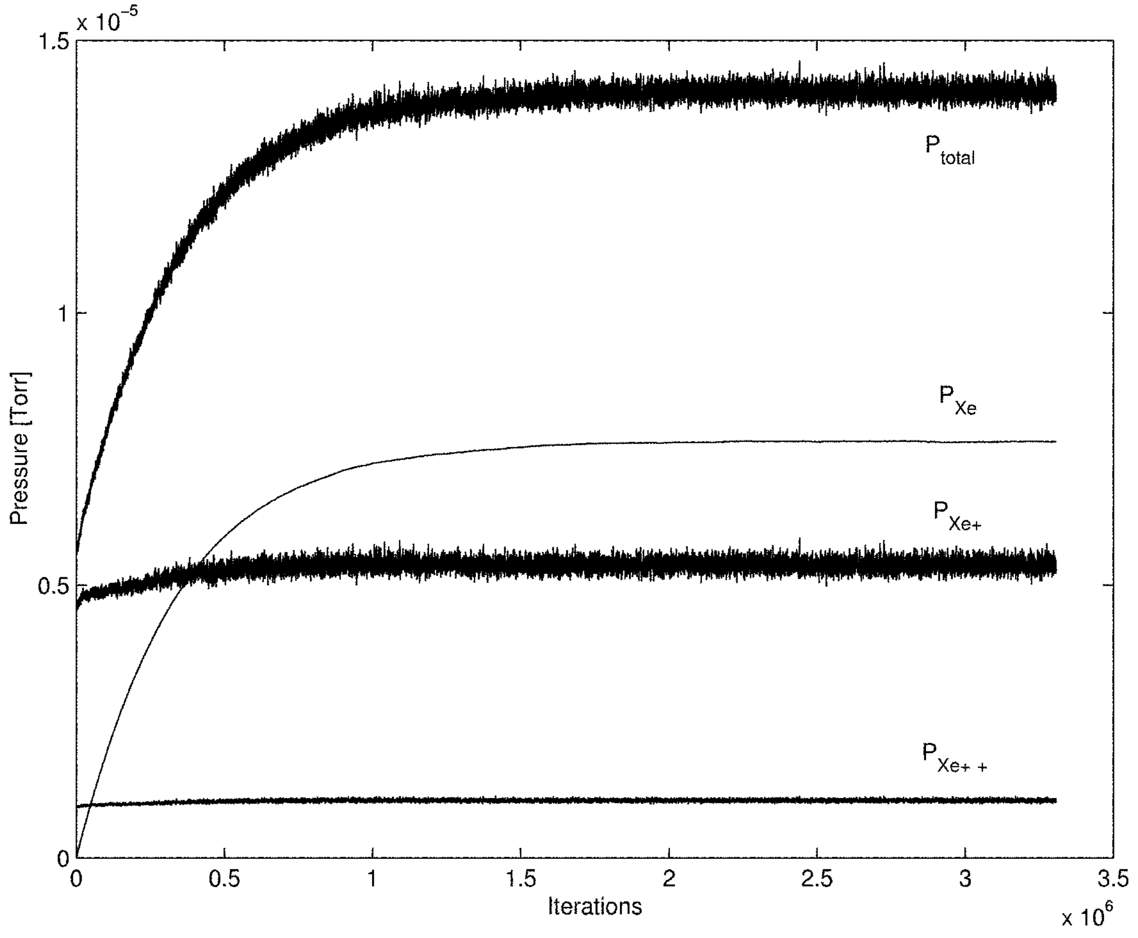

Figure 6-37: Species pressures for tank filling case.

Previously, Brenizer [3] added CEX collisions into *Hall* and found that although the overall performance parameters of the thruster were unaffected, the potential at the exit plane was elevated due to their inclusion. It was hypothesized that the slower-moving CEX ions are retained longer within the engine, thus increasing the potential. Unfortunately, Brenizer's modifications were made to a previous version of *Hall* which did not include double ions, so CEX ions from the engine are not included in the current source model. However, post-processing of data from *HPHall* is done to estimate the amount of CEX missing from the model. The rate at which CEX collisions occur is calculated with,

$$\dot{n}_{CEX} = n_i n_n c_r \sigma_{CEX}, \tag{6.5}$$

where n_i and n_n are the densities of the colliding species, c_r is their relative velocity and σ_{CEX} is the cross-section of the charge exchange collision of interest. The single ion-neutral and double ion-neutral CEX cross-sections are given by,

$$\begin{aligned}
\sigma_{Xe-Xe^{+}CEX} &= [1.81 \times 10^{-14} - 2.12 \times 10^{-15} \log c_r](I/I_o)^{-1.5}\, cm^2, && (6.6)\\
\sigma_{Xe-Xe^{++}CEX} &= (3.4069 \times 10^{-9} - 2.7038 \times 10^{-10} \ln c_r)^2\, m^2. && (6.7)
\end{aligned}$$

Using time-averaged data from *HPHall* for the BHT-200 and Equations 6.5 through 6.7, the rates for both types of CEX collisions are post-calculated. Figure 6-38 shows the single ion-neutral CEX collision rate. Figure 6-39 shows the double ion-neutral CEX collision rate.

While most of the CEX production occurs in the near-anode engine region, these ions are not a major threat since they are formed far enough inside the channel that they are accelerated through a high potential and have gained enough energy before the exit plane that they can be grouped with the normal source ions. On the other hand, the CEX ions produced near the exit plane, though fewer in number, are the most dangerous as they fall through a reduced value of potential and leave the thruster with comparatively slower velocities. In order to estimate the amount of missing CEX

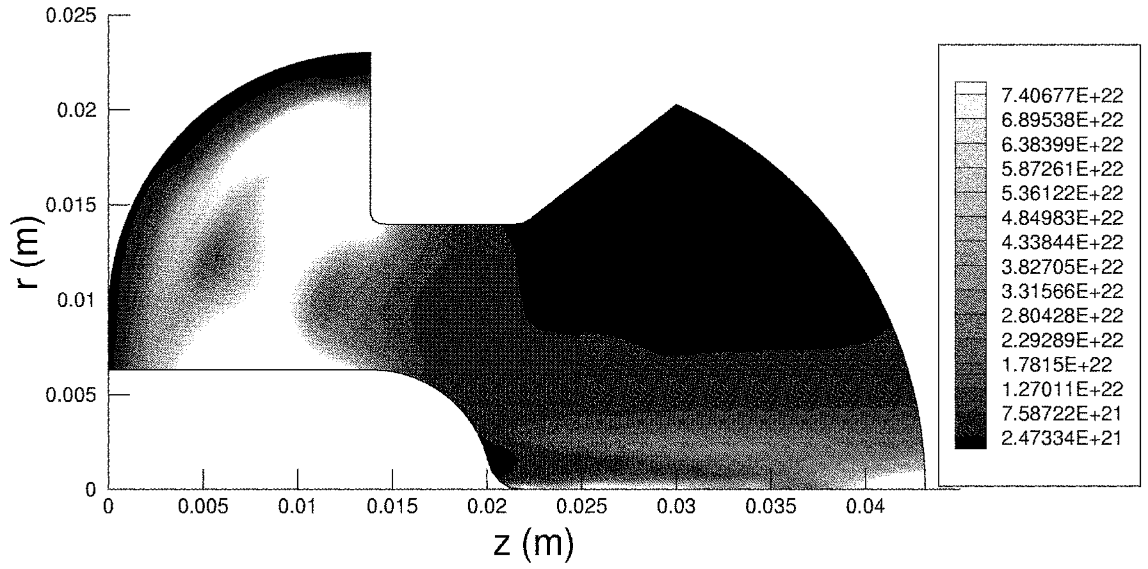

Figure 6-38: Xe-Xe$^+$ CEX rate $[1/m^3/s]$.

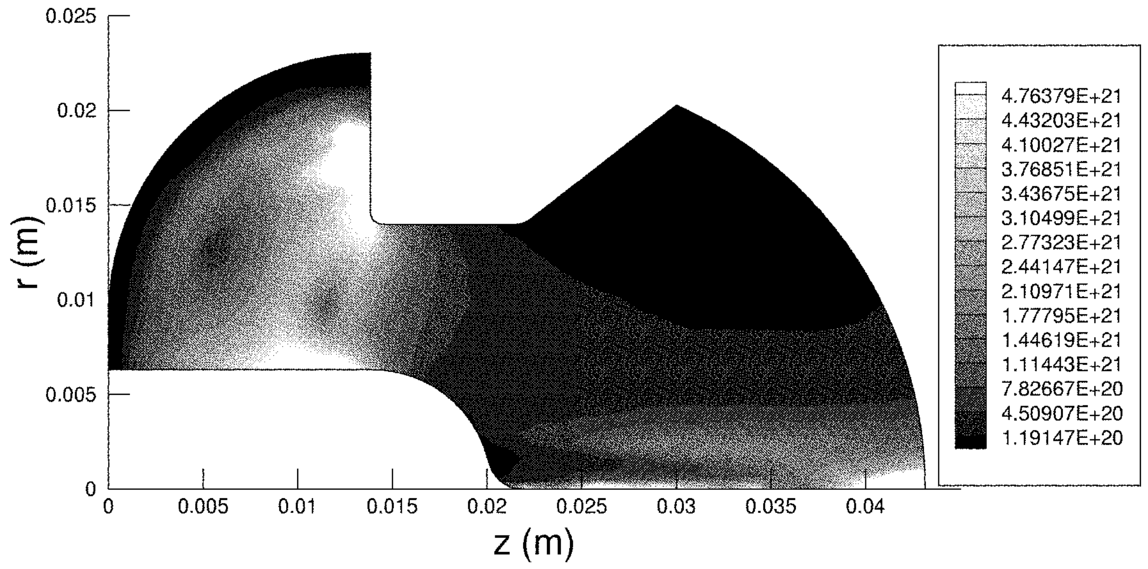

Figure 6-39: Xe-Xe^{++} CEX rate $[1/m^3/s]$.

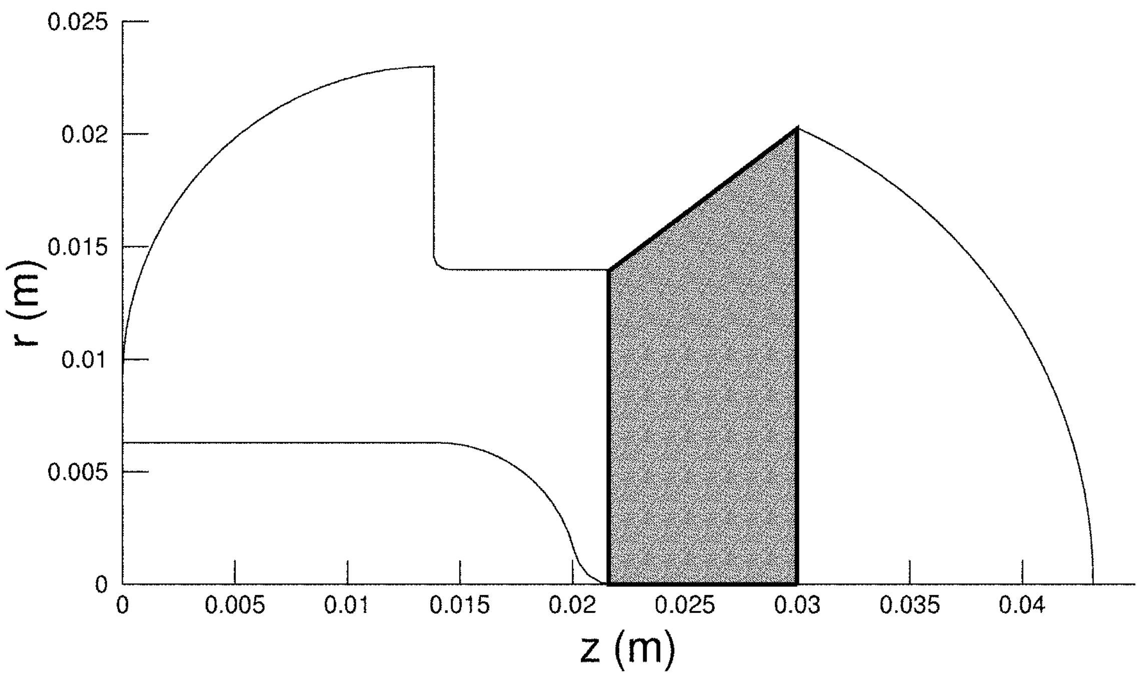

Figure 6-40: Region used to estimate missing CEX from source model.

Xe - Xe^+ CEX rate	$1.5385 \times 10^{16}\, s^{-1}$
Xe - Xe^{++} CEX rate	$9.5945 \times 10^{14}\, s^{-1}$

Table 6.1: CEX production rates in engine region before exit plane.

from the source model, a region before the exit plane at $z = .030\ m$ and where the potential is relatively small is chosen as depicted in Figure 6-40. The overall rates of CEX ion production within this region are calculated and presented in Table 6.1.

To compare these values to the amount of CEX produced in the plume, CEX ion production rates for this region are also calculated. Numbers of collisions are kept track of during the simulation and overall collison rates are derived from the collision totals and simulation time. Table 6.2 summarizes the results for various pressures.

Determining whether engine or plume CEX dominates depends on the background pressure of interest. Typically, charge exchange of the plume dictates at the higher background pressures while that of the engine becomes a more significant effect as true vacuum conditions are approached.

Pressure [$Torr$]	Xe - Xe$^+$ CEX rate [#/s]	Xe - Xe^{++} CEX rate [#/s]
0	3.336×10^{15}	3.478×10^{14}
5×10^{-7}	6.667×10^{15}	6.959×10^{14}
2.2×10^{-6}	1.320×10^{16}	1.435×10^{15}
3.34×10^{-6}	1.729×10^{16}	1.914×10^{15}
5×10^{-6}	2.283×10^{16}	2.568×10^{15}
2×10^{-5}	6.342×10^{16}	8.194×10^{15}
2.5×10^{-5}	7.841×10^{16}	1.053×10^{16}

Table 6.2: CEX production rates in plume region for various background pressures.

6.4.2 Linking of Source and Plume Models

The linkage between the source and plume models is still not ideal. An abrupt transition in modeling methodology occurs at the exit plane where the engine code stops and the plume code begins. For instance, the plume code assumes no magnetic field while there may still be a residual magnetic field in the engine code. If magnetic forces in the plume do exist, then the assumptions culminating in the use of the Boltzmann equation for electron modeling are incorrect as they presume that only pressure and electric forces are present.

Another inconsistency in modeling techniques is the treatment of electron temperature. *HPHall* allows T_e to vary across magnetic field lines while *Quasi3* assumes a constant 2 eV throughout. Other plume models [9] use significantly higher values for electron temperature such as 8 or 11 eV, so the sensitivity of *Quasi3* to this parameter is studied by running a case with 4 eV for the temperature. Figure 6-41 shows only slight broadening of the current density distribution with the higher T_e.

Figure 6-42 shows the values and averages of potential and electron temperature across the exit plane as given by *HPHall*. Although there is a distribution across the exit plane for these values that is not modeled in the plume code, the average values do correlate quite well. Notably, the average electron temperature of 1.95 eV is very close to the assumed value, thereby confirming the choice of 2 eV. Figure 6-43 shows the potential distribution of the plume as calculated by *Quasi3*. The maximum value is 2.691 V while the minimum is -6.91 V, meaning the point of highest potential just at the exit plane is at 9.601 V with respect to the infinity point of the simulation,

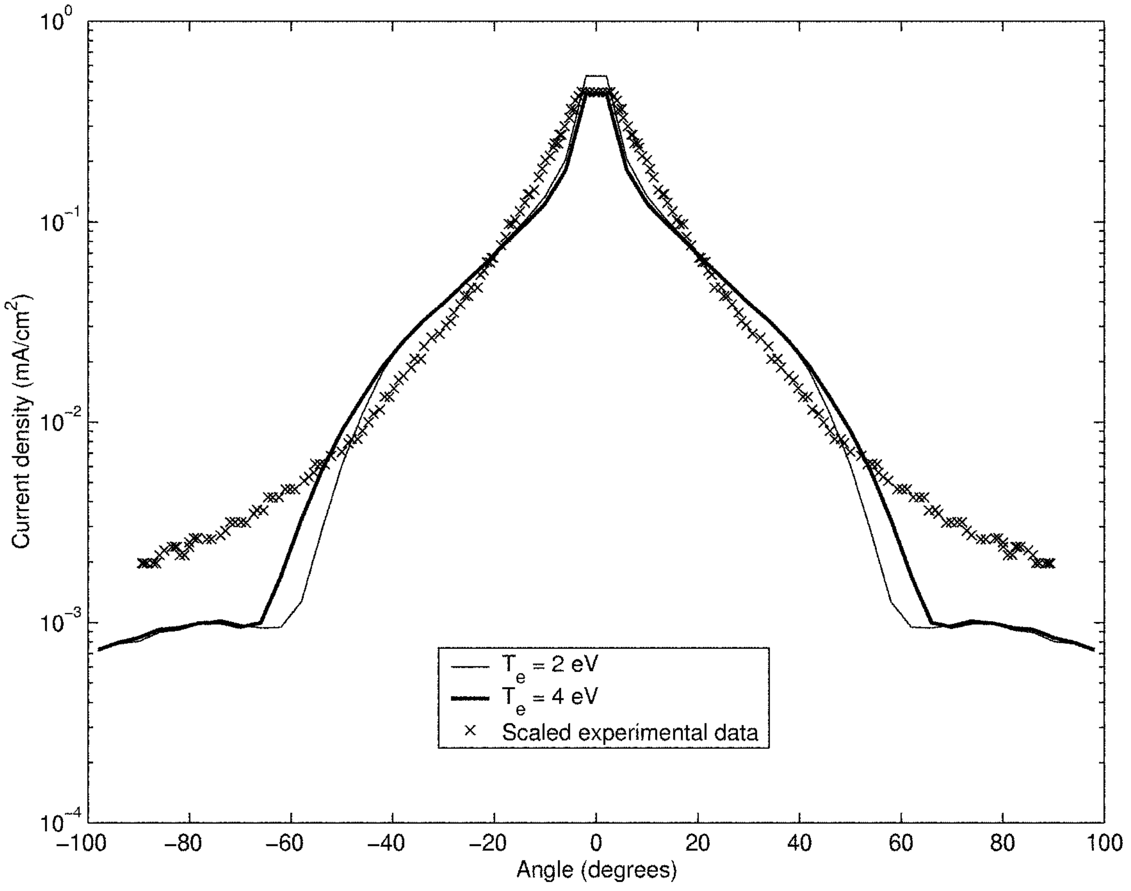

Figure 6-41: Effect of electron temperature.

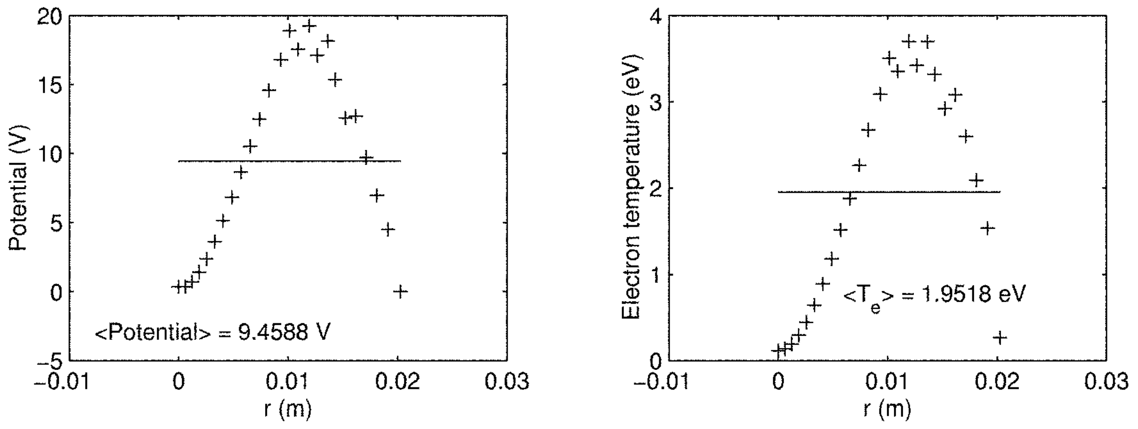

Figure 6-42: *HPHall* exit plane averages.

closely matching the average exit plane potential of 9.46 V.

Though there does appear to be promise in the linking of the current engine and plume codes, experimental observations have seen strange behavior in the region just beyond the thruster exit plane where the primary engine and cathode plasmas interact. To date, this phenomenon has not been captured by any computational model accurately.

6.4.3 Lack of Experimental Data

The lack of experimental data for the BHT-200 is a major obstacle in verifying the source model properly. Uncertainty about the scaling method makes comparison to other available engine data non-ideal.

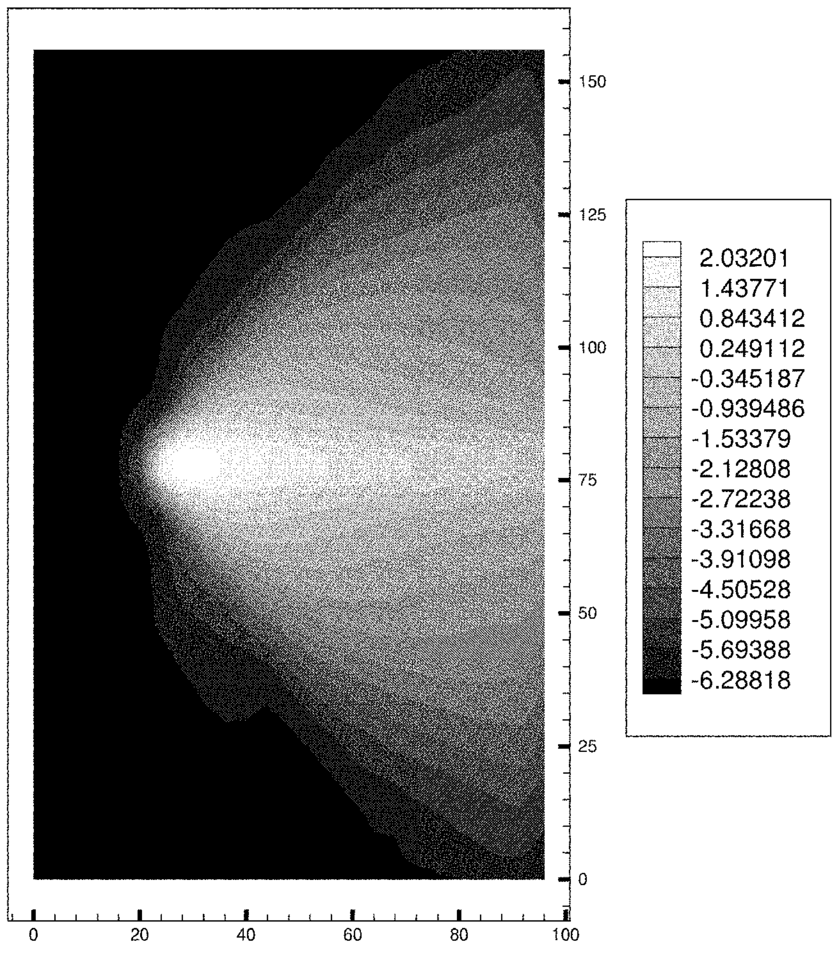

Figure 6-43: Average potential in cross-section of plume.

6.5 Conclusions

An existing plume model, *Quasi3*, has been upgraded to better simulate a tank environment. Major improvements to the source model, collision method, and sputtering method as well as minor modifications summarized in Appendix A have been made. As described in the previous sections:

- The magnitude of the simulated BHT-200 current density distribution matches well with scaled experimental data. However, the overall shape of the distribution, especially at high angles, has yet to be captured accurately.

- Increases in background pressure and accordingly background neutral density mainly affect outlying regions of the plume where an increase in pressure corresponds to an increase in current density. This phenomenon is traced to the higher frequency of CEX collisions in the denser background.

- A critical pressure exists where tank results can no longer be considered a good representation of perfect vacuum behavior. It is believed that this pressure corresponds to a balance between background and source neutral density in the primary region of CEX collision occurrence.

- Qualitative distribution of sputtering and its deposition in a simulated domain can be accomplished.

6.6 Future Work

While now better equipped to handle the tank environment, continuing issues suggest many directions for future work in plume simulation.

- Experimental data for the BHT-200 needs to be taken to verify the plume code. The data from the in-space ETEEV experiment is especially important as it will allow correlation between actual orbital data with that taken on the ground.

- CEX ions prior to the exit plane should be included to improve the source model. *HPHall* can be upgraded to model these collisions and exit plane distributions for the CEX products can be added to the *Quasi3* source model.
- Sputtered material from the engine itself should be modeled as its deposition is also a concern. *HPHall* can be altered to include a similar sputtering model to that described in Chapter 5. However, if the engine material is not elemental, alternate methods of calculating sputtering yield may need to be found since Yamamura does not tabulate data for ceramics.
- Better linkage between engine and plume code modeling methodologies should be done to assure that fundamental plume behavior is not discarded with assumptions.
- Modeling the plume on an unstructured grid would vastly improve the quality of the simulation as more realistic geometries can be modeled. Also, the current Cartesian grid may not treat certain boundary conditions correctly — for example, when two tank walls are modeled with two flat surfaces meeting along the edge, simulation of plasma in the corner edge near the intersection of the surfaces may not be modeled properly due to improper node weightings. Using an unstructured grid would require boundary conditions to be an essential part of the domain and thus avoid these concerns.
- A collision method that does not require all particles of the same species to have the same weighting should be used. The constraint of having all neutrals be the same weighting makes accurately modeling environments with a finite background neutral density exceedingly difficult and rapid simulation of tank filling nearly impossible. Using a method such as Fife's [5] where species particles may have variable weights would allow combination of macroparticles in denser areas of the plume and separation in sparser regions.
- The need to exclude collisions under certain simulation conditions can be eliminated by not using a single time counter for all collision types.

- Better modeling of the interaction between sputtering and deposition on surfaces should be considered. Presently, the model assumes a superposition of these two processes which may not be realistic. Surfaces with a large deposition of one type of material may develop a thin boundary layer of this material and modeling sputtering of the original target material becomes inaccurate as deposited material may resputter.

Appendix A

Other Code Modifications

This appendix summarizes additional code modifications made to clean up issues that arose during work.

A.1 Sheath Model

A.1.1 Sheath Drop

The sheath drop at surfaces is calculated with,

$$\Delta\phi = \frac{-kT_e}{e} \ln\left[\frac{4\Gamma_i}{n_e\bar{c}_e}\right]. \tag{A.1}$$

The relationship between the pre-sheath plasma potential and the wall potential is then given by,

$$\Delta\phi = \phi_{node} - \phi_{wall}, \tag{A.2}$$

as illustrated in Figure A-1. Previously, *Quasi3* was mistakenly treating $\Delta\phi$ as being ϕ_{wall}. This error has since been corrected.

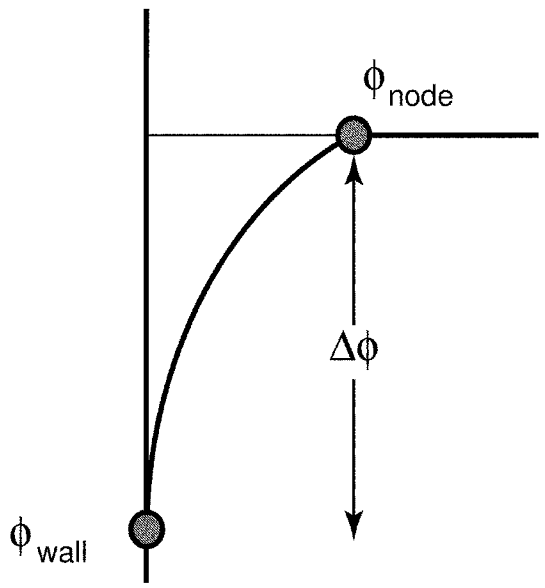

Figure A-1: Sheath drop.

A.1.2 Conducting Surface Sheaths

The previous sheath model treats all surfaces as floating with respect to plasma potential at each surface cell, effectively modeling them as insulators. However, many materials of interest that might come in contact with the plume, especially in a vacuum tank, are conductors and as such, would have a uniform wall potential. Thus, a method for modeling conducting surfaces is implemented.

The conductor wall potential is derived by balancing the ion and electron currents to the plate,

$$(I_i)_{plate} + (I_e)_{plate} = 0. \tag{A.3}$$

Converting the currents to integrals over the surface gives:

$$\iint j_i \, dA + \iint j_e \, dA = \iint e\Gamma_i \, dA - \iint \frac{e n_e \overline{c}_e}{4} e^{\frac{-e\Delta\phi}{kT_e}} \, dA = 0 \tag{A.4}$$

Substituting in Equation A.2 and rearranging gives,

$$\phi_{wall} = \frac{-kT_e}{e} \ln \left[\frac{\frac{e\overline{c}_e}{4} \iint n_e e^{\frac{-e\phi_{node}}{kT_e}} \, dA}{\iint e\Gamma_i \, dA} \right], \tag{A.5}$$

which is incorporated into *Quasi3* by integrating over conducting surfaces to find the wall potential. If the ion flux to a surface cell is smaller than some epsilon value, its contribution to the integral in the denominator of Equation A.5 is considered zero. After the wall potential is calculated, each cell is cycled through again and sheath potentials are updated with respect to the plasma potential. More detail on how to use this method is in Appendix B.

A.2 Potential Reference Method

Quasi3 calculates potentials in the simulation with respect to an assumed reference point located at infinity. Boltzmann's equation with the reference point incorporated is,

$$\frac{-kT_e}{e} \ln \left[\frac{n_e}{n_o} \right] = \phi_o - \phi. \tag{A.6}$$

Quasi3 assumes zero potential at the reference density of 1.0×10^{12} m^{-3} by default. While this assumption may hold in environments with background plasmas such as low Earth orbit (LEO), it does not necessarily apply when trying to model a vacuum tank on the ground. The amount of ionization in a gas at thermal equilibrium can be estimated by using the Saha equation [4],

$$\frac{n_i}{n_n} \approx 2.4 \times 10^{21} \frac{T^{3/2}}{n_i} e^{\frac{-U_i}{kT}}. \tag{A.7}$$

The ionization energy for xenon, U_i is 12.1 $eV/atom$. Assuming a tank pressure of 2×10^{-5} $Torr$ and standard temperature of 300 K, the background neutral density is $n_n = 6.28 \times 10^{17}$ m^{-3}. This results in an estimated background plasma density of 4.91×10^{-81} m^{-3} which is much smaller than the reference density and thus the default assumptions of *Quasi3* are erroneous.

In order to model the potential more realistically, the ability to specify a reference point for the potential within the simulation has been added. As a result, potentials can be modeled self-consistently within the domain. The user has the option to set the zero potential reference point in one of three places — at a domain node, at a surface cell, or at a specified background plasma density. In the surface cell case, the potential on the surface is set to zero and then the density and potential of the node directly above the cell is used to reference all other domain potentials. The ability to specify the background plasma density as the reference point allows more detailed modeling in a space environment because in conditions such as LEO, this density may vary widely depending on time of day.

A.3 Visualization Software

An important part of plume simulation is being able to visualize the results. Previously, *Quasi3* used *Plot3* (based on the *Visual3* and *Particle3* graphics libraries) for 3D visualization and *Surfplot* (based on the *Grafic* library) for surface visualization. While functional, these packages were also difficult to use. Furthermore, an added limitation was the inability of either *Surfplot* or the computer it was running on to accept data files containing information about more than 10 objects. This restriction of only 10 objects translated to difficulty in modeling complex geometries.

As such, *Quasi3* has been rewritten to output data files in a format that can be used in Tecplot, a commercial data visualization package. Tecplot is very user-friendly and can handle data files containing up to 32,700 zones. If each of the 6 sides of an object is represented by a zone, this means data files can contain information for up to 5,450 objects. Thus, the code is now primed to simulate complex geometries.

Appendix B

Quasi3 Manual

Much of how to use *Quasi3* is in the appendix of Oh's thesis — this section covers clarifications and usage of modifications made to the original version.

B.1 Overview

Quasi3 is named because it assumes quasi-neutrality and models the plume in three dimensions.

Using *Quasi3* has 3 components:

- grid generation
- plume simulation
- data visualization

The grid generator is actually a separate program from *Quasi3*. It is called *Mesh3* and outputs a .grid file that is used as an input into the main *Quasi3* plume simulation. *Mesh3* is used to generate the grid based on a user-defined geometry and boundary conditions.

The plume simulation is the crux of the program. *Quasi3* takes several input files and then models the plume and its interactions with surfaces. It saves output files at user-specified intervals that allow analysis of various plume properties.

The data visualization allows the user to see what is really going on in the simulation. The output files are used in Tecplot which then generates something tangible to look at instead of pages and pages of numbers.

B.2 Mesh3

Mesh3 generates a grid to represent the simulation domain. The user specifies details about the meshes and objects in a file called domain.c. (The file doesn't necessarily have to be called 'domain.c', but it must contain the same functions as domain.c.) An example domain.c file of the geometry used for the tank simulation is at the end of this section.

The sizes of meshes and objects are determined in cell widths that the user defines in normalized units. A normalized unit is equal to the reference length in the simulation. Reference quantities normalize variables throughout *Mesh3* and *Quasi3* to avoid machine computation errors when dealing with very large and very small numbers in the same calculation. The important thing to know when creating a simulation geometry is that the reference length, ref_length, is the length of a normalized unit. If the reference quantities are kept the same, ref_length is 0.010513 m. Thus, if CELL_WIDTH is set to 10 normalized units, each cell is then 0.10513 m wide. Embedded meshes have half the cell width of their parent mesh, the one they are defined on. If an embedded mesh is placed on the grid described above, its cell width would be 0.052565 m.

Oh's thesis describes how to set boundary conditions and define objects, embedded grids, and sources for the domain. When specifying widths of objects or meshes, they should be given in number of nodes. For example, if the width of the domain's parent mesh is 25 cells in the x-direction, DOMAIN_X_WIDTH should be set to 26. Offsets should be given in number of cells on the parent (if an embedded grid) or index (if an object) mesh.

A few parameters have been added to object definition. The material switch allows the user to specify the composition of objects. At present, seven materials have been incorporated into the code. It is fairly simple to add more materials as outlined in Section B.5. Currently, material options are:

- ALUMINUM
- SILICON
- SILVER
- GOLD
- IRON
- MOLYBDENUM
- UNOBTAINIUM

"Unobtainium" is a fictional material which is unsputterable — any particles that strike it will get absorbed into the surface and are deleted by the simulation. The conductswitch parameter tells the simulation whether to treat the surface as a conductor (1) or an insulator (0). Only single sides are treated as conductors which is reasonable for objects like tank walls which only have one simulation surface in contact with the plasma. However, for objects immersed in the plasma or a tank wall with a hole in it (to model the vacuum pump), the potential across the object cannot be set to the same value. The method also does not work properly if the surface lies across more than one grid. It is also recommended that if the SURFACE_CELL potential reference method is used, that the reference point not be placed on a conducting surface.

The potential reference method is specified in domain.c as well. Options are:

- SURFACE_CELL — sets potential on specified surface cell to zero, potentials in rest of domain calculated with respect to potential and plasma density at node just above cell
- DOMAIN_NODE — sets potential at node to zero

- USER_VALUES — sets potential to zero at specified plasma density
- DEFAULT_METHOD — uses default values of zero potential at ref_density

A point with reasonable values of plasma density should be chosen for the potential reference point for more self-consistent results. A good location is a node or surface cell directly in front of the thruster on the parent mesh — a point here will usually be guaranteed of having a finite density to use as a reference for the rest of the domain.

Sample domain.c file:

```
/* domain.c */
/* This file is where the simulation's geometry is specified */

#include <stdio.h>
#include <math.h>
#include <stdlib.h>
#include "constants.h"
#include "mesh3.h"
#include "globals.h"

/* Domain Definition */
#define DOMAIN_X_WIDTH 39                    /* Nodes in X direction */
#define DOMAIN_Y_WIDTH 32                    /* Nodes in Y direction */
#define DOMAIN_Z_WIDTH 40                    /* Nodes in Z direction */
#define CELL_WIDTH 4.0                       /* Normalized units */

/* Needed to calculate thruster flow rate */
#define XE_ION_MASS (0.1313/AVOGADRO)

void defineBoundaries()
/* Sets the exterior boundary conditions (along edges of the domain) */
{
  particle_bnd_switch[NORTH] = ABSORBING;
  particle_bnd_switch[SOUTH] = ABSORBING;
  particle_bnd_switch[EAST] = ABSORBING;
  particle_bnd_switch[WEST] = ABSORBING;
  particle_bnd_switch[UP] = ABSORBING;
  particle_bnd_switch[DOWN] = ABSORBING;

  neutral_flux_switch[NORTH] = FALSE;
  neutral_flux_switch[SOUTH] = FALSE;
  neutral_flux_switch[EAST] = FALSE;
  neutral_flux_switch[WEST] = FALSE;
  neutral_flux_switch[UP] = FALSE;
  neutral_flux_switch[DOWN] = FALSE;

  ion_flux_switch[NORTH] = FALSE;
  ion_flux_switch[SOUTH] = FALSE;
  ion_flux_switch[EAST] = FALSE;
  ion_flux_switch[WEST] = FALSE;
  ion_flux_switch[UP] = FALSE;
  ion_flux_switch[DOWN] = FALSE;
```

```
}

void defineEmbeddedGrids()
{
  nembeddedgrid = 3;                                    /* nembeddedgrid is a global variable */
  embeddedgrid = (embeddeddata *) calloc (nembeddedgrid, sizeof(embeddeddata));

  if (nembeddedgrid > 0) {
    embeddedgrid[0].xc = 12;        /* Width in nodes on parent mesh */
    embeddedgrid[0].yc = 14;
    embeddedgrid[0].zc = 14;
    embeddedgrid[0].xoffc = 6;      /* Offset in cells on parent mesh */
    embeddedgrid[0].yoffc = 7;
    embeddedgrid[0].zoffc = 7;
    embeddedgrid[0].parent = -1;    /* -1 indicates the top mesh is the parent */
  }

  if (nembeddedgrid > 1) {
    embeddedgrid[1].xc = 15;        /* Width in nodes on parent mesh */
    embeddedgrid[1].yc = 19;
    embeddedgrid[1].zc = 19;
    embeddedgrid[1].xoffc = 4;      /* Offset in cells on parent mesh */
    embeddedgrid[1].yoffc = 4;
    embeddedgrid[1].zoffc = 4;
    embeddedgrid[1].parent = 0;     /* -1 indicates the top mesh is the parent */
  }

  if (nembeddedgrid > 2) {
    embeddedgrid[2].xc = 13;        /* Width in nodes on parent mesh */
    embeddedgrid[2].yc = 21;
    embeddedgrid[2].zc = 21;
    embeddedgrid[2].xoffc = 8;      /* Offset in cells on parent mesh */
    embeddedgrid[2].yoffc = 8;
    embeddedgrid[2].zoffc = 8;
    embeddedgrid[2].parent = 1;     /* -1 indicates the top mesh is the parent */
  }
}

void defineObjects()
{
  nobjects = 12;                         /* nobjects is a global variable, type short int */
  objects = (box *) calloc (nobjects, sizeof(box));

  /* Piece 1: thruster */
  objects[0].index = 0;                   /* Grid in which coord are given */
                          /* This grid must surround the object ENTIRELY */
  objects[0].xc = 3;                                    /* X-width, nodes */
  objects[0].yc = 2;                                    /* Y-width, nodes */
  objects[0].zc = 2;                                    /* Z-width, nodes */
  objects[0].xoffc = 9;                                 /* X-offset, cells */
  objects[0].yoffc = 13;                                /* Y-offset, cells */
  objects[0].zoffc = 13;                                /* Z-offset, cells */
  objects[0].type = FLOATING;              /* Object surface boundary type */
  objects[0].material = UNOBTAINIUM;
  objects[0].conductswitch[NORTH] = 0;
  objects[0].conductswitch[EAST] = 0;
  objects[0].conductswitch[SOUTH] = 0;
  objects[0].conductswitch[WEST] = 0;
  objects[0].conductswitch[UP] = 0;
  objects[0].conductswitch[DOWN] = 0;
  sprintf(objects[0].name, "thruster");       /* NO SPACES ALLOWED in name */
```

```
if (nobjects > 1) {
  /* Piece 2: west wall */
  objects[1].index = 0;
  objects[1].xc = 2;
  objects[1].yc = 26;
  objects[1].zc = 34;
  objects[1].xoffc = 2;
  objects[1].yoffc = 3;
  objects[1].zoffc = 3;
  objects[1].type = FLOATING;
  objects[1].material = IRON;
  objects[1].conductswitch[NORTH] = 0;
  objects[1].conductswitch[EAST] = 1;
  objects[1].conductswitch[SOUTH] = 0;
  objects[1].conductswitch[WEST] = 0;
  objects[1].conductswitch[UP] = 0;
  objects[1].conductswitch[DOWN] = 0;
  sprintf(objects[1].name, "west_wall");
}

if (nobjects > 2) {
  /* Piece 3: east wall */
  objects[2].index = 0;
  objects[2].xc = 2;
  objects[2].yc = 26;
  objects[2].zc = 34;
  objects[2].xoffc = 35;
  objects[2].yoffc = 3;
  objects[2].zoffc = 3;
  objects[2].type = FLOATING;
  objects[2].material = IRON;
  objects[2].conductswitch[NORTH] = 0;
  objects[2].conductswitch[EAST] = 0;
  objects[2].conductswitch[SOUTH] = 0;
  objects[2].conductswitch[WEST] = 1;
  objects[2].conductswitch[UP] = 0;
  objects[2].conductswitch[DOWN] = 0;
  sprintf(objects[2].name, "east_wall");
}

if (nobjects > 3) {
  /* Piece 4: north wall */
  objects[3].index = 0;
  objects[3].xc = 33;
  objects[3].yc = 2;
  objects[3].zc = 34;
  objects[3].xoffc = 3;
  objects[3].yoffc = 28;
  objects[3].zoffc = 3;
  objects[3].type = FLOATING;
  objects[3].material = IRON;
  objects[3].conductswitch[NORTH] = 0;
  objects[3].conductswitch[EAST] = 0;
  objects[3].conductswitch[SOUTH] = 1;
  objects[3].conductswitch[WEST] = 0;
  objects[3].conductswitch[UP] = 0;
  objects[3].conductswitch[DOWN] = 0;
  sprintf(objects[3].name, "north_wall");
}
```

```
if (nobjects > 4) {
  /* Piece 5: south wall */
  objects[4].index = 0;
  objects[4].xc = 33;
  objects[4].yc = 2;
  objects[4].zc = 34;
  objects[4].xoffc = 3;
  objects[4].yoffc = 2;
  objects[4].zoffc = 3;
  objects[4].type = FLOATING;
  objects[4].material = IRON;
  objects[4].conductswitch[NORTH] = 1;
  objects[4].conductswitch[EAST] = 0;
  objects[4].conductswitch[SOUTH] = 0;
  objects[4].conductswitch[WEST] = 0;
  objects[4].conductswitch[UP] = 0;
  objects[4].conductswitch[DOWN] = 0;
  sprintf(objects[4].name, "south_wall");
}

if (nobjects > 5) {
  /* Piece 6: up wall */
  objects[5].index = 0;
  objects[5].xc = 33;
  objects[5].yc = 26;
  objects[5].zc = 2;
  objects[5].xoffc = 3;
  objects[5].yoffc = 3;
  objects[5].zoffc = 36;
  objects[5].type = FLOATING;
  objects[5].material = IRON;
  objects[5].conductswitch[NORTH] = 0;
  objects[5].conductswitch[EAST] = 0;
  objects[5].conductswitch[SOUTH] = 0;
  objects[5].conductswitch[WEST] = 0;
  objects[5].conductswitch[UP] = 0;
  objects[5].conductswitch[DOWN] = 1;
  sprintf(objects[5].name, "up_wall");
}

if (nobjects > 6) {
  /* Piece 7: down wall, piece 1 */
  objects[6].index = 0;
  objects[6].xc = 13;
  objects[6].yc = 26;
  objects[6].zc = 2;
  objects[6].xoffc = 3;
  objects[6].yoffc = 3;
  objects[6].zoffc = 2;
  objects[6].type = FLOATING;
  objects[6].material = IRON;
  objects[6].conductswitch[NORTH] = 0;
  objects[6].conductswitch[EAST] = 0;
  objects[6].conductswitch[SOUTH] = 0;
  objects[6].conductswitch[WEST] = 0;
  objects[6].conductswitch[UP] = 1;
  objects[6].conductswitch[DOWN] = 0;
  sprintf(objects[6].name, "down_wall_piece_1");
}

if (nobjects > 7) {
  /* Piece 8: down wall, piece 2 */
  objects[7].index = 0;
  objects[7].xc = 9;
  objects[7].yc = 10;
  objects[7].zc = 2;
  objects[7].xoffc = 15;
  objects[7].yoffc = 19;
  objects[7].zoffc = 2;
  objects[7].type = FLOATING;
  objects[7].material = IRON;
  objects[7].conductswitch[NORTH] = 0;
  objects[7].conductswitch[EAST] = 0;
  objects[7].conductswitch[SOUTH] = 0;
  objects[7].conductswitch[WEST] = 0;
  objects[7].conductswitch[UP] = 1;
  objects[7].conductswitch[DOWN] = 0;
  sprintf(objects[7].name, "down_wall_piece_2");
}

if (nobjects > 8) {
  /* Piece 9: down wall, piece 3 */
  objects[8].index = 0;
  objects[8].xc = 9;
  objects[8].yc = 9;
  objects[8].zc = 2;
  objects[8].xoffc = 15;
  objects[8].yoffc = 3;
  objects[8].zoffc = 2;
  objects[8].type = FLOATING;
  objects[8].material = IRON;
  objects[8].conductswitch[NORTH] = 0;
  objects[8].conductswitch[EAST] = 0;
  objects[8].conductswitch[SOUTH] = 0;
  objects[8].conductswitch[WEST] = 0;
  objects[8].conductswitch[UP] = 1;
  objects[8].conductswitch[DOWN] = 0;
  sprintf(objects[8].name, "down_wall_piece_3");
}

if (nobjects > 9) {
  /* Piece 10: down wall, piece 4 */
  objects[9].index = 0;
  objects[9].xc = 13;
  objects[9].yc = 26;
  objects[9].zc = 2;
  objects[9].xoffc = 23;
  objects[9].yoffc = 3;
  objects[9].zoffc = 2;
  objects[9].type = FLOATING;
  objects[9].material = IRON;
  objects[9].conductswitch[NORTH] = 0;
  objects[9].conductswitch[EAST] = 0;
  objects[9].conductswitch[SOUTH] = 0;
  objects[9].conductswitch[WEST] = 0;
  objects[9].conductswitch[UP] = 1;
  objects[9].conductswitch[DOWN] = 0;
  sprintf(objects[9].name, "down_wall_piece_4");
}

if (nobjects > 10) {
  /* Piece 11: witness plate 1 */
```

```
    objects[10].index = 0;
    objects[10].xc = 2;
    objects[10].yc = 6;
    objects[10].zc = 6;
    objects[10].xoffc = 18;
    objects[10].yoffc = 5;
    objects[10].zoffc = 11;
    objects[10].type = FLOATING;
    objects[10].material = SILVER;
    objects[10].conductswitch[NORTH] = 0;
    objects[10].conductswitch[EAST] = 1;
    objects[10].conductswitch[SOUTH] = 0;
    objects[10].conductswitch[WEST] = 1;
    objects[10].conductswitch[UP] = 0;
    objects[10].conductswitch[DOWN] = 0;
    sprintf(objects[10].name, "witness1");
  }

  if (nobjects > 11) {
    /* Piece 12: witness plate 2 */
    objects[11].index = 0;
    objects[11].xc = 2;
    objects[11].yc = 6;
    objects[11].zc = 6;
    objects[11].xoffc = 21;
    objects[11].yoffc = 21;
    objects[11].zoffc = 11;
    objects[11].type = FLOATING;
    objects[11].material = SILVER;
    objects[11].conductswitch[NORTH] = 0;
    objects[11].conductswitch[EAST] = 1;
    objects[11].conductswitch[SOUTH] = 0;
    objects[11].conductswitch[WEST] = 1;
    objects[11].conductswitch[UP] = 0;
    objects[11].conductswitch[DOWN] = 0;
    sprintf(objects[11].name, "witness2");
  }
}

void defineSources()
/* The source location is given by its center.  So, if you place the source on a flat edge, and then turn it to any non-zero gimble angle, */
/* you'll get particles showing up INSIDE the object it's attached to (which is obviously a bad thing).  So, be cautious in how you use */
/* these sources at least for the near future */
{
  nsources = 1;
  sources = (sourcedata *) calloc (nsources, sizeof(sourcedata));

  if (nsources > 0) {
    /* Piece 1: a hall thruster half */
    sources[0].nobj = 0;                         /* Object it's attached to */
    sources[0].xc = 12.0;       /* Coordinates on object (normalized units) */
    sources[0].yc = 2.0;
    sources[0].zc = 2.0;

    /* Vector pointing along the thruster exit direction */
    sources[0].thrustx = 1.0;
    sources[0].thrusty = 0.0;
    sources[0].thrustz = 0.0;

    /* Vector perpendicular to thrust vector pointing from center to cathode */
    sources[0].cathodex = 0.0;
    sources[0].cathodey = 1.0;
    sources[0].cathodez = 0.0;

    /* Total propellant flow rate in #/sec (cathode and anode) */
    sources[0].flow_rate = (8e-7/XE_ION_MASS);
  }
}

void initializeVariables()
{
  domain_x_width = DOMAIN_X_WIDTH;
  domain_y_width = DOMAIN_Y_WIDTH;
  domain_z_width = DOMAIN_Z_WIDTH;
  cell_width = CELL_WIDTH;

  if (POTENTIAL_METHOD == SURFACE_CELL) {
          potential_method = SURFACE_CELL;
          potential_ref_object = 1;
          potential_ref_index = 0;
          potential_ref_side = WEST;
          potential_ref_i = 24;
          potential_ref_j = 18;
  }
  else if (POTENTIAL_METHOD == DOMAIN_NODE) {
          potential_method = DOMAIN_NODE;
          potential_ref_index = 0;
          potential_ref_i = 20;
          potential_ref_j = 28;
          potential_ref_k = 28;
  }
  else if (POTENTIAL_METHOD == USER_VALUES) {
          potential_method = USER_VALUES;
          potential_ref_density = 1.0e12;
  }
  else {
          potential_method = DEFAULT_POTENTIAL;
  }
}
```

B.3 Quasi3

Quasi3 requires two input files — a .grid file generated by *Mesh3* and a .in file which contains simulation parameters. An example .in file for a perfect vacuum is at the end of this section. Table B.1 outlines values that should be included in the .in file.

There are many switches throughout *Quasi3* and *Mesh3* that the user may want to change, depending on the run case. Tables B.2 and B.3 summarize these.

Variable	Description
bkg_neutral_density [$neutrals/m^3$]	background neutral density
anode_prop_ingestion_factor	ambient neutrals ingested
anode_double_ion_fraction	fraction of ions leaving as double ions
xe_ion_wt [$realparticles/macroparticle$]	single xenon ion weighting
xe_2_ion_wt [$realparticles/macroparticle$]	double xenon ion weighting
xe_neutral_wt [$realparticles/macroparticle$]	xenon neutral weighting
sputter_wt [$realparticles/macroparticle$]	sputtered species weighting

Table B.1: Variables in a .in file.

File	Switch	Description
quasi3.h	DEFAULT_STEPS_BETWEEN_OUTPUT	Iterations between generation of output files
	DEFAULT_STEPS_BETWEEN_RESTART	Iterations between generation of restart data file
	BKG_ELECTRON_TEMP	Specify the background electron temperature
	SAVE_UNIX_DATA	Save files for use with Surfplot and Plot3
	SAVE_TECPLOT_ASCII_DATA	Save Tecplot ASCII files
	SAVE_TECPLOT_BINARY_DATA	Save Tecplot binary files
collision.c	COLLISION_FAMILY	Specify what types of collisions are modeled
source.c	ENGINE	Specify what source model to use
domain.h	SAMPLE_RADIUS	Specify radius that arc samples are taken at
	WALL_TEMPERATURE	Accommodation temperature

Table B.2: *Quasi3* switches.

File	Switch	Description
mesh3.h	POTENTIAL_METHOD	Specify how reference potential should be calculated

Table B.3: *Mesh3* switches.

Sample input file:

```
vacuum.in

bkg_neutral_density: 0.0
anode_prop_ingestion_fraction: 0.0
anode_double_ion_fraction: 0.125

xe_ion_wt: 1.13e10
xe_2_ion_wt: 1.61e9
xe_neutral_wt: 1.13e10
al_wt: 1.0e10
si_wt: 1.0e10
ag_wt: 1.0e10
au_wt: 1.0e10
fe_wt: 1.0e10
mo_wt: 1.0e10
```

B.4 Visualization Software

The original output files were written for use with *Plot3* and *Surfplot*. Because of its ease of use, output files are now generated to work with Tecplot. The original data files can still be written, but they may not contain as much information as the Tecplot files since they were not kept up-to-date with code modifications.

Tecplot files can be output either in ASCII or binary format. ASCII files are useful for debugging purposes because they can be read easily. Also, scripts can be written to post-process data from these files if needed. Binary files sometimes require less memory and take less time to write. They are also read into Tecplot much faster than ASCII files, so binary files should be written for file-intensive purposes such as movie generation.

In order to use the Tecplot output files, the Tecplot include folder needs to be added to the compiler's search path. Also, in order to write out Tecplot binary files, the TecIO.lib library needs to be added.

The user can specify what kind of data should be saved when running the simulation. This is done by modifying appropriate switches in quasi3.h as described in Table B.2. Different files are output for each option, so any combination of file types can be saved. The user specifies a file prefix for each run (v3outname) and the output files are as follows:

SAVE_UNIX_DATA:

v3outname.iteration.sdata	Surfplot file
v3outname.iteration.dat	Plot3 file

SAVE_TECPLOT_ASCII_DATA:

v3outname.surface.iteration.dat	surface data
v3outname.geometry.dat	3D geometry data
v3outname.particle.iteration.dat	3D particle data
v3outname.grid.iteration.dat	3D grid data

SAVE_TECPLOT_BINARY_DATA:

v3outname.surface.iteration.plt	surface data
v3outname.geometry.plt	3D geometry data
v3outname.particle.iteration.plt	3D particle data (all particles)
v3outname.cexxe.iteration.plt	3D CEX Xe neutral particle data
v3outname.cexxeion.iteration.plt	3D CEX Xe^+ particle data
v3outname.cexxe2ion.iteration.plt	3D CEX Xe^{++} particle data
v3outname.sputterxe.iteration.plt	3D Sputtered Xe particle data
v3outname.srcxe.iteration.plt	3D Source Xe neutral particle data
v3outname.srcxeion.iteration.plt	3D Source Xe^+ particle data
v3outname.srcxe2ion.iteration.plt	3D Source Xe^{++} particle data
v3outname.grid.iteration.plt	3D grid data

The Tecplot binary option saves many more files because of the way they need to be written. In UNIX and Tecplot ASCII, all the particle data is incorporated into one file, but Tecplot binary requires extra files. The user can select which files get loaded into Tecplot to analyze particles of interest.

When using Tecplot to visualize in 3D, generally three files need to be loaded in — the geometry, particle, and grid files. (For binary, if the user is interested in a specific kind of xenon particle, then the appropriate file should be loaded as well.) The geometry file contains information about the meshes and objects, so usually the Tecplot Mesh and Boundary layers should only be active for these zones. The particle file contains information about all the particles in the simulation — using the Tecplot Scatter layer is best for visualization. Finally, the grid file contains information about the grid nodes, so the Tecplot Contour layer can be used to visualize this.

When viewing surface data in Tecplot, after loading the surface data file, the user should do the following: Data→Alter→Shift Cell-Centered Data. Then pick all zones on the left and all variables on the right with the exception of x and y (the first two). This is because the surface data saved should be cell-centered, but Tecplot deals with data as node-centered by default. The surface data files have been written so that this should work correctly.

B.5 Adding Sputtered Materials to Simulation

The code can easily handle more materials for sputtering, as long as the sputtering coefficients for them are known. The method currently employed is that of Yamamura [20] in which sputtering coefficients for xenon impinging on target elements are tabulated. If molecular compounds are desired in the simulation, alternate ways of calculating the sputtering yield need to be implemented. Code modifications to add elemental materials are summarized in Tables B.4 and B.5.

File	Function	Change
constants.h	n/a	add to list of materials

Table B.4: Changes to *Mesh3*.

File	Function	Change
sputter.h	n/a n/a	add to list of materials change NSPUTTER
sputter.c	calculateSputterCoefficients	add data for material to list
domain.c	defineSpecies	add species to function declaration add background attributes information
domain.h	n/a	change NSPECIES add to list of species change NLABELS add to list of labels
input.c	readInputFile	add species to function declaration add check_ add else if statements for reading in species_wt add checking if material specified add printf material wts add fprintf material wts
main.c	main	declare species wt add species in readInputFile function call add species in defineSpecies function call add species to print status information
functions.h	n/a	add species in readInputFile function declaration add species in defineSpecies function declaration
tecplot.c	saveTecplotSurfaceData saveTecplotBinaryData	add in data for surface writing
	saveTecplot3Ddata saveTecplotBinary3DData	add in zonename when writing out particle data

Table B.5: Changes in *Quasi3*.

Bibliography

[1] Asare, B. K., Computational Modeling of Expanding Plasma Plumes in Vacuum and in a Tank, Master's Thesis, Massachusetts Institute of Technology, September 1999.

[2] Blateau, V., M. Martinez-Sanchez, O. Batishchev, and J. Szabo, "PIC Simulation of High Specific Impulse Hall Effect Thruster," 27th International Electric Propulsion Conference, Pasadena, CA, October 2001.

[3] Brenizer, D. M., Hybrid-PIC Modeling of Hall Thruster Plumes, Master's Thesis, Massachusetts Institute of Technology, September 2000.

[4] Chen, F. F. *Introduction to Plasma Physics and Controlled Fusion, Volume: Plasma Physics*, Plenum Press, New York, 1984.

[5] Fife, J. M., Hybrid-PIC Modeling and Electrostatic Probe Survey of Hall Thrusters, PhD Thesis, Massachusetts Institute of Technology, September 1998.

[6] Gavryushin, V. M., and V. Kim. "Effect of the Characteristics of a Magnetic Field on the Parameters of an Ion Current at the Output of an Accelerator with Closed Electron Drift." *Soviet Physics Technical Physics*, 26(4):505-507, April 1981.

[7] Katz, I., G. Jongeward, V. Davis, M. Mandell, and I. Mikellides, "A Hall Effect Thruster Plume Model Including Large-Angle Elastic Scattering," 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Salt Lake City, Utah, July 2001.

[8] Martinez-Sanchez, M., and B. Asare, "Computational Modeling of Plasma Plumes in a Vacuum Tank," 26th International Electric Propulsion Conference, Kitakyushu, Japan, October 1999.

[9] Mikellides, I., G. Jongeward, B. Gardner, I. Katz, M. Mandell, and V. Davis, "A Hall-Effect Thruster Plume and Spacecraft Interactions Modeling Package," 27th International Electric Propulsion Conference, Pasadena, CA, October 2001.

[10] Oh, D. Y., Computational Modeling of Expanding Plasma Plumes in Space Using a PIC-DSMC Algorithm, ScD Thesis, Massachusetts Institute of Technology, February 1997.

[11] Pacros, A., Preliminary Diagnostics Package for the ETEEV Shuttle Experiment, Master's Thesis in Progress, Massachusetts Institute of Technology, March 2002.

[12] Pencil, E., T. Randolph, and D. Manzella, "End-of-life Stationary Plasma Thruster Far-field Plume Characterization," 32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Lake Buena Vista, Florida, July 1996.

[13] Qarnain, S., Issues Regarding a Complete Computational Model of a Hall Thruster from the Acceleration Channel through the Plume, Master's Thesis, Massachusetts Institute of Technology, June 1998.

[14] Rapp, D., and W. E. Francis. "Charge Exchange Between Gaseous Ions and Atoms." *Journal of Chemical Physics*, 37(11):2631-2645, 1962.

[15] Sakabe, S., and Y. Izawa. "Simple Formula for the Cross Sections of Resonant Charge Transfer Between Atoms and Their Positive Ions at Low Impact Velocity." *Physical Review A*, 45(3):2086-2089, 1991.

[16] Szabo, J., M. Martinez-Sanchez, and O. Batishchev, "Fully Kinetic Hall Thruster Modeling," 27th International Electric Propulsion Conference, Pasadena, CA, October 2001.

[17] Szabo, J., M. Martinez-Sanchez, and J. Monheiser, "Application of 2-D Hybrid PIC Code to Alternative Hall Thruster Geometries," 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Cleveland, Ohio, July 1998.

[18] Thomas, S., Developing a Space Shuttle Experiment for Hall and Pulsed Plasma Thrusters, Master's Thesis, Massachusetts Institute of Technology, February 2001.

[19] Wehner, G. K. "Velocities of Sputtered Atoms." *Physical Review*, 114(5):1270-1272, January 1959.

[20] Yamamura, Y., Y. Itikawa, and N. Itoh. IPPJ-AM-26, "Angular Dependence of Sputtering Yields on Monatomic Solids." Institute of Plasma Physics, Nagoya University, June 1983.