Hall Thruster Plume Simulation Using a Hybrid-PIC Algorithm

by

Mark Michael Santi

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

September 2003

Author .....
Department of Aeronautics and Astronautics
July 28, 2003

Certified by .....
Jaime Peraire
Professor
Thesis Supervisor

Accepted by .....
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chair, Committee on Graduate Students

Hall Thruster Plume Simulation Using a Hybrid-PIC Algorithm

by

Mark Michael Santi

Submitted to the Department of Aeronautics and Astronautics
on July 28, 2003, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Hall thrusters offer the potential to significantly decrease the cost of satellite operations, but concerns over the high energy plasma plume's impact on the spacecraft have limited their general acceptance. In contrast to the cost and difficulty in obtaining relevant experimental data, accurate numerical tools for modeling the plume are inexpensive and can offer insight into ways of alleviating integration problems. A new hybrid-PIC simulation called *Aquila* has been developed to expand on the capabilities of previous computational tools. *Aquila* functions on unstructured, tetrahedral grids to obtain the greatest flexibility in modeling the geometry in question. A Poisson solver has been implemented to account for regions of the plume where neutrality breaks down. Comparisons of simulated results and vacuum tank experimental data of current density and potential show considerable agreement. Simulations using the non-neutral potential solver demonstrate that *Aquila* correctly discriminates the quasineutral from the non-neutral regions and provides information about situations that require a non-neutral approach.

Thesis Supervisor: Jaime Peraire
Title: Professor

Acknowledgments

I would first like to thank both of my advisers, Dr. Peraire and Dr. Martinez-Sanchez, for their help with this research. Dr. Peraire was always able to answer my questions concerning the numerics of the problem and provided excellent guidance whenever I was unsure about how to proceed. Dr. Martinez-Sanchez answered my many physics questions and was able to interpret and explain all the results that I didn't understand. Both of them offered me direction and encouragement even when it seemed progress was at a standstill.

Thanks also to my entire family, Mom, Dad, Emily, and grandparents. You provided a kind word and support when I felt stuck and made the extra effort to keep in touch even when I was too preoccupied to reciprocate the gesture. I really appreciate all you've done for me over the years.

I am indebted to the Air Force for funding this research, and I would especially like to thank the guys at the AFRL, Mike, Doug, Matt, and Kirtley, for their advice, expertise, and beer. I truly enjoyed working on this project with all of you. Also, I must thank Bob Haimes for helping me get GridEx up and running.

Of course, I would not have made it to this point without my friends in the SPL: Shannon, Jorge, Paulo, Noah, Yassir, Kay, Murat, Tatsuo, and Ollie (even though he's not really in the lab). I can honestly say that I have never experienced a work environment that was more fun and that is all due to you guys. You were always willing to overlook my eccentricities and occasional insanity to provide the support that I frequently needed. A special thanks must go to Shannon who besides being an excellent research partner and thesis critic, always provided a half-full attitude to temper my incessant negativity. From the ACDL, I would like to thank Victor, David, Matthieu, Garrett, Jerome, Joe, Yann, Curran, Sean, Sudeep, James, Vivian, and Tony for putting up with a lot of harassment over the past two years and for also making the lab an enjoyable workplace. Finally, I must thank my friends from Vanderbilt - Cindy, Sebastian, Sarah, Aaron, Bess, Alpana, and Ben - who always offered a welcome distraction from the rigors of research. They frequently managed to

pull my mind away from the frustrations of work and gave me much needed support throughout this process. I don't know if I could have finished this project without the help of all of you. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Electric Propulsion

Electric propulsion (EP) is a growing technology with many characteristics that appeal to satellite providers in government and industry. EP systems accelerate a charged propellant through an electric field to produce thrust. Unlike chemical propulsion systems, EP engines separate the propellant from the power supply, meaning the propellant exit velocity is not limited by the internal energy of the fuel. The best available chemical rockets generally have a specific impulse on the order of 450 seconds, but electric propulsion devices such as Hall thrusters have specific impulses typically between 1,000 and 2,000 seconds. This disparity in specific impulse represents a significant improvement in fuel efficiency for spacecraft with electric propulsion devices.

The standard rocket equation demonstrates this advantage:

$$1 - \frac{M_f}{M_o} = e^{-\frac{\Delta v}{gIsp}}, \tag{1.1}$$

where M_f is the mass of fuel and M_0 is the total vehicle mass. For a given Δv required by the mission, increasing the specific impulse, I_{sp}, results in a smaller mass fraction of fuel for the spacecraft, allowing more of the mass to be payload. Thus, EP has

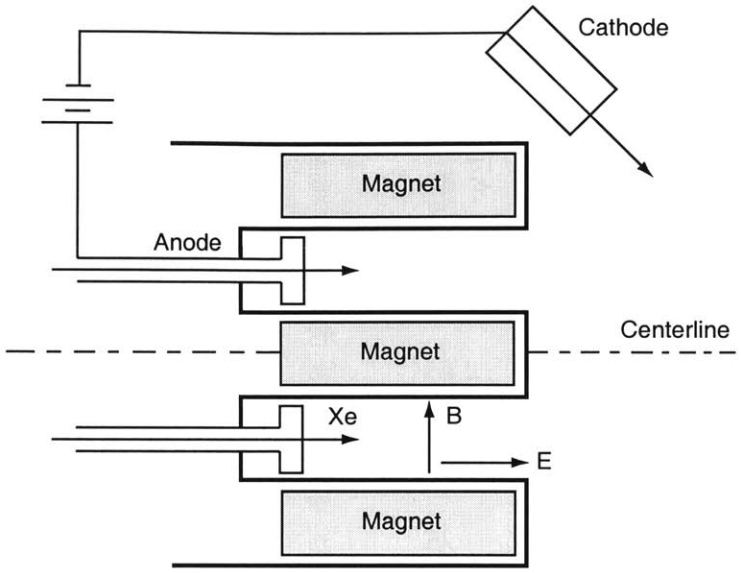

Figure 1-1: Cross-section of typical Hall thruster.

garnered more attention as a viable cost-saving technology.

1.1.2 Hall Thrusters

Hall thrusters are a particular type of electric propulsion that are well-suited for missions such as station-keeping and orbit transfer. The thruster is generally an annular device with an interior anode that emits propellant while an external cathode emits electrons (Figure 1-1). Magnets in the inner and outer portions of the annulus generate a radial magnetic field that traps electrons in the acceleration channel where they collide with and ionize the propellant flowing from the anode. Because of their small inertia, electrons are easily captured by the magnetic field and drift within the annular region (thus the name 'closed-drift' thruster), whereas the propellant ions have enough momentum to overcome the magnetic field and are accelerated through the channel by an axial electric field to create thrust.

1.2 Motivation

A primary disadvantage of Hall thrusters is the lack of understanding of how the high energy plasma emitted from the engine influences the spacecraft. As the plasma plume expands from the thruster exit, a variety of particle interactions occur that can have a significant impact on the spacecraft. Because only a fraction of the propellant is ionized, the plume contains a combination of neutrals, ions, and double ions. Due to the electrostatic acceleration through the thruster, the charged particles have much higher velocities than the uncharged neutrals, producing a comparable density of neutrals and ions in the plume. Charge-exchange collisions (CEX) occur when an ion and a neutral exchange an electron, thereby creating a fast neutral and a slow ion. Whereas most ions from the thruster have sufficient axial momentum to continue moving in the thrust direction, slow ions created in CEX collisions may accelerate through the plume's radial potential gradient and generate backflow in the direction of the spacecraft. When propellant above a certain threshold energy impacts with the spacecraft, material from the surfaces is sputtered and can degrade the performance of sensitive components. Also, these high energy particles can interfere with communication systems and sensitive instrumentation on the spacecraft or contaminate other spacecraft in a satellite array.

Because these engines are designed to operate in space, testing under true operating conditions is a costly and time-consuming process. On-ground vacuum chamber tests have provided a large amount of data to assist in the understanding of Hall thrusters, but the imperfect vacuum conditions cast doubts on information relating to processes that are strongly influenced by a background of neutral particles. Due to the difficulty in obtaining reliable experimental data for a space environment, accurate numerical tools would be a major contribution to the study of these engines. Although many simulations of EP plumes exist, the assumptions and simplifications necessary to obtain information in a reasonable amount of time have limited the utility and application of these codes.

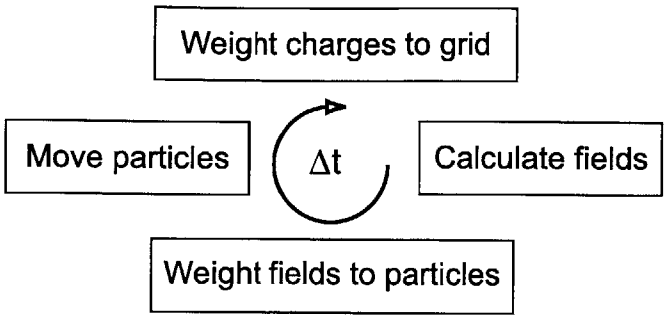

Figure 1-2: Computational cycle of PIC method

1.3 Hybrid-PIC Method

The Particle-in-Cell (PIC) method is a numerical technique that kinetically models a physical system by tracking the motion of a representative number of particles. The real particles in the system are lumped into macro-particles to obtain a feasible number of simulated particles for computation. A continuum approximation of properties is calculated by weighting the particles to a grid. Both external and self-induced fields are calculated on the grid and weighted back to the particles. These fields determine the acceleration and subsequent velocity of the particles, which are moved for a particular time step based simply on the equations of motion. Figure 1-2 shows the computational cycle that the PIC method follows [11] [2].

The PIC method is appropriate for plasma simulations because the long-range Coulomb interactions between particles dominate [2]. In the case of a plasma, charge is weighted to the grid to avoid directly calculating the forces between each particle pair. This charge density, $\rho(x, y, z)$, is used to find the electric potential, ϕ, governed by Poisson's equation,

$$\nabla^2\phi = -\frac{\rho(x, y, z)}{\epsilon_o}. \tag{1.2}$$

The gradient of the potential gives the electric field, $E(x, y, z)$,

$$E(x, y, z) = -\nabla\phi. \tag{1.3}$$

The force on a particle is then determined from the Lorentz equation,

$$\vec{F} = q\left(\vec{E} + \vec{v} \times \vec{B}\right). \tag{1.4}$$

Hybrid-PIC methods treat certain types of particles kinetically, while others are modeled as a fluid. In plasma simulations, heavy species (neutrals, ions, and double ions) are modeled as particles, and electrons are treated as a fluid. Because of their low inertia and high mobility, electrons react quickly to any potential gradients and can be represented accurately with a Maxwellian distribution [5]. By treating the electrons as a fluid, not only is considerable memory saved by not storing the electron particle information, but the numerical time step need not resolve the electron plasma time (10^{-9} to 10^{-11} seconds), thus speeding computation time. Although the hybrid-PIC method is much more efficient than the N-body problem of directly calculating the force between each particle pair, it suffers from noise and numerical heating issues because of the finite number of particles being modeled [11].

1.4 Previous Work

1.4.1 Quasi3

Quasi3 is a Hall thruster plume code developed by Oh [13]. It uses a hybrid-PIC algorithm with a DSMC collision model to simulate the expansion of the plume. *Quasi3* provides useful information about the erosion of surface materials and the characteristics within the plume, but suffers from certain limitations that curtail its utility.

The most restrictive aspect of *Quasi3* is that computational grids must be structured. Cartesian grids cannot accurately represent the actual geometry of a satellite or vacuum chamber because the objects are limited to be rectangular in shape and must have boundaries parallel to grid lines. Generating new meshes for different simulations is a complicated and tedious process because of the method of geometry definition. Embedded meshes offer a certain degree of refinement to regions of the computational domain, but not to the degree necessary to fully describe the complicated plume re-

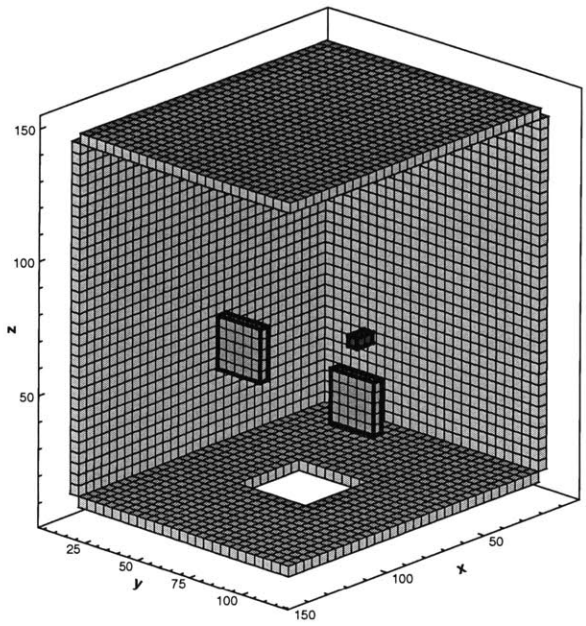

Figure 1-3: Quasi3 grid of MIT vacuum chamber

gion. Thus, the simulation is limited to run on unrealistic computational grids such as the square vacuum chamber in Figure 1-3 [6].

Throughout the computational domain, *Quasi3* assumes that the plasma is quasineutral, which means that the density of ions, n_i, is equal to the density of electrons, n_e,

$$n_i = n_e. \tag{1.5}$$

By making this assumption, solving Poisson's equation is no longer necessary, and the potential is calculated by assuming a Maxwellian electron distribution and inverting Boltzmann's equation,

$$n_e = n_o e^{-\frac{e\phi}{kT_e}}. \tag{1.6}$$

Quasineutrality is a valid assumption for Low Earth Orbit or vacuum tank simulations where the Debye length of the plasma is much shorter than the scale of the computation. However, in situations such as Geosynchronous orbit or the wake of a witness plate in a tank experiment, quasineutrality no longer holds. For these simula-

tions, solution of Poisson's equation is required to accurately calculate the potential. Therefore, the situations for which *Quasi3* is valid are limited.

1.4.2 Other PIC Simulations

Other research groups are using similar numerical tools to analyze a variety of problems. Sonnendrucker [14] has implemented the PIC method on a two-dimensional unstructured grid to solve Maxwell's equations. At the University of Michigan, Boyd [3] has developed a hybrid-PIC DSMC simulation similar to *Quasi3* and has compared the simulation results to data gathered on a Russian Express satellite. Mikellides, et.al. [12] at SAIC have developed the Environmental Work Bench for modeling a range of spacecraft plume interactions. However, no single simulation currently combines a simple way to produce suitable grids with the flexibility to solve a variety of problems with the same tools.

1.5 Aquila

Aquila is a new hybrid-PIC simulation that expands on the capabilities of the *Quasi3* simulation by using unstructured tetrahedral grids and incorporating a Poisson solver for non-neutral regions of the plume. The grid aspects and potential solver are discussed in-depth in the following chapters, but because the development of this simulation has been a collaborative effort, the source, collision, and surface interaction models are only briefly described here.

1.5.1 COLISEUM

Aquila operates within COLISEUM, a computational framework developed by Fife, et.al. [9] at the Air Force Research Lab to facilitate the development of plasma simulations. COLISEUM provides a standard way of building a plasma simulation and handles overhead by providing routines for input and output of information. By standardizing how information is passed into the simulation, plasma modules integrated

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 1.1: BHT-200 source model parameters.

into COLISEUM can be used to run simulations of varying fidelity without altering the overall setup. Thus, the same input files can be used to get a first estimate of plume behavior using a crude algorithm or to obtain a more detailed description of the plume with another module.

1.5.2 Source Model

The source model provides the input to the plume simulation and must be accurate if there is any hope of obtaining meaningful results. Cheng [6] originally developed the *Aquila* source model for *Quasi3*, but reimplemented it for use in the new framework and geometry. The source model represents the conditions at the exit of the BHT-200 thruster developed by the Busek Company. The information is generated using time-averaged results from Fife's *HPHall* engine code [8]. Ions and double ions are treated separately by *HPHall*; therefore, separate distributions of flux and both axial and radial velocity are determined for each species. The source model also indicates two distinct peaks in the radial velocity distribution for both single and double ions as a result of particles originating from the far-side of the thruster. Thus, different distributions are also created for these near-side and far-side populations. The flux

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

Table 1.2: Collisions simulated in *Aquila*.

and axial and radial velocity distributions are determined as a function of radial position from the center of the thruster exit plane, and these equations are then sampled to determine the appropriate number and characteristics of particles to be inserted into the plume simulation at each iteration. The source model parameters for the BHT-200 are shown in Table 1.1.

1.5.3 Collision Model

Aquila uses the Direct Simulation Monte Carlo (DSMC) method to model the effects of important elastic and charge-exchange collisions. The five collision types listed in Table 1.2 are modeled within *Aquila*. Celik [4] has implemented a No-Time-Counter (NTC) DSMC model to remove the statistical complications experienced when particles in the same simulation must represent a significantly different number of actual particles. This difference in macroparticle weighting is a necessary condition for simulations within a vacuum chamber, where the background of neutrals always far outnumbers the number of ions. Only particles within the same computational cell are considered valid collision partners. Collision pairs are then selected randomly, and their probability of collision is determined from their relative velocity and collision cross-section.

1.5.4 Surface Interactions

Cheng [6] utilizes a modified Yamamura method for calculating sputtering yield and outgoing angular distribution. When a particle strikes a surface within the simulation, the incident particle's energy and angle as well as the target material properties are used to determine the amount of material removed from the target surface.

1.6 Outline of Research

This research has focused on expanding the capabilities of current Hall thruster plume simulations. Chapter 2 describes the numerical implementation of unstructured grids in a hybrid-PIC simulation. Chapter 3 explains the calculation of potential assuming quasineutrality in the plume, and Chapter 4 describes the potential solution when quasineutrality breaks down. Chapters 3 and 4 also present results from the simulation and comparisons to analytical solutions as well as some experiments conducted at MIT. Chapter 5 gives conclusions and suggestions for future work.

Chapter 2

Grid Implementation

2.1 Overview

Aquila functions on unstructured, tetrahedral grids in order to accurately represent complex geometries. The computational domain is defined by means of solid models constructed with CAD software packages, and the tetrahedral grid is then created based on this solid model using an automatic grid generator. Tetrahedral discretization offers the most flexibility in generating grids that can sufficiently represent complicated satellite configurations. Though tracking particles in a tetrahedral mesh is a more difficult and time-consuming problem than in the rectangular case, *Aquila* incorporates techniques rendering this a relatively efficient process.

2.2 Geometry Definition and Grid Generation

Efficient methods of geometry definition significantly streamline the process of developing an appropriate grid for simulating the plume region. Solid Works CAD software is used to create solid models of the computational region for *Aquila* simulations. Two-dimensional renderings are created either with Solid Works' built-in drawing tools or by specifying curves from an input file. These 2D drawings are then extruded into 3D solids, and by combining multiple drawings, complex objects can be accurately modeled. Individual components of a complicated structure can be created

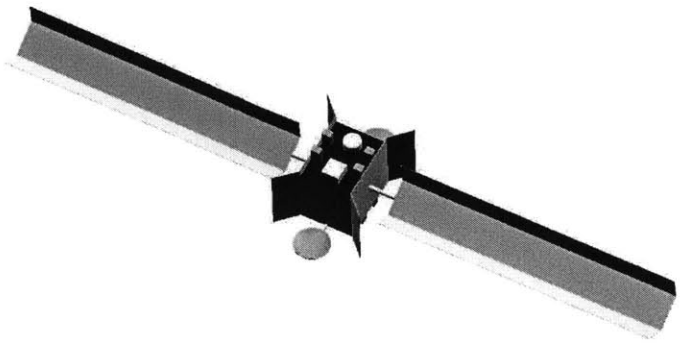

Figure 2-1: Solid Works model of a satellite.

and then joined together into a single entity. Figure 2-1 shows a satellite composed of 29 separate parts modeled in SolidWorks. Components such as the solar arrays are easy to maneuver into any orientation by simply changing the parameters that define how they are attached to the rest of the overall structure. SolidWorks allows output into a parasolid text format, a generic format that many grid generation packages can interpret. The solid model must be a water-tight volume (no open boundaries), and all surface normals must point inward toward the volume being gridded in order for the grid generator to properly interpret the structure.

GridEx is a grid generating package created at NASA Langley [10] for CFD applications, but the tools are applicable to plasma simulations as well. GridEx separates the meshing processes of the surface and volume into separate phases that do not require communication between one another. This decoupling of the surface and volume discretization allows for greater flexibility. The geometry definition is read from the parasolid text file created within SolidWorks, and the user can choose to only create a surface triangulation for simulations such as ray tracing or to also generate the full 3D volume grid for simulations such as PIC. Directional refinement gives the user the ability to pinpoint regions of the computational domain that require more

Variable	Description
dS	grid spacing at source
Constant Dist	radial distance from source where grid spacing remains dS
Distance	radial distance at which grid spacing recovers background spacing

Table 2.1: GridEx source definition variables

grid resolution. GridEx provides three types of source terms for defining the desired grid refinement. The user defines a point, line, or triangle within the computational domain that needs a more precise definition. The user-defined variables in Table 2.1 dictate the level of refinement that GridEx uses at each source position. For the purposes of a Hall thruster simulation, a line source originating at the thruster exit plane and extending along the exhaust direction normally provides sufficient grid refinement for the expanding plume. The grid metrics in Table 2.1 are specified at each end of the line source; therefore, a smaller grid spacing is provided at the thruster end of the source to achieve finer grid resolution in this high density region.

COLISEUM also offers the option of generating the volume grid given a surface triangulation. This discretization uses the same fundamental routines as GridEx, but COLISEUM does not currently support source definition and directional grid refinement with this method. Basically, the tetrahedral volume is "grown" from the surface assuming no sources exist. The geometry requirements for a water-tight volume and inward-facing normals also exist for this option. Thus, the users of COLISEUM have the ability to use another grid generation package such as Cosmos Works that only performs surface discretizations.

2.3 Finite Element Method

The interpolation functions (also called basis or shape functions) for the tetrahedral grid are based on the volume coordinates of a particular location within the elements. Volume coordinates are obtained by calculating the relative volumes of the four tetrahedra created by connecting the interior particle location to the nodes of the element.

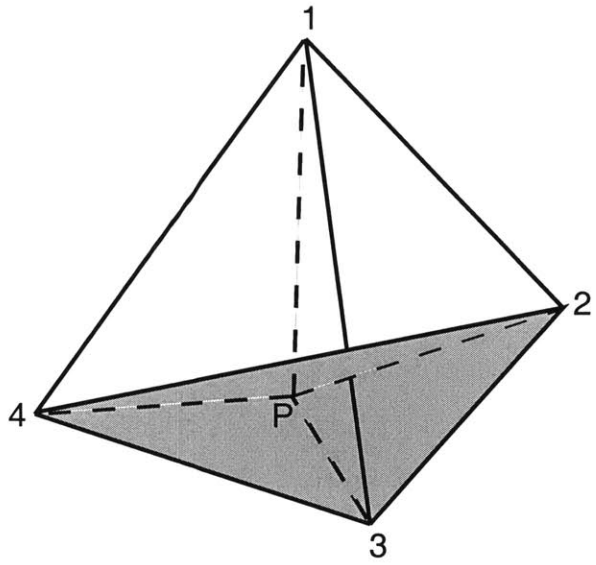

Figure 2-2: Sample tetrahedron with interior position p

Figure 2-2 shows a sample tetrahedron with a particular location P chosen in the interior. For a given position, the shape function for each node is the ratio of the volume of a tetrahedron with corners at the three other nodes of the element and the interior point to the total volume of the tetrahedron. In Figure 2-2, the volume coordinates for node 1 is the ratio of the volume of the shaded region to the total volume of the tetrahedron,

$$L_1 = \frac{V_{P234}}{V_{1234}}. \tag{2.1}$$

The volume of a tetrahedron is the determinant of a four by four matrix of the coordinates of its nodes,

$$V_{1234} = \frac{1}{6}\begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}. \tag{2.2}$$

Thus, the volume coordinates of each node are the ratio of the determinants of two 4 by 4 matrices. This ratio simplifies to a linear function based on the coordinates of

the tetrahedral nodes,

$$L_1 = \frac{\begin{vmatrix} 1 & x_P & y_P & z_P \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}}{6V} = \frac{\alpha_1 + \beta_1 x_P + \gamma_1 y_P + \delta_1 z_P}{6V}. \tag{2.3}$$

The four coefficients in equation 2.3 are each determinants of a 3 by 3 matrix. For each of the four nodes in each element, these four coefficients are calculated once and used throughout the simulation. For each node, the equation is of the form,

$$L_i = \frac{\alpha_i + \beta_i x + \gamma_i y + \delta_i z}{6V}. \tag{2.4}$$

During every iteration, the charge of each particle within the simulation is interpolated to the nodes using the basis functions, which are precisely the volume coordinates for the linear case. The charge is weighted to the nodes with the equation,

$$q_i = \sum_k q_k N_i(\underline{x}_k). \tag{2.5}$$

The basis functions also interpolate the force due to the electric field back to the particle,

$$F_k = q_k \sum_i E_i N_i(\underline{x}_k). \tag{2.6}$$

A drawback to the finite element method for electric field applications is that with linear basis functions, the field is only piecewise constant. Higher order basis functions can be used, but these will also produce discontinuous fields. A hybrid implementation is necessary to obtain continuous electric field functions. Discontinuous fields can possibly lead to tesselations in the grid and bunching of particles, but with a tetrahedral grid this is not a significant problem. The PIC method requires approximately 5 to 10 particles per cell in order to achieve good levels of accuracy because the inter-

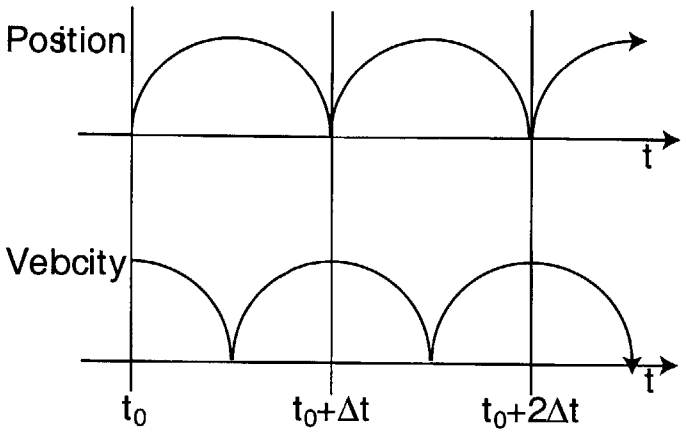

Figure 2-3: Leapfrog time integration scheme

polation to the grid assumes a continuum of the charge distribution. If a reasonable number of particles are not present in each cell, the continuum approximation does not provide sufficient accuracy.

2.4 Particle Mover

2.4.1 Leapfrog Method

Aquila uses the leapfrog time marching scheme for moving particles. Figure 2-3 shows that the position and velocity of the particles are known at different points during the computational time step. The equations of motion for the particles that are treated kinetically are drawn from Newtonian physics,

$$\frac{dx_p}{dt} = v_p \tag{2.7}$$

$$m_p \frac{dv_p}{dt} = F_p. \tag{2.8}$$

Using the leapfrog algorithm, these equations are discretized to the form,

$$\frac{x_p^{n+1} - x_p^n}{\Delta t} = v_p^{n+\frac{1}{2}} \tag{2.9}$$

$$m_p \frac{v_p^{n+\frac{1}{2}} - v_p^{n-\frac{1}{2}}}{\Delta t} = F_p. \tag{2.10}$$

The leapfrog method is second order accurate in time and has no amplitude error for timesteps that resolve the plasma frequency. More accurate time marching schemes such as the fourth order Runge-Kutta method are available, but MacNeice [11] claims that the leapfrog scheme has the best balance of accuracy, stability, and efficiency for particle simulations. Because the PIC method requires a large number of particles to be moved at each iteration, the efficiency of the time algorithm is a serious consideration, and the leapfrog method appears to be the best option.

2.4.2 Particle Search

The primary drawback to unstructured grids for PIC simulations compared to structured, Cartesian meshes is the extra cost of locating particles. Accurately finding the particles element location is necessary to properly weight the charges to the grid and the forces back to the particles. For each particle at each iteration, the particle mover performs a directional search based on the particle's location during the previous time step. For each element in the domain, the numbers corresponding to the elements that border on each face are stored in memory. If the four shape functions determined by the particle's position are between 0 and 1 for a given element, the particle lies in the interior of that element. Otherwise, the particle's new element location must be determined. Figure 2-4 shows a 2D analogy of how the shape functions, N_i, facilitate the search. After the position vector of a particle is updated in the move operation, the particle's previous element location is checked by calculating the shape functions at the particle's new position. If any shape function is not between 0 and 1 and the particle has moved to another element, the next element to be checked corresponds to the neighboring element with the most negative shape function. This procedure is performed recursively until the new element is found or the particle strikes a surface (see Figure 2-5).

To determine if a particle crossed a domain boundary, the point where the particle

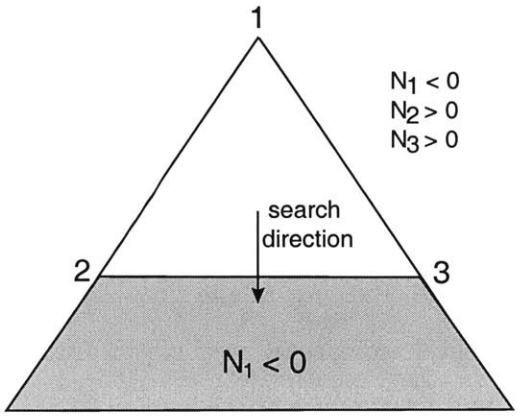

Figure 2-4: Shape functions determine search direction

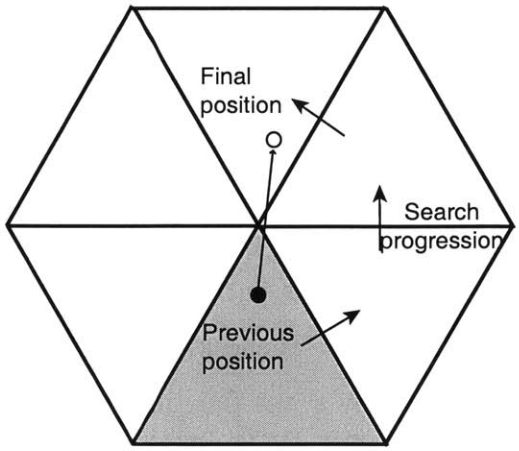

Figure 2-5: Recursive search algorithm

crossed in the plane of the surface is calculated based on the updated position and velocity. This position is then connected to the three nodes of the triangular surface element, and the areas of the three new triangles are calculated and compared to the total area of the full surface element. When a particle crosses the boundary, the appropriate surface interaction operation is chosen based on the incident particle properties and the target material.

2.5 Particle Storage

The position, velocity, and element location are stored for each particle in the computational domain. In addition to these necessary values, particles within a given element are referenced to one another using a linked list in order to perform collisions efficiently. A block data structure is used in order to speed the calculation. The block structure stores many particles in sequential locations in memory; therefore, each update of the position and velocity requires fewer memory calls and increases overall efficiency. As the computation progresses, particles are deleted, and a list of open memory locations is kept and filled by new particles. In this way, all available memory within a block is utilized before another particle block is allocated. Because particles consume the bulk of the simulation memory, they are stored in a zone data structure to facilitate a future transition to parallel computing.

Chapter 3

Quasineutral Potential

3.1 Overview

As mentioned earlier, the quasineutral assumption means that the number density of ions and electrons is equal at each point in the domain,

$$n_e = n_i. \tag{3.1}$$

This assumption significantly reduces the computation time in a hybrid-PIC simulation as the potential can be calculated directly from the momentum equation. Therefore, when quasineutrality is appropriate, the potential is determined using a direct analytical solution, making Poisson's equation unnecessary.

According to Chen [5], quasineutrality is a valid assumption as long as the domain of interest is many Debye lengths in overall dimension. For most problems, only a thin sheath region at boundaries contain any disparities with the quasineutral assumption. The Debye length,

$$d_D = \sqrt{\frac{\epsilon_o k T_e}{e^2 n_e}}, \tag{3.2}$$

is a measure of the shielding distance of the plasma and represents the distance over which non-neutralities are shielded out by the electrons. Thus, for applications such as Low Earth Orbit and vacuum tank simulations when the Debye length is very

small compared to the overall dimension of the system, quasineutrality is valid. Only in special circumstances, such as a wake region, will the plume not behave according to quasineutrality.

3.2 Derivation of Quasineutral Potential

Oh [13] shows that the general electron momentum equation for the case of a Hall thruster plume under normal operating conditions does not need to include inertial, magnetic, or collisional contributions. After these simplifications, the formulation equates the electrical and pressure terms,

$$\nabla P_e = e n_e \nabla \phi. \tag{3.3}$$

Assuming the electrons have a Maxwellian distribution, the ideal gas law is used to describe the electron pressure,

$$P_e = n_e k T_e. \tag{3.4}$$

By combining Equations 3.3 and 3.4, the potential is found to be,

$$\nabla \phi = \frac{k T_e}{e} \left(\frac{\nabla T_e}{T_e} + \frac{\nabla n_e}{n_e} \right). \tag{3.5}$$

If the temperature and density are assumed to have a polytropic relationship,

$$\frac{T_e}{T_{eo}} = \left(\frac{n_e}{n_{eo}} \right)^{\gamma - 1}, \tag{3.6}$$

then the potential takes the form,

$$\nabla \phi = \frac{\gamma}{\gamma - 1} \frac{k T_{eo}}{e} \nabla \left(\frac{n_e}{n_{eo}} \right)^{\gamma - 1}. \tag{3.7}$$

Thus, the potential behaves according to the equation,

$$\phi - \phi_o = \frac{\gamma}{\gamma - 1} \frac{kT_{eo}}{e} \left[\left(\frac{n_e}{n_{eo}} \right)^{\gamma - 1} - 1 \right]. \tag{3.8}$$

At this point, *Aquila* offers two options for calculating the potential assuming quasineutrality. An assumption of isothermal electrons implies that $\gamma = 1$, and by l'Hospital's rule, Equation 3.8 simplifies to the inverse of Boltzmann's relation from Equation 1.6,

$$\phi - \phi_o = \frac{kT_e}{e} ln \left(\frac{n_e}{n_{eo}} \right), \tag{3.9}$$

where ϕ_o and n_{eo} are reference quantities taken from a specific point in the simulation. Alternatively, if the polytropic constant γ is known from experiments, the potential can be self-consistently calculated directly from Equation 3.8 by assuming a reference potential at some point within the computational domain. Therefore, for each run of the simulation, a point must be selected where all potential values are referenced to, for both the constant and polytropic temperature cases.

3.3 Quasineutral Results

Part of the appeal of COLISEUM is its versatility in simulating different geometries within the same fundamental framework. Simulation results on a vacuum tank geometry and on a satellite configuration in perfect vacuum are presented.

3.3.1 Vacuum Tank Simulations

The vacuum tank pictured in Figure 3-1 is used to simulate experiments performed in the MIT vacuum chamber. The front and back walls have been removed for easier viewing. Unlike the previous vacuum chamber grid created for *Quasi3* (Figure 1-3), this grid closely matches the actual chamber geometry. The chamber, roughly cylindrical in shape, is 137 *cm* in length and 68.5 *cm* in radius. The thruster is 10 *cm* in diameter, but the thruster exit plane where particles are injected into the simulation is only 4 *cm* in diameter. Figure 3-2 shows the surface triangulation of

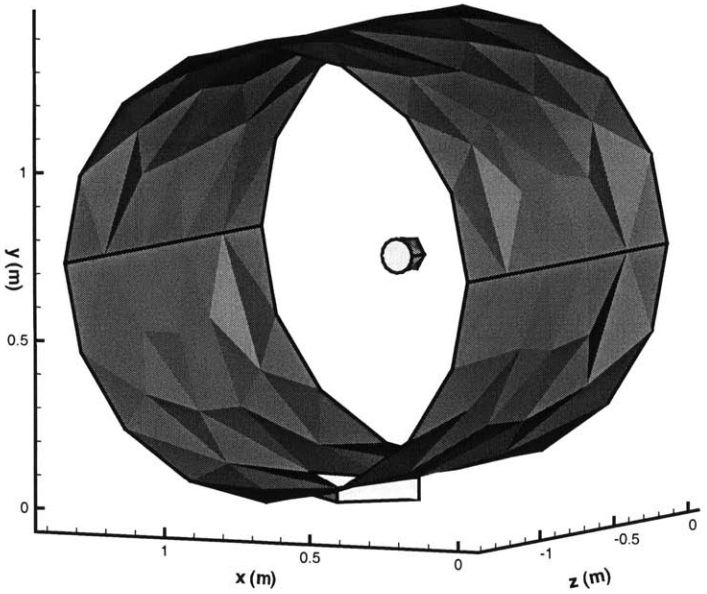

Figure 3-1: Simulated vacuum tank geometry.

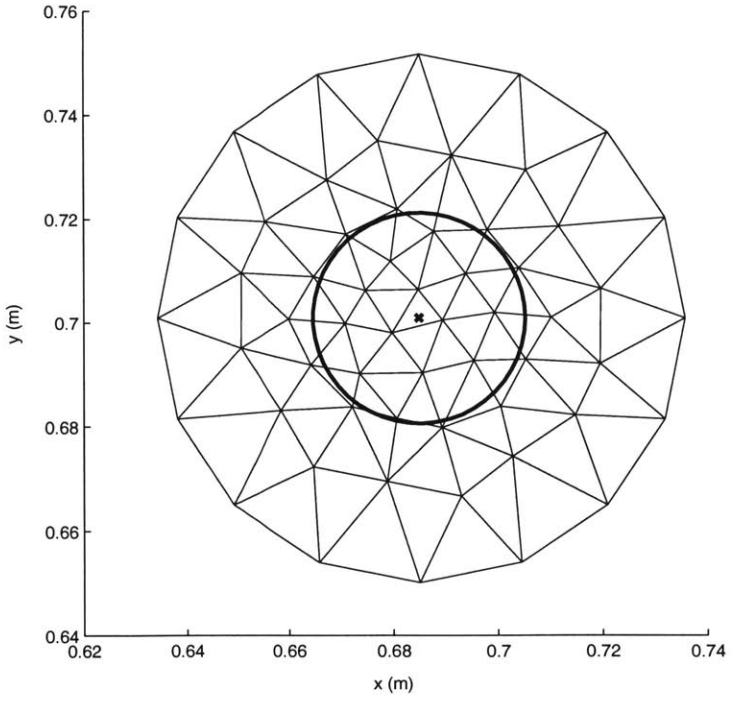

Figure 3-2: Surface grid for simulated thruster face.

the thruster face with the actual exit region circled. The tetrahedral grid contains a single linear grid source originating at the center of the thruster exit and terminating 50 cm downstream.

The simulation mimics experiments performed by Azziz [1] in both the Busek and MIT vacuum chambers. For comparison, cases are run with both the constant temperature and the polytropic temperature models. The constant temperature is chosen to be 2 eV. The polytropic simulation assumes a temperature of 2.8 eV at a distance of 25 cm from the thruster exit with a γ of 1.3 as determined from Azziz's experimental results. Each simulation is run for 15,000 iterations with a computational time step of 10^{-7} s. The thruster is the BHT-200 with the parameters listed in Table 1.1, and simulated current density and potential probes are placed in arcs of 25 cm and 47 cm in front of the thruster exit. Current density measurements are found by tracking the charge of ions crossing a hemisphere at different angular bins at the specified radial distance from the thruster. Potential measurements are directly calculated at the distance and angle specified for the simulation by using the finite element approximations.

Figures 3-3 and 3-4 show comparisons of current density for the nominal thruster discharge voltage of 300 V with a background pressure of 2.2×10^{-5} $Torr$. At 25 cm, both temperature models predict the mid-angle region of the current density profile well, but the centerline of the constant temperature model has a peak that does not appear in experimental data. Conversely, the polytropic model approximates the centerline well. However, at angles between about 60 and 75 degrees, both models underpredict the current density. The real temperature near the center of the exit plane is higher than the 2 eV constant value; thus, for the polytropic model, the higher temperature at the exit disperses ions and spreads the flux. At 47 cm, both models underpredict the high angle region again, but unlike the 25 cm case, the polytropic model also underpredicts the centerline region.

Figures 3-5 and 3-6 compare the potential for a discharge voltage of 250 V with a background pressure of 3.2×10^{-5} $Torr$. Because the source model for the simulation is for the 300 V thruster, velocities are scaled to match the 250 V operating condition

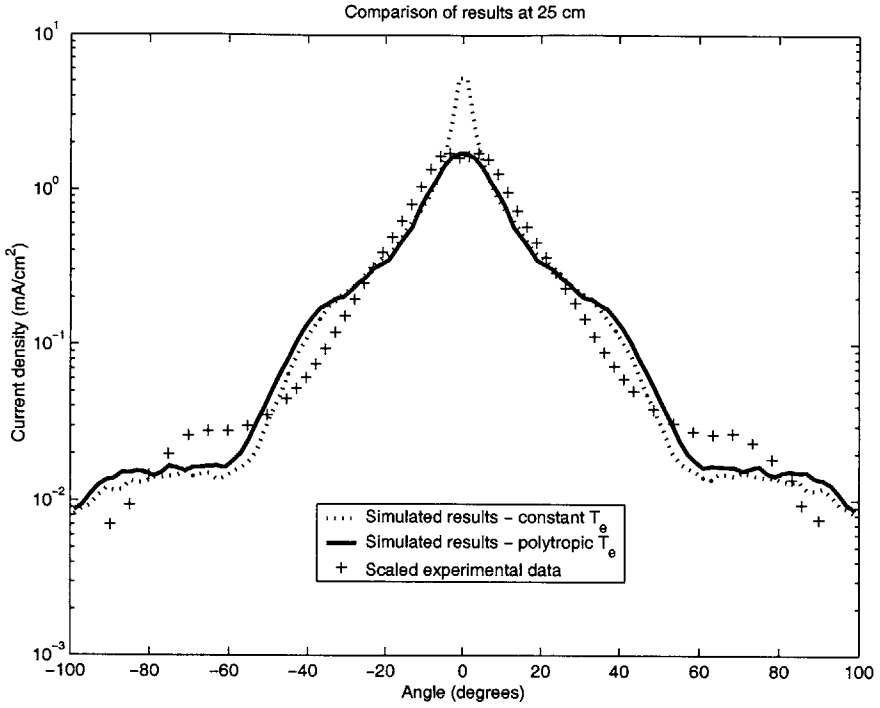

Figure 3-3: Comparison of simulated results and experimental data of current density at 25 *cm*.

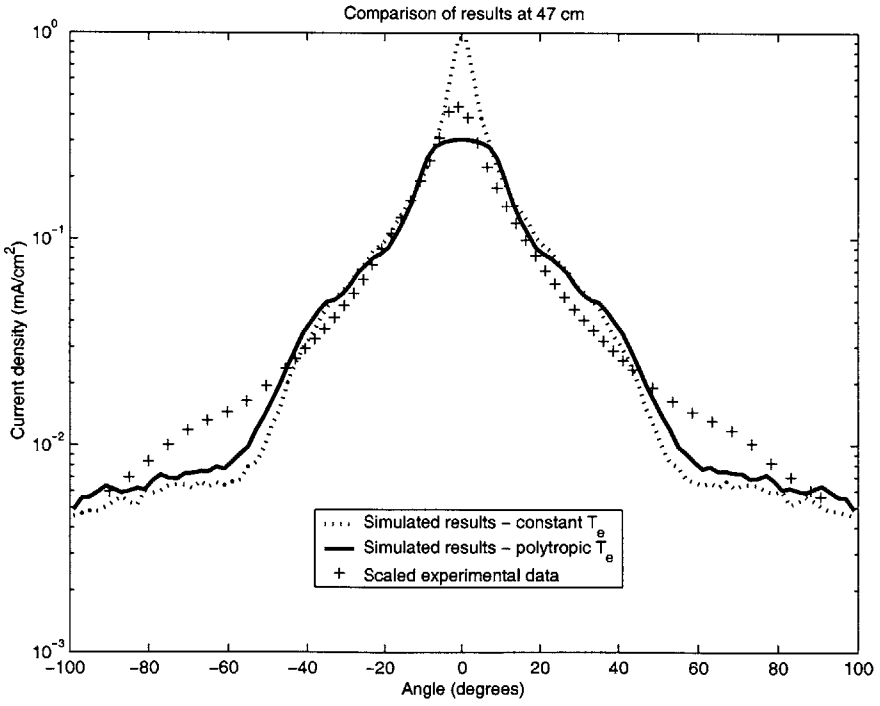

Figure 3-4: Comparison of simulated results and experimental data of current density at 47 *cm*.

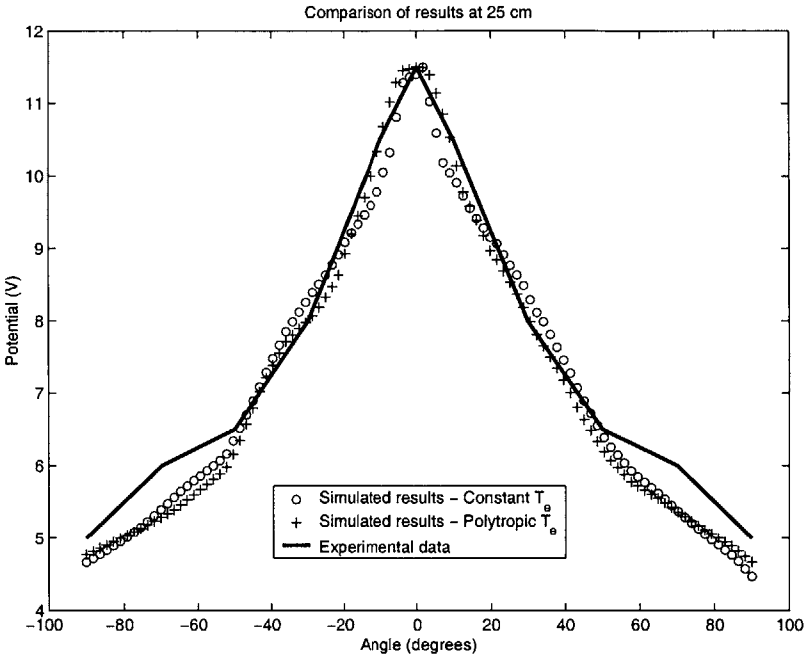

Figure 3-5: Comparison of simulated results and experimental data of potential at 25 *cm*.

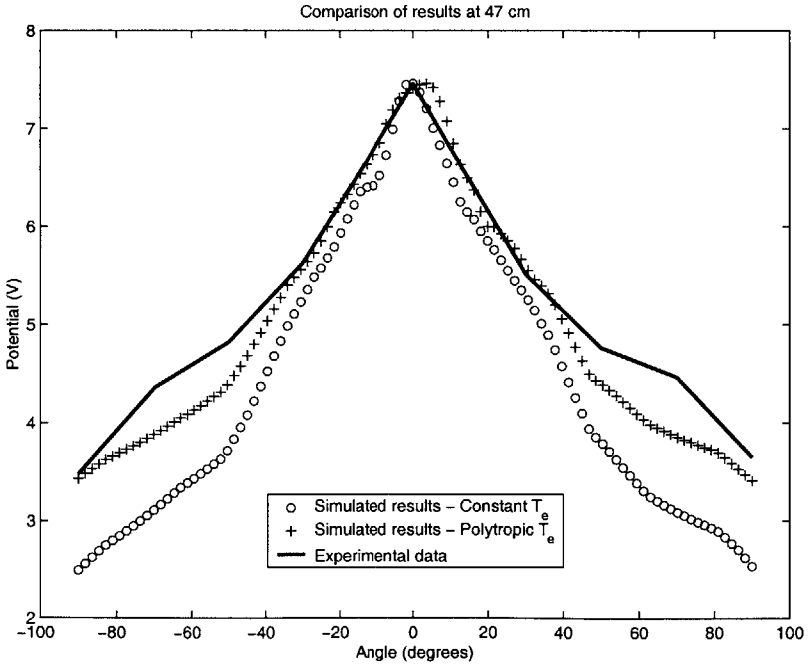

Figure 3-6: Comparison of simulated results and experimental data of potential at 47 *cm*.

used in the experiment by,

$$\frac{v}{v*} = \sqrt{\frac{V}{V*}}, \tag{3.10}$$

where v is velocity, V is the operating voltage, and the starred quantities are for the 300 V nominal operating condition. The peaks of the simulated potential are matched to the experimental data for purposes of comparison. Both temperature models agree well with the potential data at 25 cm, but at 47 cm, the overall potential drop for the constant temperature case is significantly larger than that indicated by the experimental data, suggesting that temperature drops axially.

These four figures indicate that the current density near the centerline is extremely sensitive to the chosen value of γ, whereas the radial potential drop does not have as strong of a dependence. Figures 3-7 and 3-8 show the potential contours directly in front of the thruster at the same cross-section for the constant and polytropic cases. The contours in the polytropic case are closely grouped directly downstream of the thruster, creating a strong radial potential gradient in this region. In contrast, the constant temperature case does not demonstrate this bunching of contours. Therefore, ions in the near-thruster region experience a much stronger radial potential gradient with the polytropic temperature model. The reason for this stronger gradient is the higher electron temperature in this region resulting from the high density of particles. The velocity phase-space plots for the ions in each simulation provide further evidence. Figures 3-9 and 3-10 show the axial and radial velocities of the single ions from the source and those resulting from CEX and elastic collisions. Although the fundamental structure of the two plots is the same, the radial velocity in the polytropic case for the CEX ions has a noticeably higher peak than the constant temperature case. Even the high energy source ions have a higher maximum radial velocity in the polytropic case. Due to the larger radial potential gradient, more particles are diverted to the wings, thus reducing the centerline current density peak that has been observed in previous calculations with a constant electron temperature [4].

Figures 3-11 and 3-12 show the influence of the grid and the chosen value of γ

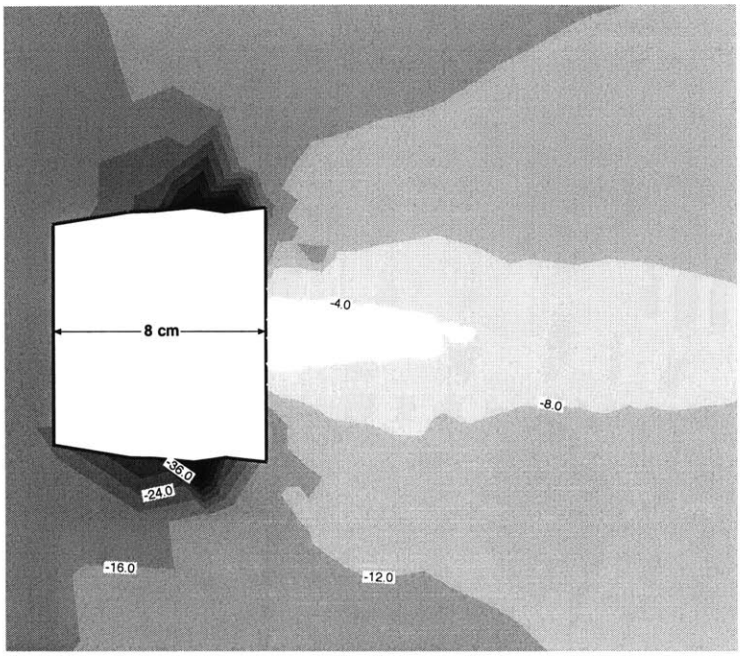

Figure 3-7: Contour plot of potential around the thruster for constant T_e case.

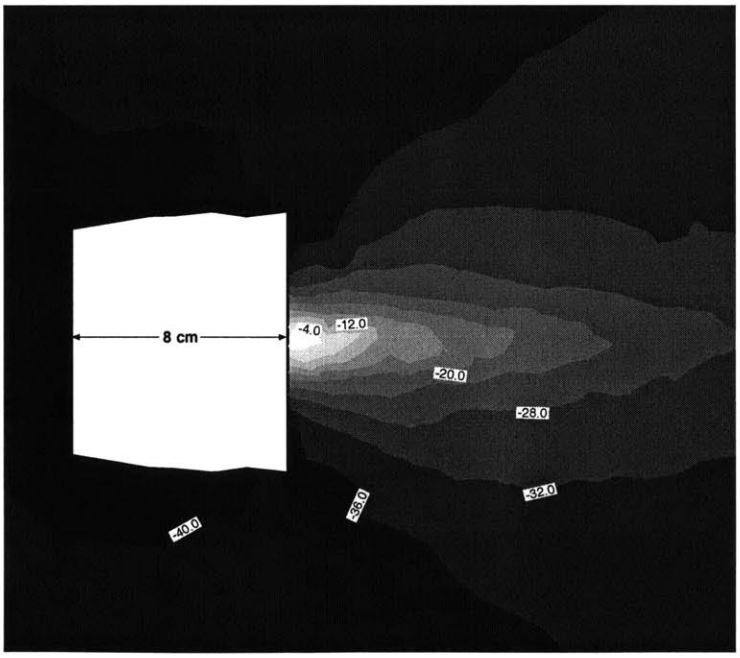

Figure 3-8: Contour plot of potential around the thruster for polytropic T_e case.

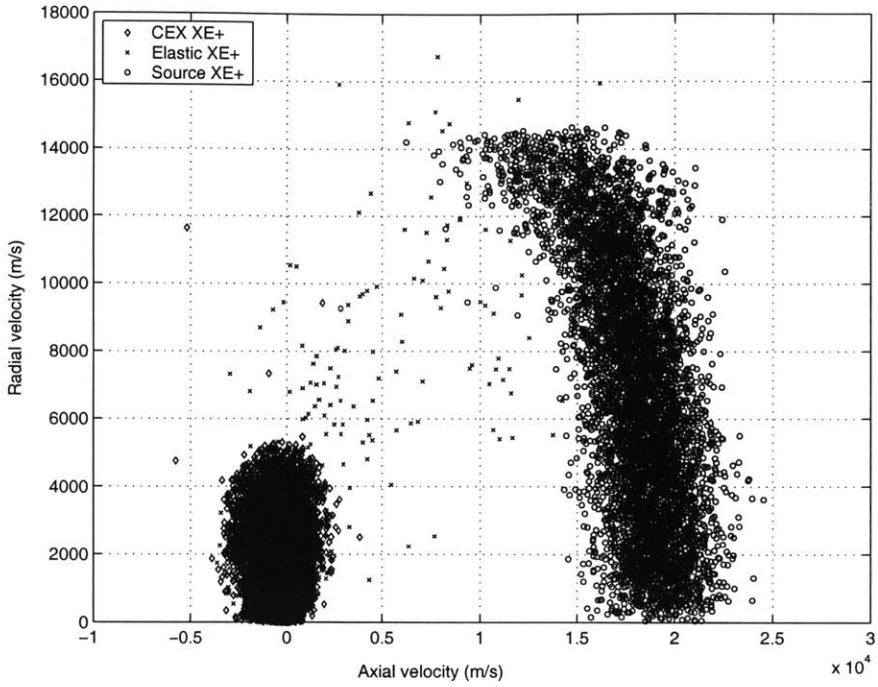

Figure 3-9: Velocity phase-space plot of single ions for constant T_e case.

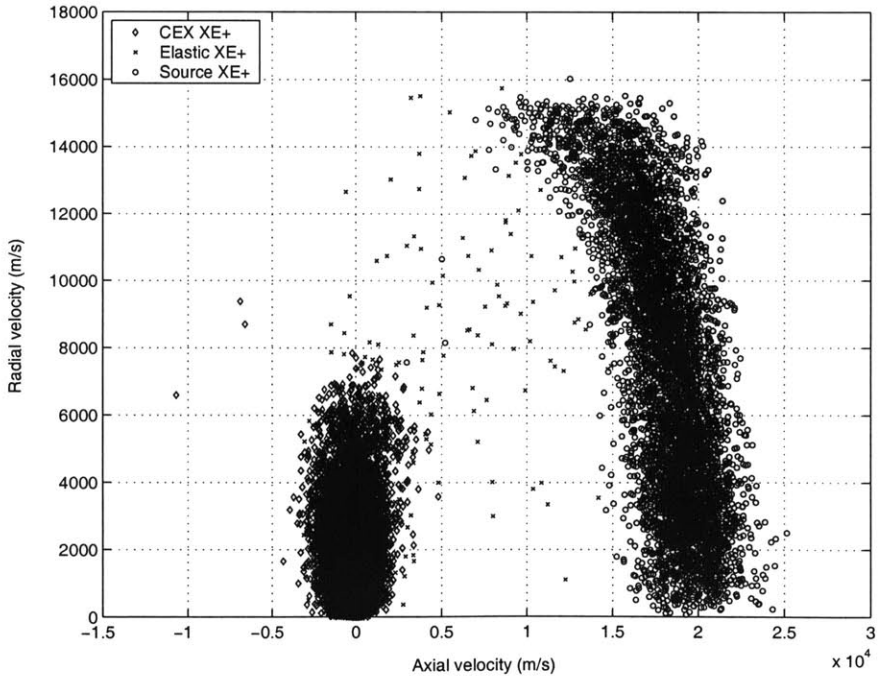

Figure 3-10: Velocity phase-space plot of single ions for polytropic T_e case.

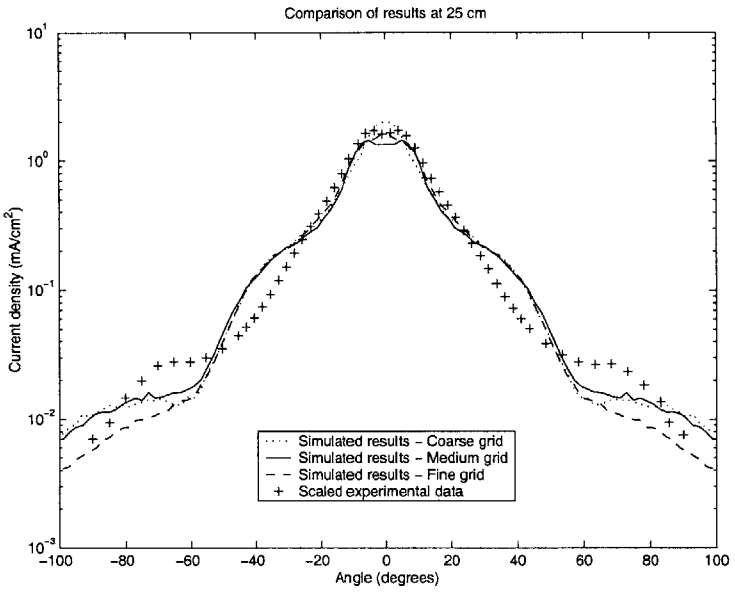

Figure 3-11: Comparison of current density at 25 *cm* with three grid refinements at the thruster exit.

on the current density results. Figure 3-11 compares the current density at 25 *cm* using three different grid refinements at the thruster exit. The coarse grid contains approximately 4 triangular edges across the exit plane diameter, the medium grid approximately 9, and the fine grid 16. All three cases are run assuming a polytropic temperature model with a γ of 1.25. The three cases produce similar results except the fine grid appears to underpredict the wing region slightly worse than the other two. Only the medium refinement exhibits the dip in current density at the centerline that is also observed in the experimental data. Varying γ for the polytropic temperature model has a more pronounced impact on the current density results than the grid (Figure 3-12). As γ increases, the current density at the centerline decreases and agrees with experimental data with a γ of approximately 1.3, but all simulations still underpredict the wing region.

Regardless of temperature model and grid refinement, all simulations exhibit a significant discrepancy in current density in the wing region. This region is dominated by CEX ions, which are of particular interest to spacecraft designers as these ions cause the most damage. Figure 3-13 shows a logarithmic contour plot of the CEX ion number density averaged over the final 5000 iterations of the polytropic simulation.

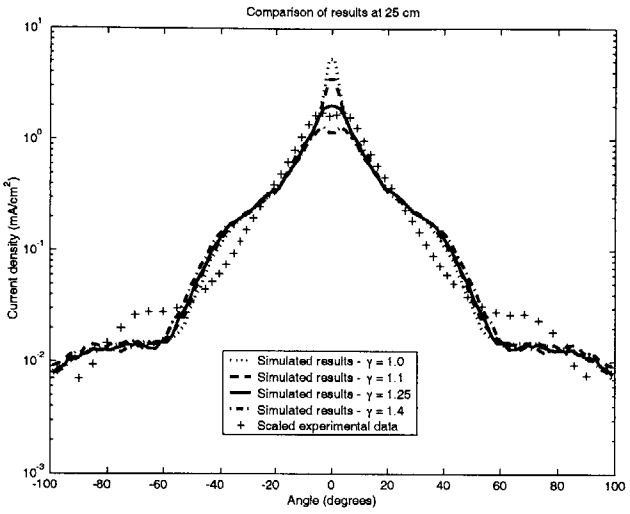

Figure 3-12: Comparison of current density at 25 *cm* with four values of γ.

The CEX particles do not exhibit grouping to the thruster sides, but this is a result of scattering along the entire plume instead of immediately outside the thruster exit as expected in a true vacuum.

Comparing simulations with each temperature model to the experimental data reveals that the polytropic model captures the plume physics significantly better than the constant temperature model. The radial potential gradient that pushes slow ions to the wings is more pronounced using the polytropic model, resulting in an appropriate centerline current density measurement. The tetrahedral grid with refinement in the primary plume region provides results that are more accurate than previous simulations of similar problems [6]. Run times for the simulations are less than 2 hours despite the extra cost of moving particles on the tetrahedral grid. The ability to model the actual geometry of the vacuum tank also makes *Aquila* a more useful tool for gathering information about geometry-dependent properties such as sputtering and deposition.

3.3.2 Satellite in a Perfect Vacuum

To demonstrate the advantages of using unstructured grids, a simulation is performed with the simplified satellite in Figure 3-14. The light-colored surfaces represent the

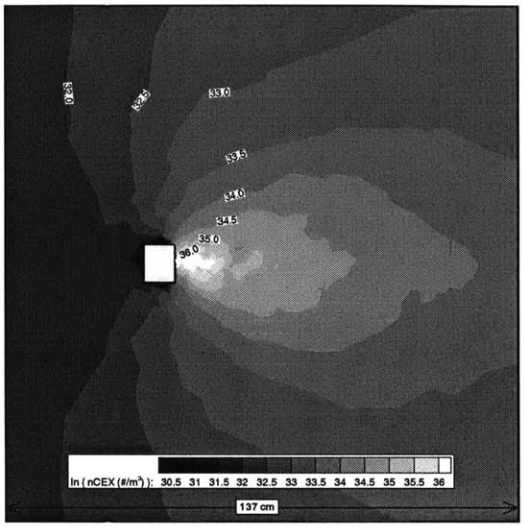

Figure 3-13: Logarithmic contour plots of CEX number density in polytropic T_e case.

actual spacecraft surfaces, whereas the darker color shows the edge of the computational domain. The geometry has a crude satellite structure with two solar arrays, four solar reflectors, the bus, and the thruster at the center of the bus. The simulation is run assuming perfect vacuum conditions with no background of neutrals. This case is run for only 6000 iterations because it reaches a steady-state more quickly than the previous tank cases. The simulation uses a polytropic temperature model with a γ of 1.25 at 25 cm from the thruster exit.

Though this hypothetical simulation does not have any experimental data with which to compare, the logarithmic contour plots of number density in Figures 3-15, 3-16, and 3-17 are informative as they show that ions from the source and from CEX and elastic collisions are in their expected regions. Source ions form the familiar plume pattern, while the CEX ions develop very distinctive wings. The elastically scattered ions also exhibit concentrations in the 60 degree region as suggested by theory. The primary difference between this case and the vacuum tank simulations in the previous section is the lack of a background of neutrals in the satellite simulation. Because *Aquila* requires all particles of the same species within the simulation to have the same weighting (for purposes of the DSMC collision model), the presence of a background in the vacuum chamber necessitates an extremely large neutral weighting compared

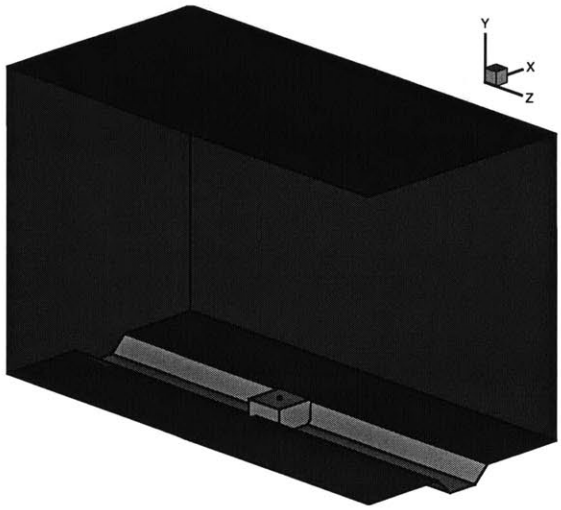

Figure 3-14: Geometry of a simplified satellite.

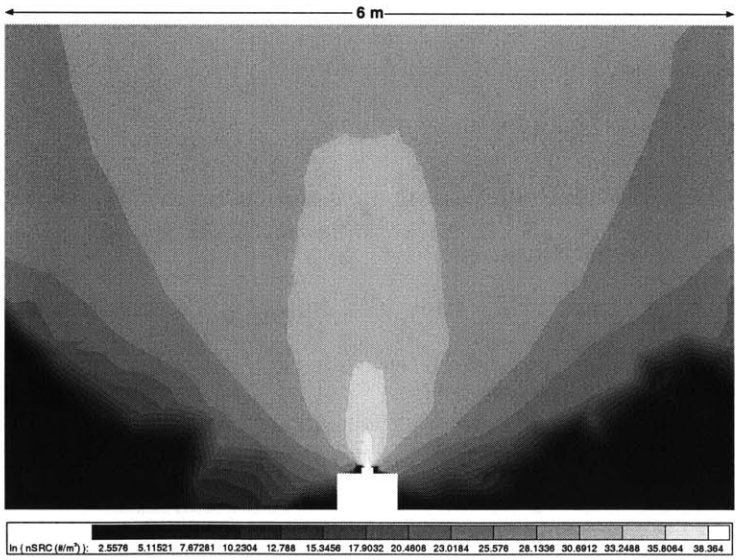

Figure 3-15: Logarithmic contour plot of number density for source ions.

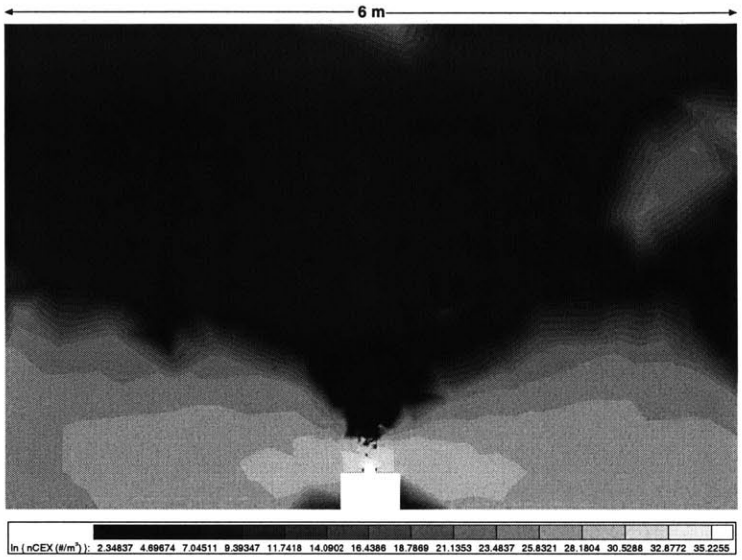

Figure 3-16: Logarithmic contour plot of number density for CEX ions.

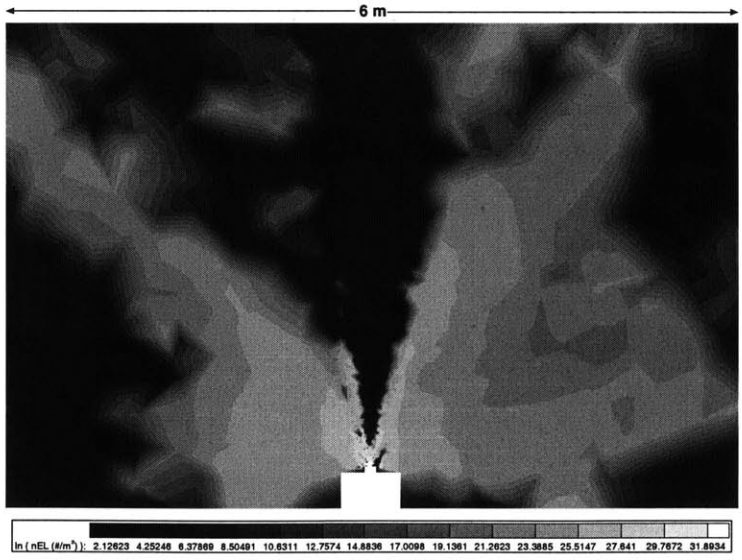

Figure 3-17: Logarithmic contour plot of number density for elastically scattered ions.

to the ions and double ions from the thruster. In the pure vacuum simulation, the particles are given equal weightings. This simulation confirms that ions created at the source or resulting from collisions are ending up in their expected locations within the plume, but does not offer any clear insight into the causes of the discrepancies between the experimental data and simulated results in the vacuum tank simulations.

Chapter 4

Non-neutral Potential

4.1 Poisson's Equation

The potential, ϕ, due to space charge given by Poisson's equation is a function of the electron density, ρ_e, and the ion density, ρ_i,

$$\nabla^2\phi = \frac{\rho_e - \rho_i}{\epsilon_o}, \tag{4.1}$$

but the electron density is a function of the potential obtained by solving the simplified electron momentum equation, as in Section 3.2. Thus, Poisson's equation takes on the form,

$$\epsilon_o\nabla^2\phi - \rho_e(\phi) = -\rho_i. \tag{4.2}$$

Depending on the chosen model, the electron density term assumes a different form, but the solution method remains the same. The next section describes the discretization of Equation 4.2 assuming a Boltzmann relation for the density and a constant electron temperature, T_e,

$$\epsilon_o\nabla^2\phi - \rho_{eo}e^{\frac{e}{kT_e}(\phi-\phi_o)} = -\rho_i, \tag{4.3}$$

where ρ_{eo} and ϕ_o are reference quantities. However, the underlying steps are identical

for other electron models such as a polytropic relation.

4.2 Newton-Raphson Solver

Aquila uses a Newton-Raphson method to solve Equation 4.3, which rearranges to the form,

$$\nabla^2\phi - Ae^{B\phi} = f, \tag{4.4}$$

and is subject to Dirichlet and Neumann boundary conditions,

$$\phi = \phi_D \quad \text{on } \Gamma_D \tag{4.5}$$

$$\frac{\partial\phi}{\partial n} = g \quad \text{on } \Gamma_N. \tag{4.6}$$

Newton-Raphson is an iterative technique that determines the derivative of a function to obtain a new estimate until the desired level of accuracy is obtained. Generically, the method is used for finding the roots of a function and can be expressed in the following manner,

$$p^{i+1} - p^i = -\frac{f(p^{i+1})}{f'(p^i)}, \tag{4.7}$$

where f is any function and p is the desired root of the function.

Equation 4.4 is discretized using the finite element variational formulation,

$$\int_\Omega v\nabla^2\phi d\Omega - \int_\Omega vAe^{B\phi}d\Omega = \int_\Omega vfd\Omega, \tag{4.8}$$

where v is the variation and Ω is the domain volume. Green's Theorem transforms the first term in Equation 4.8 into two terms containing only first order derivatives,

$$\int_\Omega v\nabla^2\phi d\Omega = -\int_\Omega \nabla v\cdot\nabla\phi d\Omega + \int_{\Gamma_N} gvd\Gamma_N, \tag{4.9}$$

and Poisson's equation becomes,

$$-\int_{\Omega} \nabla v \cdot \nabla \phi d\Omega - \int_{\Omega} v A e^{B\phi} d\Omega = \int_{\Omega} v f d\Omega - \int_{\Gamma_N} g v d\Gamma_N. \tag{4.10}$$

The residual, R, is simply the difference between the expected and the desired value, or the left and right hand side of Equation 4.10,

$$R(v;\phi) = \int_{\Omega} v f d\Omega - \int_{\Gamma_N} g v d\Gamma_N + \int_{\Omega} \nabla v \cdot \nabla \phi d\Omega + \int_{\Omega} v A e^{B\phi} d\Omega. \tag{4.11}$$

Taking the derivative of the residual, DR, with respect to the potential gives

$$DR(v,w;\phi) = -\int_{\Omega} \nabla v \cdot \nabla w d\Omega - \int_{\Omega} v A B w e^{B\phi} d\Omega, \tag{4.12}$$

where w is a second variation. Equation 4.12 provides the means to minimize the residual with the Newton-Raphson method,

$$\phi^{i+1} - \phi^i = DR^{-1}(v;\phi^i)[-R(v;\phi^i)]. \tag{4.13}$$

In this case, the roots of the residual are determined by incrementing the potential. Discrete representations of these quantities are determined by inserting the basis functions as an approximation for the variations across the elements,

$$\phi \approx \sum N_j \phi_j. \tag{4.14}$$

Combining Equations 4.11 and 4.12 with Equation 4.14 yields the vector,

$$R_k = \int_{\Omega} N_k f d\Omega - \int_{\Gamma_N} g N_k d\Gamma_N - \int_{\Omega} \nabla N_k \cdot \nabla \Sigma N_j \phi_j d\Omega - \int_{\Omega} N_k A e^{B\Sigma N_j \phi_j} d\Omega, \tag{4.15}$$

defined at all nodes, k, and the square matrix,

$$DR_{jk} = \int_{\Omega} \nabla N_k \cdot \nabla N_j d\Omega + AB \int_{\Omega} N_k N_j e^{B\phi} d\Omega. \tag{4.16}$$

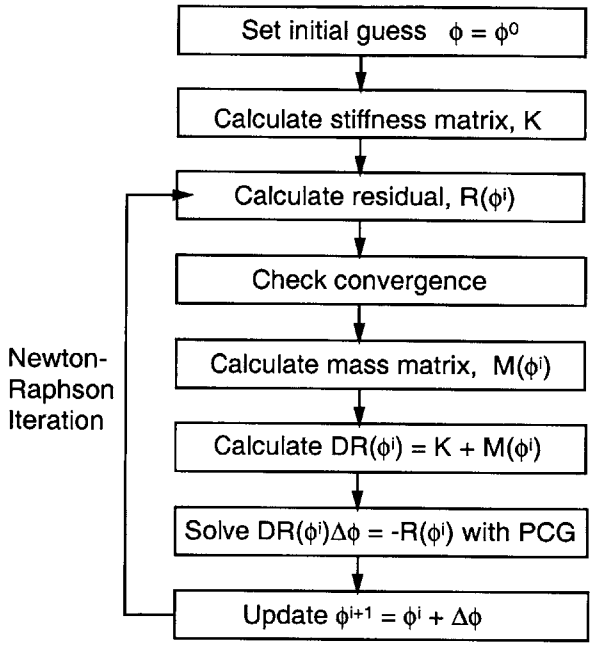

Figure 4-1: Flowchart of Poisson solution method

The first term on the right of Equation 4.16 is the element stiffness matrix, and the second term is a modified version of the mass matrix. Thus, Equation 4.7 becomes a linear system,

$$\underline{\underline{DR}}(\underline{\phi}^i)\underline{\Delta\phi} = -\underline{R}(\underline{\phi}^i). \tag{4.17}$$

This equation can now be solved for ϕ with any of a variety of linear solvers. *Aquila* utilizes a preconditioned conjugate gradient solver (PCG) with diagonal preconditioning.

Figure 4-1 shows the Poisson algorithm. The stiffness matrix depends only on the geometry and is calculated once, whereas the mass matrix must be recalculated every time the potential is updated. In general, the overall scheme only requires 2 to 4 Newton-Raphson iterations of updating the potential before reaching a converged solution, but the embedded PCG solver requires more iterations (approximately 100) to determine the appropriate $\Delta\phi$. This method of solving Poisson's equation has proven to be sufficiently fast for this application, but due to numerical constraints necessitating additional modifications to the basic problem, the non-neutral solution

requires significantly more computation time than the quasineutral methods.

4.3 Stability Issues

Equation 4.3 suffers from stability issues arising from the Boltzmann electron term. This instability occurs when the grid spacing is not sufficient to resolve the variation in potential. A one-dimensional version of the Newton-Raphson solver is used to examine this effect more closely. The notation of Equation 4.4 is used again here, but A, B, and f are set as constant input parameters to the solver as the number of nodes, N, across the domain is varied. Dirichlet boundary conditions are imposed on each end point with the one side being set to 1 and the other to 0. In physical terms, this problem is the equivalent of having a plasma with a constant electron temperature and a constant density of ions between two infinite plates with potentials of 1 and 0.

Figures 4-2 and 4-3 illustrate the effect of decreasing electron temperature while keeping other parameters constant. As B increases, corresponding to a decrease in electron temperature, more grid refinement is necessary to capture the change in ϕ because the potential curve becomes more steep. For instance, in Figure 4-3, the proper shape of the curve is not captured until 32 nodes are placed along the line, and for the cases with 4 and 8 nodes, ϕ dips below zero. Figures 4-4 and 4-5 demonstrate a similar yet less severe effect as A and f increase while B is held constant. The numerical parameters, A and B have a direct relation to the Debye length,

$$d_D = \frac{1}{\sqrt{AB}}. \tag{4.18}$$

The Debye length offers a convenient means of accounting for the observed instability. By examining similar plots for many different values of A and B, it is found the instability arises when the grid spacing, Δx, is greater than the Debye length,

$$\Delta x > d_D. \tag{4.19}$$

Thus, using the Debye length as a comparison to the grid spacing to ensure there is

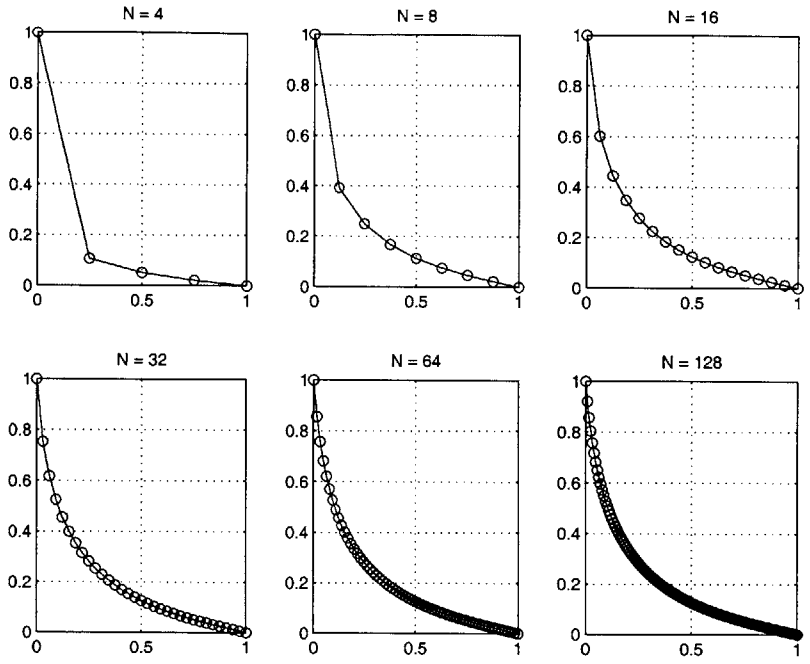

Figure 4-2: Potential solutions from the 1D Newton-Raphson solver for varying grid spacing. Conditions: $A = 1$, $B = 6$, $f = 1$.

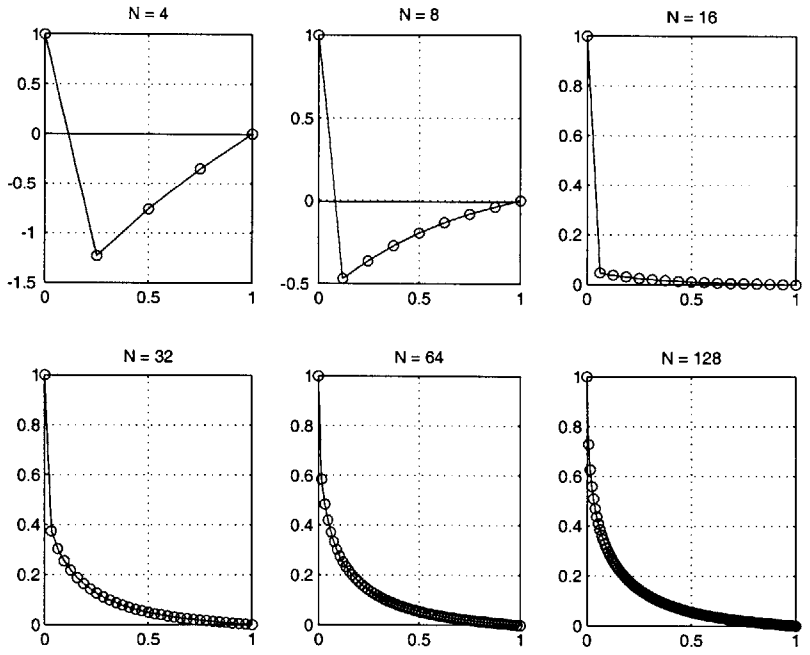

Figure 4-3: Potential solutions from the 1D Newton-Raphson solver for varying grid spacing. Conditions: $A = 1$, $B = 10$, $f = 1$.

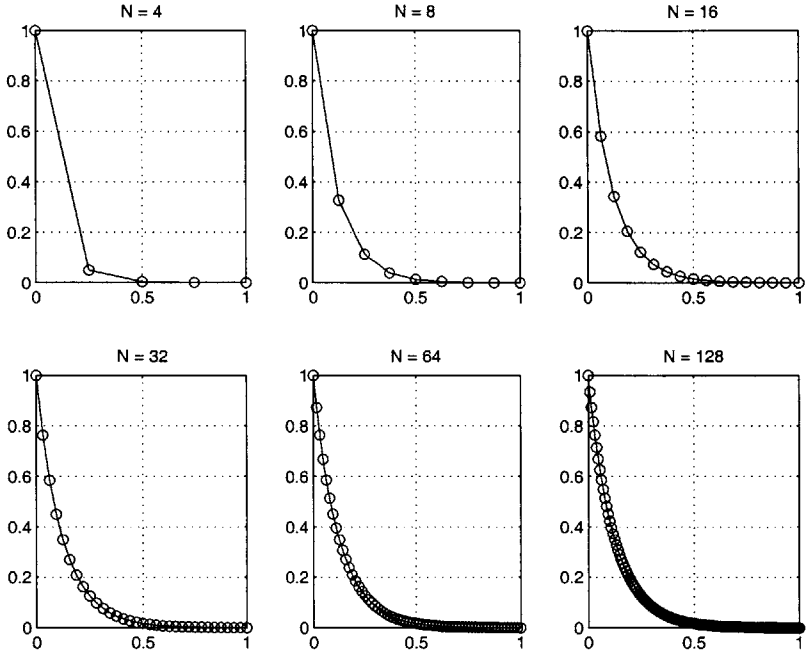

Figure 4-4: Potential solutions from the 1D Newton-Raphson solver for varying grid spacing. Conditions: $A = 128$, $B = .5$, $f = 128$.

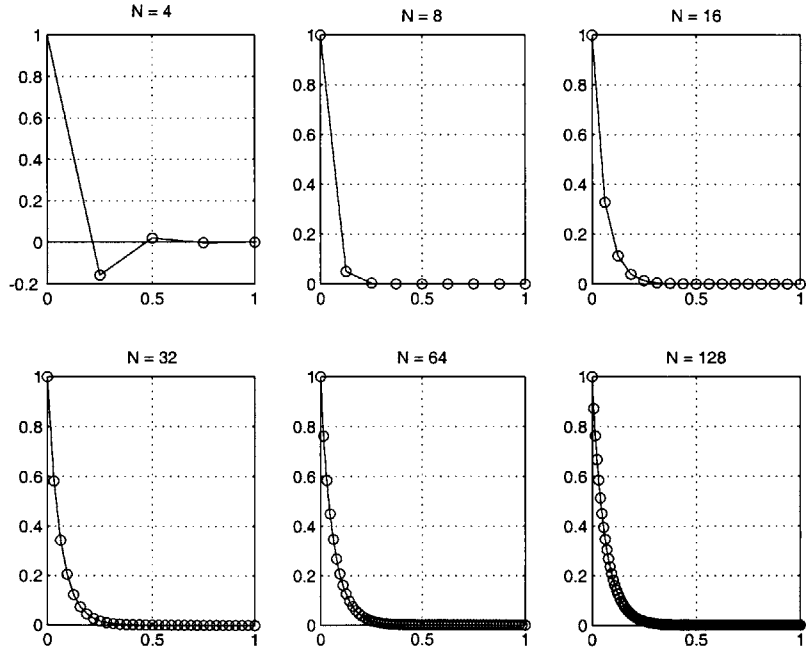

Figure 4-5: Potential solutions from the 1D Newton-Raphson solver for varying grid spacing. Conditions: $A = 512$, $B = .5$, $f = 512$.

Figure 4-6: Potential contour at a snapshot in time showing a numerical instability.

sufficient resolution provides a way to avoid unrealistic potential calculations.

Figure 4-6 demonstrates the consequences of not properly resolving the grid in certain regions of the domain. The figure shows a potential contour plot of a thruster with its thrust direction pointing to the right in the figure. The light-colored point directly above the thruster indicates that this point has a significantly higher potential than that immediately in front of the thruster exit, where the potential should be highest. Though this is a converged solution of Poisson's equation, the region above the thruster has an abnormally large concentration of charge due to the motion of a few simulated particles. Non-physical results such as this example can lead to anomalous electric fields and unrealistic particle motion. These discrepancies also demonstrate a drawback of the PIC method because the finite number of particles in the simulation do not necessarily produce a smooth charge distribution that can accurately represent a continuum approximation.

4.4 Poisson Switch

Solving Poisson's equation at every point in the computational domain is not necessary because in most regions of the plume, the plasma can be considered quasineutral, and potential can be solved for using the methods discussed in Chapter 3. In addition, attempting to solve Poisson's equation in regions where the grid does not resolve the Debye length can produce incorrect results (as in the previous section) that lead to abnormally large particle acceleration. Therefore, a switch has been implemented to allow for the solution of Poisson's equation only in regions determined to be non-neutral and sufficiently refined.

Each node in the domain undergoes two tests to determine if it will be part of a region that needs to be solved with Poisson's equation. The first check directly determines the level of non-neutrality and must account for how the potential was determined in the previous time iteration. If the potential was previously determined by inverting Boltzmann's equation, then the density of ions and electrons will be precisely equal because of the assumption made in using Equation 1.6. Therefore, a direct comparison of densities would be useless, but the Laplacian operator in Equation 1.2 serves as a measure of the non-neutrality in the system in regions that were quasineutral in the previous iteration,

$$N = \left| \frac{\epsilon_o \nabla^2 \phi}{\rho_i} \right|, \tag{4.20}$$

where N is the measure of non-neutrality. If the potential at a node was previously calculated with Poisson's equation, the level of neutrality is directly calculated from the tracked ion density and the electron density resulting from Boltzmann's equation,

$$N = \left| \frac{\rho_e - \rho_i}{\rho_i} \right|. \tag{4.21}$$

N is then compared to a minimum level of neutrality (usually 1 percent) to determine if the non-neutral solver is necessary. The second check is simply a comparison of the Debye length to the average grid spacing at a node. If the Debye length is smaller than

Neutrality Check	$N < \epsilon$	$N > \epsilon$	
Resolution Check	Unnecessary	$d_D < \Delta x$	$d_D > \Delta x$
Solution Method	Quasineutral (Equation 3.9)	Quasineutral (Equation 3.9)	Non-neutral (Equation 4.3)

Table 4.1: Poisson checks. N is a measure of neutrality, and ϵ is the necessary level to assume quasineutrality.

the grid spacing, then Poisson's equation is not used to solve for potential because of stability concerns. Grid nodes that are determined to be quasineutral or that lie in unresolved regions are given a potential value by inverting Boltzmann's equation and are treated the same as Dirichlet boundaries within the actual solver. Table 4.1 shows the progression of checks necessary for a non-neutral case.

Hence, the solution of Poisson's equation for this application requires multiple checks to ensure the obtained potential solution is valid. In the primary plume region, the plasma is found to always be quasineutral, so inverting Boltzmann's equation is a valid method for determining the potential. However, if an obstruction exists in the line of sight of the plume, the wake region will not necessarily exhibit quasineutrality because the ions will be unable to turn sharply enough to enter the wake region, while electrons can, to a limited extent, diffuse into it.

4.5 Non-neutral Results

The non-neutral potential solver is only needed for specialized problems containing large density gradients. The following example demonstrates a circumstance when the quasineutrality assumption breaks down and actually solving Poisson's equation is necessary.

4.5.1 Prandtl-Meyer Fan Analogy

The grid in Figure 4-7 is used to model a problem analogous to the supersonic fluid flow over a sharp corner expanding into vacuum (see explanation in Figure 4-8).

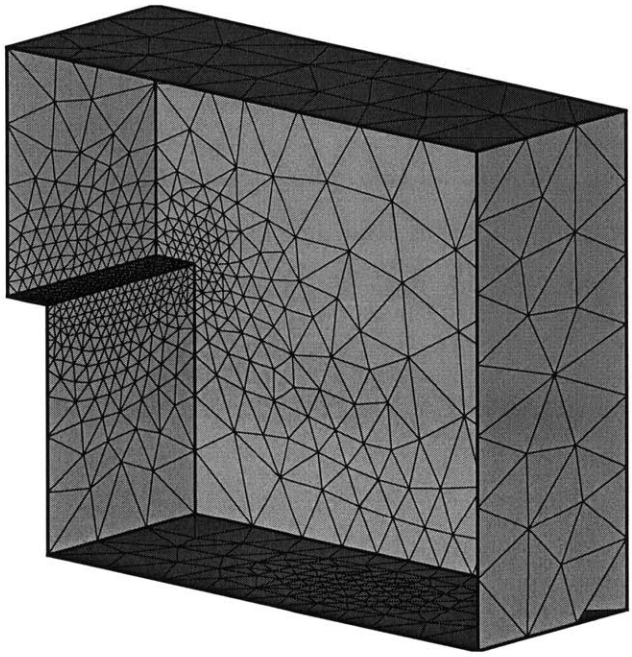

Figure 4-7: Computational grid for expansion into vacuum past a sharp corner.

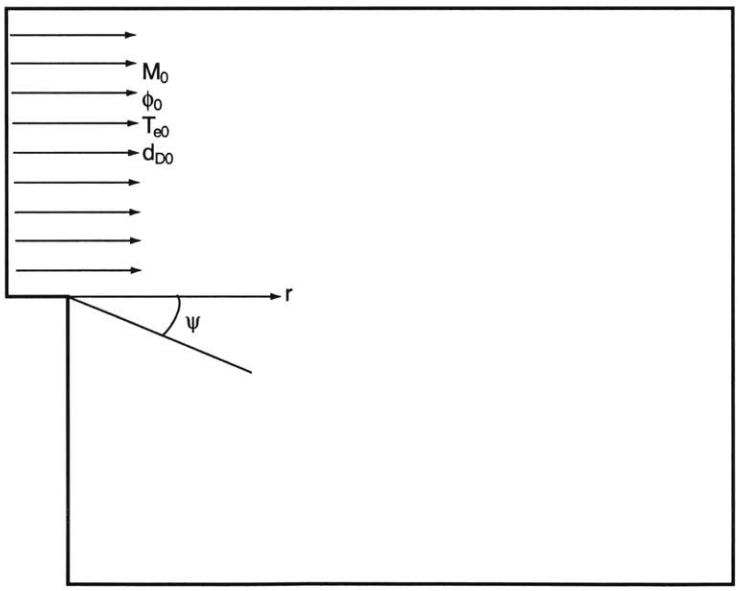

Figure 4-8: Side view of computational space showing plasma flow configuration.

The flow generates a series of expansion waves originating at the corner, known as a Prandtl-Meyer expansion fan. For the purposes of this analysis, a population of only ions (no neutrals) is injected from the region in the upper left-hand side of Figure 4-7. The incoming flow has a Mach number of 3 and a temperature of 2 eV. The ions are given a uniform velocity with small thermal distributions in the x, y, and z directions to avoid particle bunching. The far wall and bottom wall of the grid are treated as perfectly absorbing, while the side walls specularly reflect particles to create the effect of infinite width.

This problem can also be solved analytically. The Mach number, M_i, in a plasma flow is given by,

$$M_i = \frac{v_i}{\sqrt{\gamma \frac{kT_e}{m_i}}}, \tag{4.22}$$

where γ is the exponent used in the polytropic relation and m_i and v_i are the ion mass and velocity respectively. Assuming quasineutrality, analytical expressions based solely on the initial flow conditions can be determined for potential (ϕ), Mach number (M), number density (n_e), and level of neutrality (N):

$$\phi - \phi_o = \frac{\gamma}{\gamma - 1}\frac{kT_{eo}}{e}\left[\frac{2}{\gamma + 1}\left(1 + \frac{\gamma - 1}{2}M_o^2\right) sin^2\left(\frac{\psi_\infty - \psi}{K}\right) - 1\right], \tag{4.23}$$

$$M = \sqrt{1 + K^2 cot^2\left(\frac{\psi_\infty - \psi}{K}\right)}, \tag{4.24}$$

$$\frac{n_e}{n_{eo}} = \left(\frac{1 + \frac{\gamma-1}{2}M_o^2}{1 + \frac{\gamma-1}{2}M^2}\right)^{\frac{1}{\gamma-1}}, \tag{4.25}$$

$$N = \frac{n_{eo}}{n_e}\frac{4\gamma}{(\gamma+1)^2}\left(1 + \frac{\gamma - 1}{2}M_o^2\right)\left(\frac{d_{D0}}{r}\right)^2 cos\left[\frac{2}{K}(\psi_\infty - \psi)\right]. \tag{4.26}$$

Quantities with the subscript o are values for the incoming flow, ψ_∞ is the minimum possible flow turning angle, and K is given by,

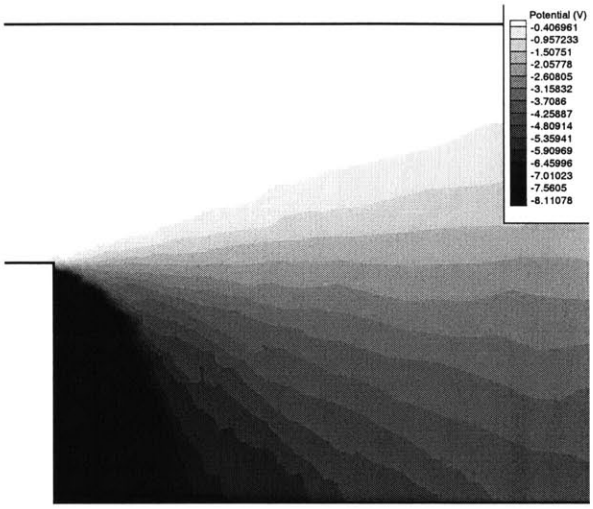

Figure 4-9: Potential contours of expansion assuming $\gamma = 1.3$.

$$K = \sqrt{\frac{\gamma + 1}{\gamma - 1}}. \tag{4.27}$$

The potential profile in Figure 4-9 is produced using the polytropic temperature model and assuming quasineutrality. Figure 4-10 shows how the simulation results compare to the analytical solution. The three curves of simulation results demonstrate how the potential varies along an arc at the specified distance away from the corner. The analytical and simulated results agree until approximately 30 degrees, at which point the nearest radius fails to represent the analytical solution. The analytical solution also provides an estimate of the neutrality of the expansion as a function of the inlet conditions and angle. Figure 4-11 shows where the analytical solution predicts when the neutrality becomes less than 1 percent. The angles where the simulation potential deviates from the analytical potential in Figure 4-10 matches with the angles predicted for neutrality breakdown.

To test the non-neutral solver, a simulation assuming quasineutrality is compared to one that implements the Poisson switch. Both cases are run with a constant temperature of 2 eV on the geometry in Figure 4-7. In spite of the fact that the analytical solution assumes quasineutrality, the non-neutral potential solution actually corresponds more closely with the analytical predictions than the quasineutral

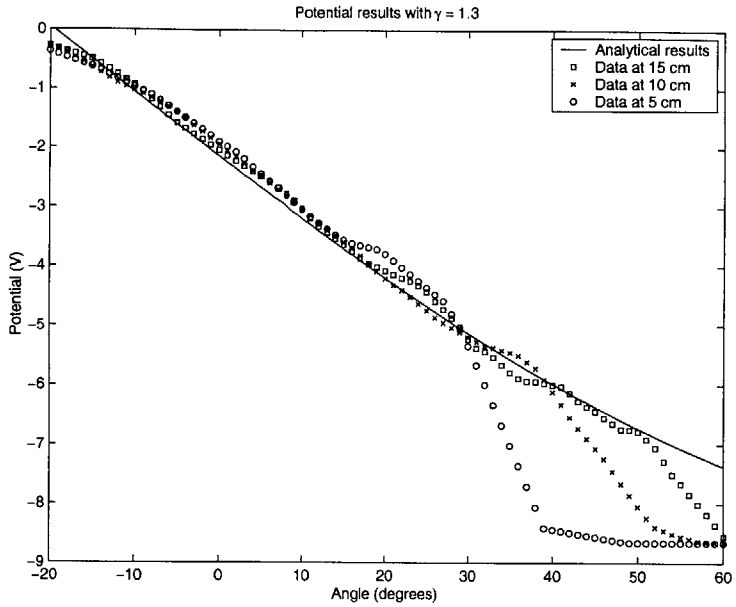

Figure 4-10: Comparison of potential at different radial distances to analytical results for $\gamma = 1.3$ (assuming quasineutrality).

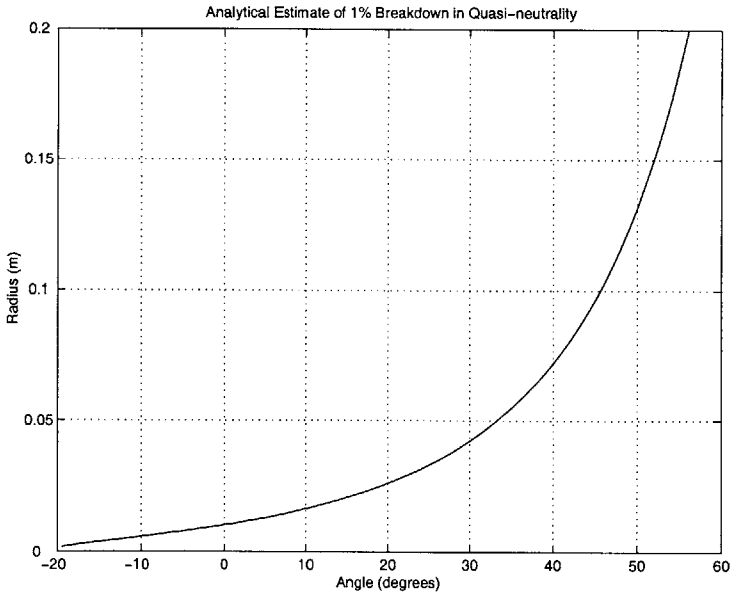

Figure 4-11: Analytical estimate of radial distance where quasineutrality drops below 1 percent as a function of angle for $\gamma = 1.3$.

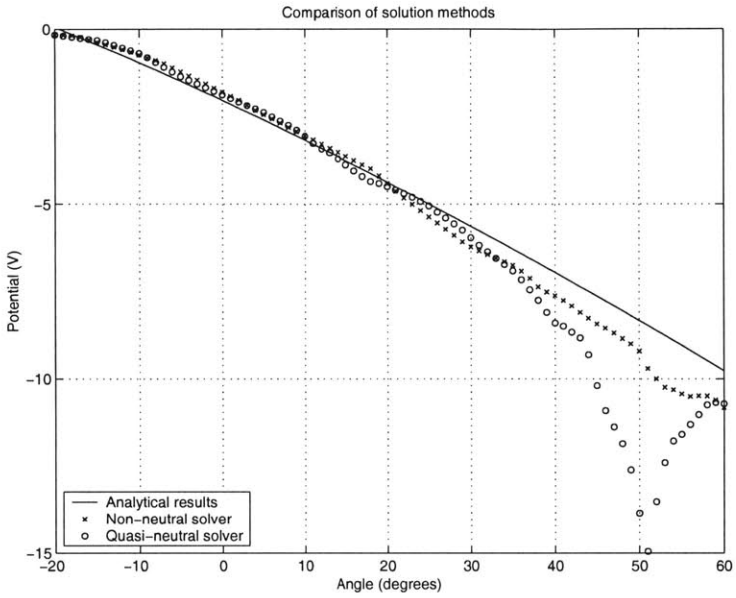

Figure 4-12: Comparison of results from the non-neutral and the quasineutral solvers with analytical results.

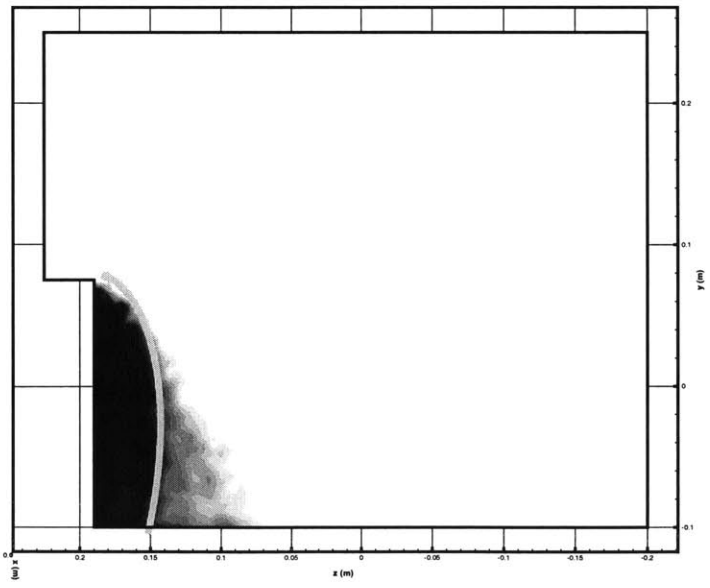

Figure 4-13: Comparison of computed non-neutral region (black portion of contour plot) to predicted 1 percent non-neutral region (gray line).

solution (Figure 4-12). Also, to check if the switching mechanism is determining an appropriate region to solve the non-neutral problem, Figure 4-13 shows the portion of the problem that is solved by Poisson's equation (black region in the lower left-hand corner) compared to the gray line that represents the 1 percent quasineutrality predicted by the analysis. The non-neutral regions coincide well with only a small region of disagreement in the portion expected to be quasineutral.

The non-neutral switch calculations deviate from the analytical results assuming quasineutrality in Figure 4-12, but not the precipitous drop in potential seen in the quasineutral approximation. Agreement in the predicted regions of neutrality between the analysis and the calculation indicates that the switching mechanism in the non-neutral solver is providing a reasonable estimate of where to use Poisson's equation to determine potential. The non-neutral potential solver provides the ability to obtain an accurate potential estimate in regions of largely varying electron density.

4.5.2 Plume Shield

Currently of interest to spacecraft designers is the effectiveness of a shield in the plume to protect sensitive surfaces. The region immediately behind the shield is precisely the wake situation where the non-neutral solver may be necessary. Figures 4-14 and 4-15 show the computational setup used for examining the neutrality properties behind a shield. The refinement of this grid matches that of Figure 3-1 with an additional grid source placed immediately behind the shield.

Five cases are run with the BHT-200 firing into the specified vacuum chamber geometry. Four cases are run with a background neutral density varying from 2×10^{-4} $Torr$ to 2×10^{-7} $Torr$ in order of magnitude increments, and one case is run in true vacuum conditions. Figure 4-16 demonstrates how the non-neutrality in the wake region changes with the background pressure. The contours represent the fraction of simulation time that the regions are determined to be non-neutral. In the perfect vacuum case, the non-neutral wake region is clearly defined, but as the pressure increases, the non-neutral plume regions gradually disappear. This effect is a result of the increased charge-exchange collision frequency with the larger background density.

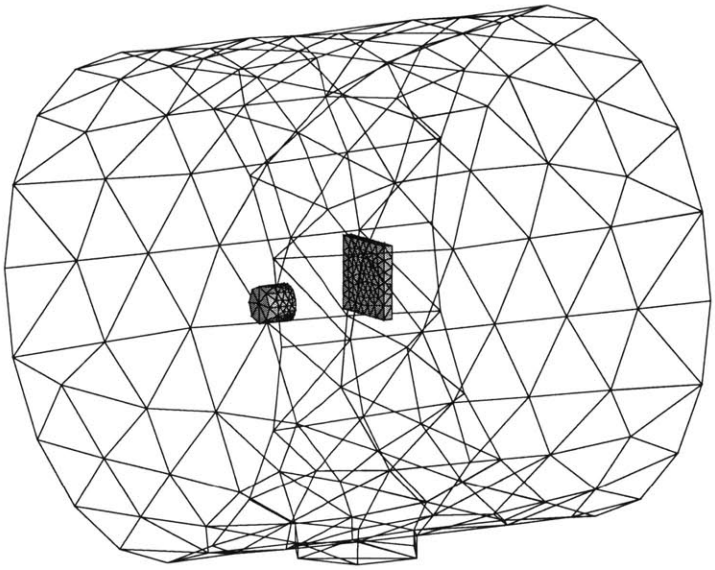

Figure 4-14: Plume shield geometry.

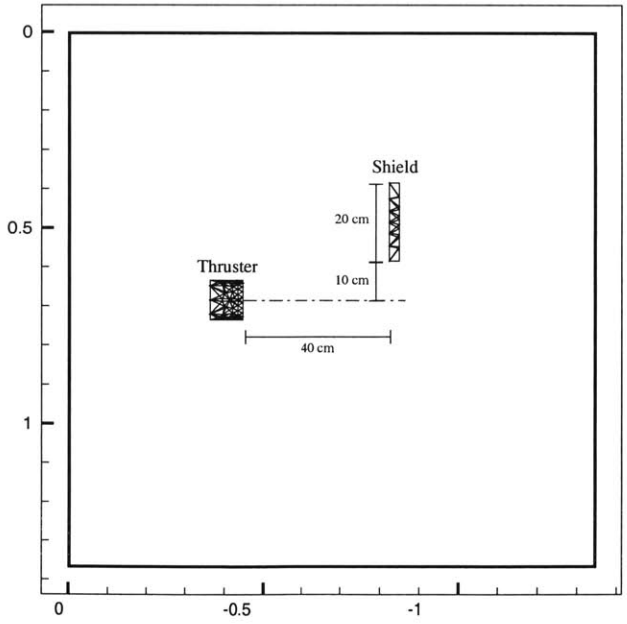

Figure 4-15: Detailed layout of shield geometry.

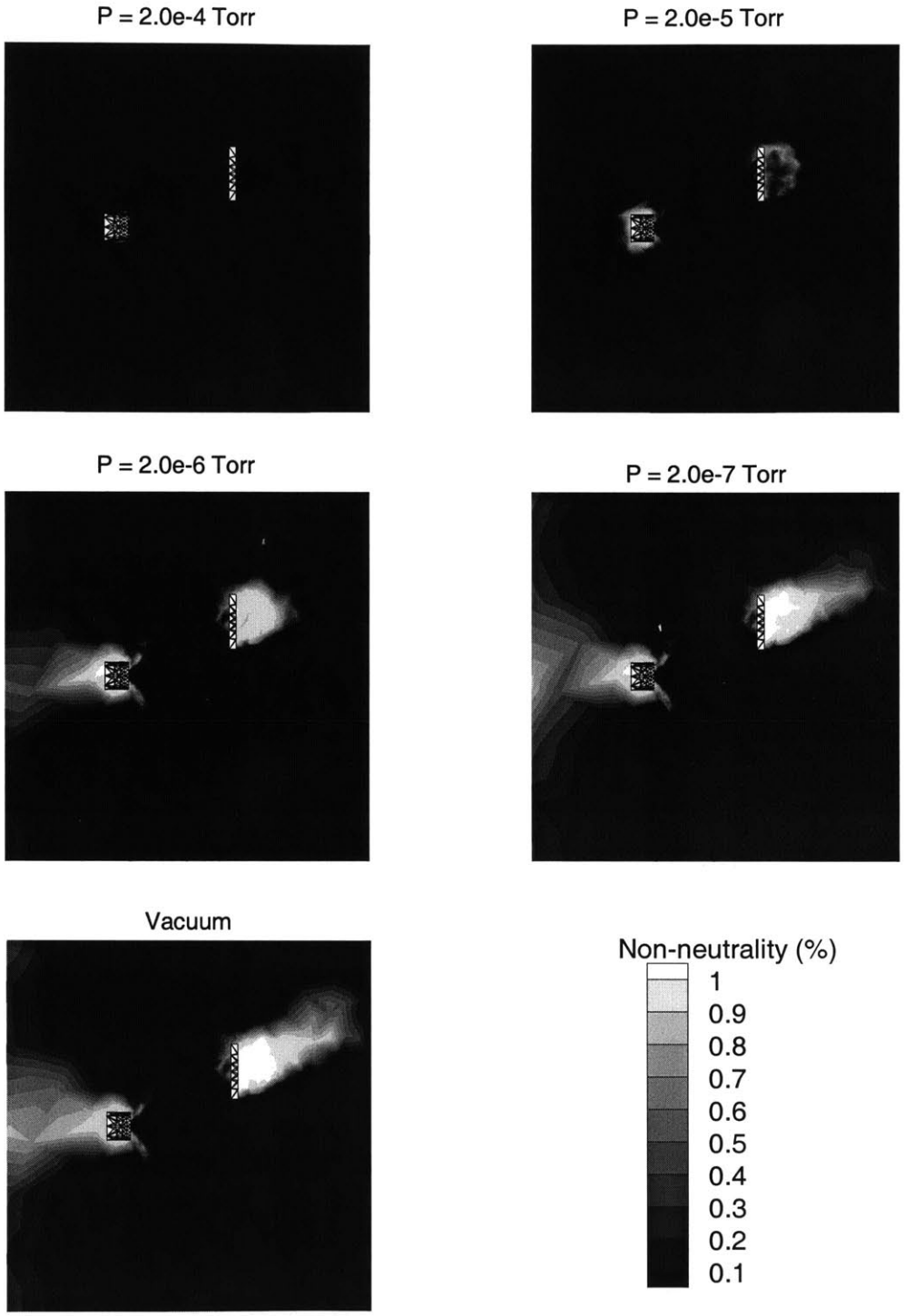

Figure 4-16: Fraction of simulation time that non-neutral solver is used. Dark regions indicate quasineutrality and light regions indicate non-neutrality.

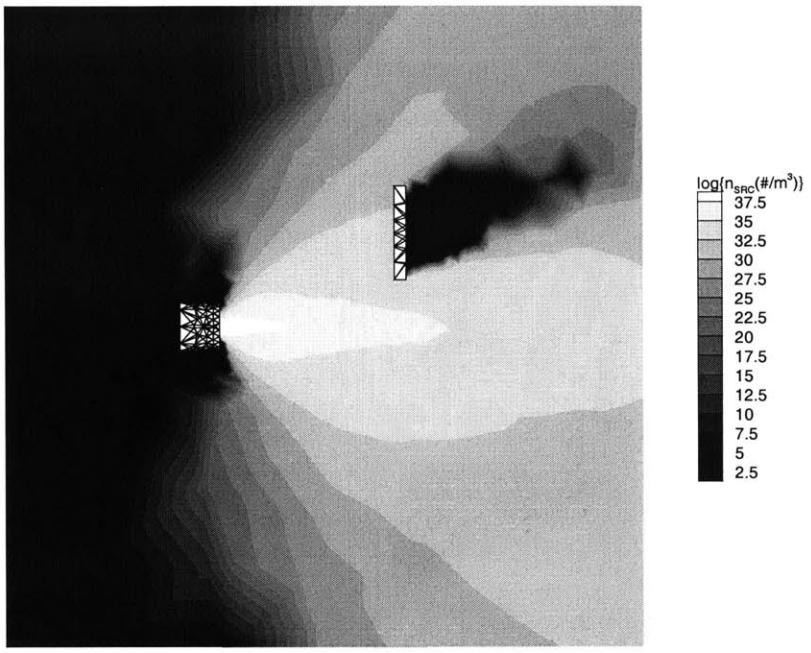

Figure 4-17: Logarithmic contours of source ions for $P = 2.0 \times 10^{-5}\ Torr$.

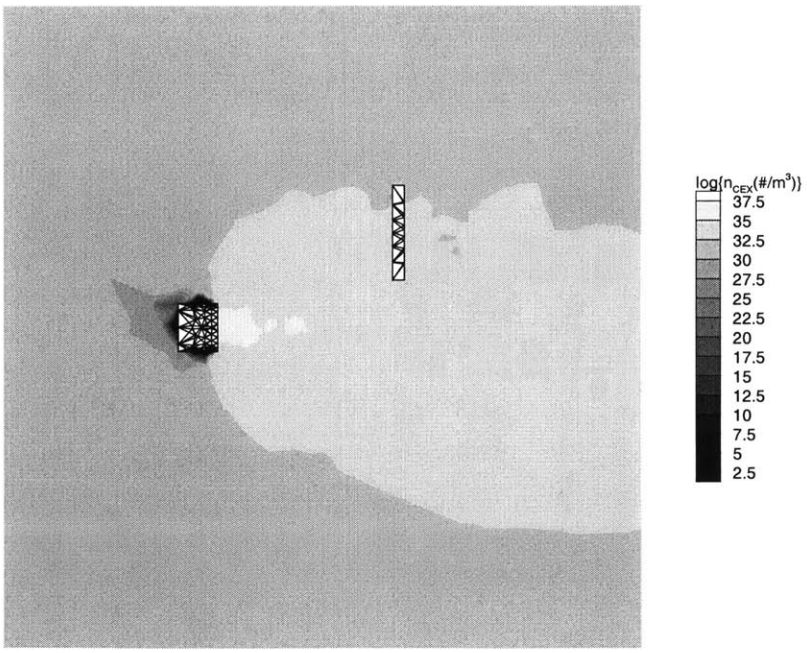

Figure 4-18: Logarithmic contours of CEX ions for $P = 2.0 \times 10^{-5}\ Torr$.

The slow CEX ions fall into the wake more easily than the fast source ions; therefore, the increased CEX density in the cases with a background neutral density leads to quasineutrality behind the shield. Figures 4-17 and 4-18 are logarithmic contour plots of the number density of source and CEX ions respectively for the 2×10^{-5} $Torr$ case. Behind the shield, there are no source ions, but the CEX density behind the shield is comparable to the density in the primary plume region, leading to quasineutrality in the wake. Therefore, the presence of a background pressure has a significant influence on the neutrality properties of the plume.

Chapter 5

Conclusions

5.1 Summary of Results

Aquila, a new hybrid-PIC plume model, has been successfully implemented, expanding upon the capabilities of previous simulations. Geometries can now be defined using CAD solid modeling packages, and the corresponding unstructured, tetrahedral grid is automatically generated. Tetrahedral grids provide the means to accurately represent complicated geometries for examining surface interactions. A non-neutral potential solver accounts for specialized problems where quasineutrality breaks down. Results using both the quasineutral and the non-neutral potential solvers have served to verify the fundamental operation of *Aquila*.

- Simulated current density results compare favorably with experimental vacuum chamber data. With a constant temperature model, the centerline region has a significant peak that does not agree with the experimental data, but by increasing the value of γ in the polytropic model, this peak is eliminated. Increasing γ provides a stronger radial potential gradient that pushes particles from the centerline, thus reducing the current density. Unfortunately, although results at 25 *cm* agree along the centerline with a value of 1.3 for γ, the current density falls below the experimental data by 47 *cm*. All simulations underpredicted the current density in the wing region dominated by CEX ions.

- With a source model scaled to the experimental operating condition of 250 V discharge voltage, potential results show strong agreement with experimental data. The overall potential drop of 7 V between the centerline and 90 degrees is found with both the constant and polytropic temperature models at 25 cm. At 47 cm, the polytropic model predicts the expected 4 V difference while the constant temperature run shows a larger drop.

- Running *Aquila* with true vacuum conditions provides evidence that the separate portions of the code are operating as expected. Source ions are forming the familiar expanding plume pattern. The number density of ions resulting from CEX collisions shows a distinctive wing region, and ions resulting from elastic collisions are found in the expected 60 degree region off centerline.

- Based on simulation results mimicking the analytical solution of a plasma flow expanding into vacuum, the non-neutral potential switch is not only providing more reasonable potential approximations than assuming quasineutrality, but is also predicting the correct region of the flow where neutrality breaks down.

5.2 Suggestions for Future Work

Although *Aquila* expands upon the capabilities of plume codes such as *Quasi3*, certain aspects still need improvement.

- Hybrid weighting functions would provide continuous electric fields and avoid any bunching of particles. Although not a major concern for tetrahedral grids, continuous electric fields would certainly increase the accuracy of the field calculation.

- Dealing with unresolved, yet non-neutral, regions of the plume needs to be improved. Currently, the non-neutral solver simply inverts Boltzmann's equation to obtain the potential, but perhaps a better approximation for this potential can be determined without imposing a quasineutral solution. Because non-

neutral regions are only present in regions of extremely low density, imposing an electron density of zero might be a reasonable assumption.

- A Monte Carlo collision model (MCC) would remove the necessity that all particles of a single species must have the same weighting. Allowing for variable species weighting would enable a scheme such as the one introduced by Fife [8] where particles are combined in dense regions and separated in sparse ones. This flexibility would also provide a more accurate charge distribution and alleviate some of the resolution problems in the non-neutral solver.

- Adaptive grid refinement would enable the grid to change during the simulation, dynamically conform to the structure of the plume, and locally refine non-neutral regions.

- Ionization collisions in the plume should be included to account for the extra current density measured experimentally. Perhaps this creation of low energy ions in the plume will partially explain the discrepancies in the wing region of the current density profile.

- Tools for increasing the speed of the calculation would help the simulation to reach a steady state in a more reasonable time. Currently, collaborators at the Air Force Research Lab are adding neutral subcycling to assist the heavy neutrals in reaching a steady state distribution quickly. Techniques such as orbit-averaging [7] where particle charge is calculated over several time-steps rather than instantaneously would enable fewer tracked particles to accurately represent the overall distribution. Moving the simulation to parallel processors would distribute the work load to several machines and increase the computational speed.

Bibliography

[1] Azziz, Y., Instrument Development and Plasma Measurements on a 200-Watt Hall Thruster Plume. Master's Thesis, Massachusetts Institute of Technology, September 2003.

[2] Birdsall, C. K. and A. B. Langdon, *Plasma Physics Via Computer Simulation.* Institute of Physics Publishing, Bristol, 1991.

[3] Boyd, I. A., "Hall Thruster Far Field Plume Modeling and Comparison to Express Flight Data." AIAA 2002-0487, AIAA 40th Aerospace Sciences Meeting, January 2002.

[4] Celik, M., M. M. Santi, S. Y. Cheng, M. Martinez-Sanchez, and J. Peraire, "Hybrid-PIC Simulation of a Hall Thruster Plume on an Unstructured Grid with DSMC Collisions." IEPC-030134, 28th International Electric Propulsion Conference, March 2003.

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

[6] Cheng, S. Y., Computational Modeling of a Hall Thruster Plasma Plume in a Vacuum Tank. Master's Thesis, Massachusetts Institute of Technology, February 2002.

[7] Cohen, B. I., T. J. Williams, "Semi-Implicit Particle Simulation of Kinetic Plasma Phenomena." *Journal of Computational Physics*, 97:224-234, 1991.

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

[9] Fife, J. M., M. R. Gibbons, W. A. Hargus, D. B. VanGilder, D. E. Kirtley, and L. K. Johnson, "The Development of a Flexible, Usable Plasma Interaction Modeling System." AIAA 2002-4267, 38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference, July 2002.

[10] Jones, W. T., "An Open Framework for Unstructured Grid Generation." AIAA 2002-3192, 32nd AIAA Fluid Dynamics Conference and Exhibit, St. Louis, Missouri, 2002.

[11] MacNeice, P., Particle-Mesh Techniques. NASA Contractor Report 4666., July 1996.

[12] 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.

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

[14] Sonnendrucker, E., J. J. Ambrosiano, and S. T. Brandon, "A Finite Element Formulation of the Darwin PIC Model for Use on Unstructured Grids." *Journal of Computational Physics*, 121:281-297, 1995.