COMPUTATIONAL METHODS FOR THE DIRECT SIMULATION MONTE CARLO TECHNIQUE WITH APPLICATION TO PLUME IMPINGEMENT

A Dissertation
Presented to the Faculty of the Graduate School
of Cornell University
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

by

Keith Christopher Kannenberg

May 1998

COMPUTATIONAL METHODS FOR THE DIRECT SIMULATION MONTE CARLO TECHNIQUE WITH APPLICATION TO PLUME IMPINGEMENT

Keith Christopher Kannenberg, Ph.D.
Cornell University 1998

The impingement of gas plumes from small thrusters on spacecraft surfaces is modeled using the direct simulation Monte Carlo (DSMC) technique. Strategies for improving the efficiency and resolution of DSMC simulations are presented. These methods include variable scaling of simulation parameters and computational grid design. Computational cost of a simulation can be reduced by as much as two orders of magnitude using these methods.

A parallel, three-dimensional implementation of the DSMC method is described. A flexible, cell-based grid methodology is employed which allows the use of structured, unstructured or hybrid grids. This provides the ability to handle complex geometric configurations. An efficient particle tracing scheme is used for particle movement which eliminates the need for an explicit sorting step. Strong numerical performance and high parallel efficiencies are obtained.

Several impingement problems are investigated in order to test the accuracy and performance of the DSMC method and implementation for plume flows. The first problem is a nitrogen plume from a resistojet nozzle impinging on an axisymmetric body. The plume flow is started with exit plane data from a simulation of the diverging section of the nozzle. Good agreement is found with experimental data for surface pressure over a range of body locations.

A free jet of nitrogen impacting on an inclined flat plate is examined as a three-dimensional test case. The three-dimensional code is validated through comparison with an axisymmetric simulation of the case where the plume axis is normal to the plate. Good agreement is found with experimental measurements of surface pressure, shear stress and heat transfer for several plate orientations. Free molecular analysis is found to over predict surface properties while providing good qualitative agreement.

The plume of a hydrazine thruster mounted on a model satellite configuration is investigated to demonstrate the ability to compute complex configurations. Surface properties and integrated impingement effects are calculated on a solar array panel. The plume is shown to transfer a significant fraction of the thruster's momentum and energy to the array. Free molecular analysis is less accurate as a result of multi species and boundary layer effects.

Biographical Sketch

The author was born in New York City in 1972. In August of 1989 he made the journey upstate to Ithaca, NY, in order to attend Cornell University. He earned a Bachelor of Science degree in Mechanical Engineering in January of 1993. After graduation he opted to remain at Cornell for graduate studies in the field of Aerospace Engineering. A very brief stay in the Master of Engineering program was followed by a switch to the M.S./Ph.D. program. Three years later, in January of 1996, he was awarded a Masters of Science degree. He then spent another two and a half years in Upson Hall in order to earn his Doctorate in May of 1998.

To Mom. You'd be proud of me.

Acknowledgements

The work presented in this thesis was supported by NASA Lewis Research Center under grant NAG3-1451 and by the Air Force Office of Scientific Research under grant number F49620-96-1-0210. Computational resources were provided by the Cornell Theory Center and the National Aerodynamic Simulation program at NASA Ames Research Center.

I've been lucky enough to spend almost nine years at Cornell of which more than five have been spent in graduate school. In this time many people have given me their friendship, support and/or guidance. The following is a short and almost certainly incomplete list of those to whom I owe my thanks for my time on the hill:

My advisor, Iain Boyd.

My special committee members, P.C.T. deBoer and Stephen Vavasis.

My current officemates (and friends), Dan Karipides, Doug VanGilder and Jitendra Balakrishnan. Thanks for your help with my work and, more importantly, for helping to keep me sane (almost).

My former officemates, Shankar Subramanian, Tom Dreeben, Walt Welton and Wyatt Phillips.

Thanks to the many other grad students in the M&AE department who I've spent time with, Some of these people include Tim Fisher, Elizabeth Griffith, Heather Relation, Paul Van Slooten, Jamal Yusof.

Thanks to Tobe Cookingham for always being around to talk to when I needed it. You always seemed to know when something wasn't quite right.

To Sharon Calhoun for being a second Mom throughout grad school.

To Dad for all your support and patience through the years.

And to God for always being a "very present help in trouble."

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation - Spacecraft Plume Impingement

Spacecraft in orbit require propulsion systems for various functions including attitude control and station keeping. Attitude adjustments may be necessitated by such mission requirements as the pointing of antennas or scientific instruments. Power requirements may require attitude adjustments to ensure optimal sun coverage of solar arrays. Deceleration caused by orbital drag must be accounted for in order to maintain a stable orbit.

Low thrust rockets are often employed to meet these on orbit requirements. The impulse needed for these tasks is generally small so that small thrusters can handle the task. An array of thrusters placed around the spacecraft firing in different directions is needed to provide the full range of orbital and attitude control. Each task uses a number of thrusters firing in unison. The use of more than one thruster allows greater control of the forces and torques exerted on the spacecraft.

Many different types of control thrusters have been or will soon be flown. Table 1.1 shows typical thrust levels and specific impulses for different types of thrusters [27]. The simplest is a cold flow thruster where thrust is generated by expanding a cold gas such as helium through a nozzle. This type of thruster produces low thrust and has a low specific impulse. However, it is also the simplest to design and operate.

Chemical rockets generate energy through chemical reactions between a propellant and an oxidizer. The hot gas is then expanded through a nozzle to produce thrust. There are many variations of this type including choice type of propellant (mono- or bipropellant). These devices have reasonably high thrust as a result of a large mass flow but low specific impulses. Chemical control thrusters are generally fired for a short duration during a single firing.

Electric propulsion devices use electrical power in various ways to accelerate a propellant. Electrothermal thrusters use electrical power to heat the propellant which is then expanded through a nozzle. These include resistojets which heat the gas using electric heaters and thermal arcjets where an electrical arc current is used. In electrostatic devices or ion thrusters the propellant is ionized by electron bombardment and then accelerated across a potential. Electromagnetic thrusters generate a plasma and use electromagnetic fields to accelerate the propellant. Examples include solid pulsed plasma, hall thrusters and arcjets. Electric thrusters tend to have high specific impulse, low mass flow rates and low thrust. As a result of this low thrust, typical operation involves long duration or pulsed burns.

When a control thruster is fired, a plume of exhaust gases is produced. While the bulk flow is directed along the axis of the thruster, the plume will also expand radially. In a vacuum environment, this radial expansion can be very broad. There will also be a noticeable amount of backflow gas which flows behind the thruster. Figure 1.1 shows a schematic of a thruster plume.

Under many conditions the plume gases can impinge on spacecraft surfaces. Backflow gases can strike the spacecraft directly. Peripheral parts of the spacecraft such as solar arrays may be located within the sides of the main plume. Close maneuvers involving multiple spacecraft can result in direct impingement by the main plume. Figure 1.2 shows each of these types of impingement. In each of these impingement scenarios there is a transfer of mass, momentum and energy from the plume back to the surface.

Impingement of a plume on a spacecraft can have a number of undesirable effects. Momentum transfer results in additional forces and torques which affect the attitude of the spacecraft. Depending on the orientation of the spacecraft, its component surfaces and the plume, these forces can be a significant fraction of the total thrust. The propulsion system must then compensate for these forces at the cost of additional propellant and possibly electrical power. Increasing the amount of propellant required means either the lifespan of the spacecraft is decreased or more propellant must be carried, reducing the amount of mass available for payload.

When a plume molecule strikes the spacecraft it transfers kinetic and internal energy to the surface. In most cases this results in heating of the surface. Thermal control is an important part of spacecraft design. Many electronic components can only function properly in a range of temperatures. The thermal loads caused by plume impingement must be accounted for and dissipated by the thermal control system.

The flux of particles striking a surface can have detrimental effects on the surface. Many of the gas species which are used as propellants can be classified as contaminants. An example is ammonia which is a component species in hydrazine chemical rockets. Deposition of these species on sensitive surfaces such as solar arrays can cause degradation which reduces the useful lifetime of the component and thus the spacecraft. The impact of high energy charged particles in electric thruster plumes can also damage spacecraft components.

Impingement effects are an important consideration in the design of a spacecraft. Propulsion requirements should be met with a minimum of impingement in order to insure optimal performance and lifetime. Control thrusters must be located and firing patterns designed so that the plume effects are minimized. The plume impingement that does occur must be modeled and accounted for in the design of the guidance, propulsion, thermal and other systems. Accurate modeling of thruster plumes can result in better, more efficient designs which utilize reduced engineering safety factors. If as a result of plume modeling a spacecraft's lifetime can be increased through lower propellant use or reduced degradation the overall economic cost of the spacecraft will be reduced.

1.2 Approach

The plume of a thruster firing into a vacuum rapidly expands to extremely low densities. Under these rarefied conditions the number of inter-molecular collisions is insufficient to maintain local equilibrium in the flow. As a result, the assumptions of continuum fluid mechanics become invalid and the gas cannot properly be modeled as a continuum fluid. In order to accurately model the flow, nonequilibrium effects must be included. A microscopic approach which considers the particle nature of a gas is appropriate.

Table 1.1. Thrust and specific impulse ranges for thruster types.

Thruster Type	Thrust	Specific Impulse (sec)
Cold Flow	< 5 mN	30–200
Chemical	0.03–100 N	200–230
ResistoJet	2–100 mN	200–300
Thermal Arcjet	2–700 mN	400–1500
Ion Thruster	0.01–200 mN	1500–5000
MPD Arcjet	2–2000 mN	1000–8000

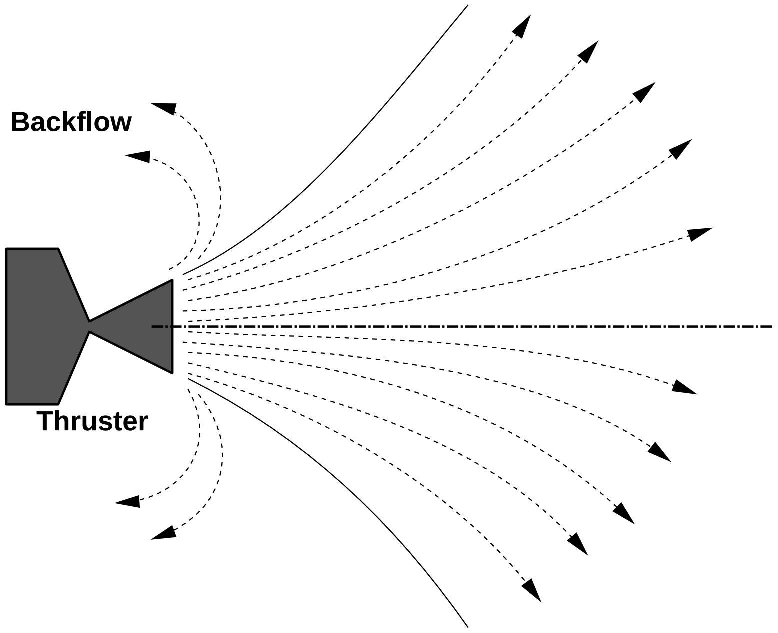

Figure 1.1. Thruster plume schematic.

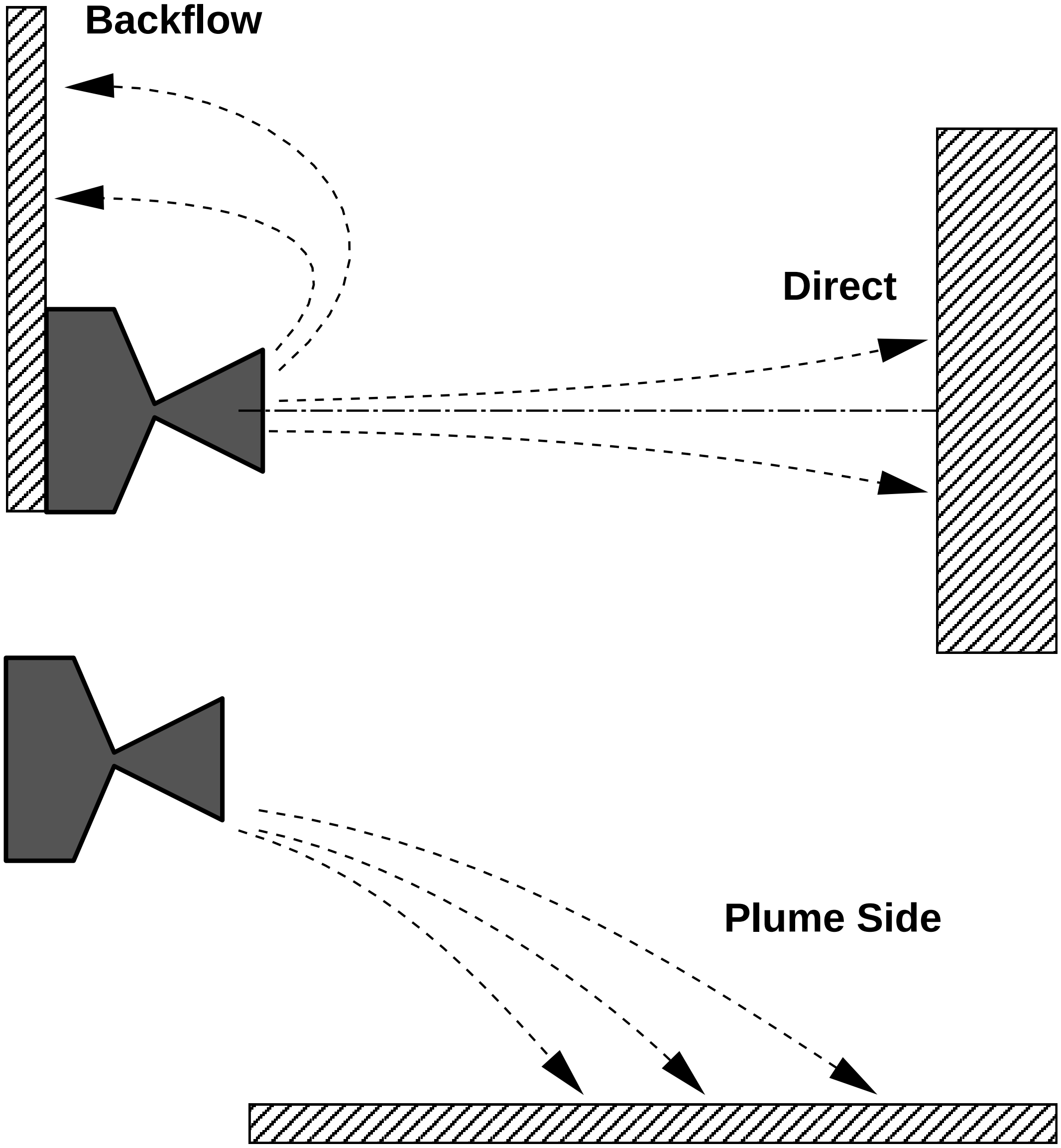

Figure 1.2. Schematic of possible impingement interactions with spacecraft surfaces. Backflow and direct impingement (top) and lateral impingement (bottom).

The direct simulation Monte Carlo (DSMC) method [4] is employed for the numerical simulation of plume flows in this thesis. DSMC models a gas statistically on the microscopic level using particles which represent the molecules of a real gas. Nonequilibrium effects in the flow field and boundary interactions are directly simulated at the molecular level.

The DSMC method has been successfully applied to a diverse range of nonequilibrium gas flows. Examples include atmospheric re-entry of hypersonic blunt bodies [8], microchannel devices [23] and materials processing [15].

In the field of propulsion, a wide variety of devices have been simulated. Previous studies have examined cold flow thrusters [9], chemical rockets [11], resistojets [2,10], arcjets [7] and ion thrusters [29]. Simulations of these devices generally have been begun in the vicinity of the nozzle throat and extend downstream into the near field plume. DSMC simulations have shown good agreement with experimental measurements taken within the nozzle, at the exit plane and in the near plume.

Plume impingement effects have been considered in a number of studies. Giordano and Niccoli simulated the near field plume of a bipropellant chemical thruster to be used for orbit insertion of ESA's MSG satellite [16]. Simulations were used to assess the heat transfer caused by the backflow during spacecraft design. Ivanov, et al, modeled the near and far field plumes of an argon cold flow thruster [18]. Their results show reasonable agreement with experimental measurements of plume density and pressure.

Simulations of three-dimensional interactions between plumes and a spacecraft have been performed. Ivanov, et al, used a multizone Navier-Stokes, DSMC and Test Particle Monte Carlo approach to simulate plumes and impingement effects [19]. Lumpkin, et al, performed simulations of impingment effects occurring during Shuttle-Mir docking [22]. Rault examined the interaction between a thruster plume and the orbital environment and the resultant impingement effects [24].

1.3 Goals and Overview

The goal of this work is to develop and validate the capability to simulate complex plume impingement flows using the DSMC technique. In order for such a capability to be useful from an engineering standpoint, it must be accurate, numerically efficient and flexible enough to be applied to a wide variety of flows.

The DSMC method tends to be computationally expensive. A large number of particles are needed to accurately represent a flow field and the cost of a simulation is, to first order, proportional to the number of particles used. The cost can be particularly high when large, complex problems are considered. However, finding ways to reduce this cost is an important part of developing a modeling capability.

Parallel computing is a powerful tool for tackling large computational problems. By splitting the computational work between different processors, the real time required for a simulation can be reduced dramatically. An ideal parallel algorithm reduces the calculation time by a factor equal to the number of processors employed. The DSMC algorithm parallelizes well. Several previous studies have used parallel implementations of the DSMC method for plume impingement studies [19,22,24], but most have utilized only a small number of processors. The numerical results in this work are computed using a large scale parallel implementation of the DSMC method which can operate with any number of processors [13].

Several techniques for reducing the overall cost a DSMC simulation are examined. Their relative benefits are measured in the context of impingement flows. Each method also has an effect on the accuracy of a simulation and this is assessed. A methodology for efficiently simulating impingement flows is developed.

The flow field of a plume impinging on a spacecraft generally is three-dimensional. A three-dimensional implementation of the DSMC method is required to simulate such flows. It is particularly important for this implementation to be numerically efficient due to the large size of three-dimensional calculations. It is also important that it be flexible to accommodate the complex geometries found in engineering problems. The development of a three-dimensional version of an existing DSMC code is discussed.

Validation of the DSMC method for plume flows is performed by simulating simple impingement problems for which experimental date are available. Measurements of flow field properties within a plume are not widely available, especially for three-dimensional flows. However, surface impingement properties, such as pressure and heat transfer, are available. DSMC predictions of surface properties are compared with experimental data. It is assumed that correct prediction of surface properties implies correct modeling of the plume itself.

An additional estimate of surface properties can be obtained analytically if the plume is assumed to be free molecular at the surface. DSMC results are compared with free molecular theory and the accuracy of these estimates is assessed.

An axisymmetric problem is considered for initial validation of DSMC. The impingement of a resistojet plume on an conical body located at the plume axis is simulated. Then a simple three-dimensional problem, the impingement of a free jet on an inclined flat plate, is considered. Finally, a representative satellite-thruster configuration is simulated as a demonstration result.

In the scope of this work only the fluid mechanics of plume flows are considered. The goal is to develop the capability to simulate these flows. This is facilitated by the consideration of neutral, non-reactive flows and simple surface models. Once the capability is established and verified, more advanced physical effects such as chemistry, surface models or electric field effects can be added.

Chapter 2

DSMC Method and Implementation

2.1 Direct Simulation Monte Carlo Method

2.1.1 Overview

The direct simulation Monte Carlo method (DSMC) is a particle method for the simulation of nonequilibrium gas flows [4]. The gas is modeled at the microscopic level using particles which each represent a large number of physical molecules or atoms. The physics of the gas are modeled through the motion of particles and collisional interactions between them. Mass, momentum and energy transport are considered at the particle level. The method is statistical in nature. Physical events such as collisions are handled probabilistically using largely phenomenological models which are designed to reproduce real fluid behavior when examined at the macroscopic level.

A microscopic approach to the simulation of gas flows can be advantageous due to the ability to capture nonequilibrium effects. Implicit in the use of continuum fluid mechanics is the assumption that the gas is in local equilibrium. That is, the velocities and energies of the gas particles are distributed according to equilibrium distribution functions. This allows the gas to be modeled as a continuum fluid using such thermodynamic quantities as temperature. In order to maintain equilibrium there must be sufficient collisions occurring in the gas. In many flows of interest this is not the case and the gas cannot be modeled as a continuum fluid without neglecting nonequilibrium effects. By considering the gas on the microscopic level as a collection of discrete particles the DSMC method is able to capture nonequilibrium effects directly.

The nonequilibrium nature of a problem can be measured using the Knudsen number. This nondimensional parameter is the ratio of the mean free path length in the gas to the length scales of the flow:

$$\mathrm{Kn} = \lambda/L \tag{2.1}$$

The mean free path is the average distance travelled by a gas particle between collisions. If the length scales of the flow are comparable to the mean free path then there will be insufficient collisions to maintain equilibrium. Flows with Knudsen numbers on the order of 10^{-3} or smaller are continuum flows and may be modeled using more conventional fluid mechanics. A Knudsen number greater than 10^1 indicates the flow is essentially free molecular, where the number of intermolecular collisions is small enough to be neglected. Between these extremes are transitional flows where collisional behavior is important but there are insufficient collisions to ensure local equilibrium. Examples of such transitional flows include the rapidly expanding gas plume from a thruster firing into a vacuum or the shock interaction in front of a hypersonic vehicle.

The primary application of DSMC is to transitional flows. The method can be applied to continuum flows but is more expensive numerically than continuum computational fluid mechanics techniques and provides no additional accuracy. For free molecular flows the DSMC algorithm simplifies to the test particle Monte Carlo scheme.

The DSMC method constitutes a probabilistic, numerical solution to the Boltzmann equation for gas flows [4, pages 208–210]. The Boltzmann equation is an integral differential equation which describes the rate of change of the single particle distribution function for particle velocity. It considers changes in velocity due to convection, acceleration due to external forces and collisions. By tracking representative particles through space and considering collisions between particles the DSMC method directly simulates the mechanics of the Boltzmann equation.

2.1.2 Particle Representation

Particles in a DSMC simulation are considered to represent a large number of real gas particles. Each simulation particle can be considered either an average of the real particles it represents or as a representative sample from the particle distribution function. The action of a large number of particles over a period of time simulates the distribution function.

Particles are generally considered to be point particles or spheres with associated properties such as species, velocity and internal energy. The actual shape of molecules and the associated three-dimensional force fields are generally not considered for computational simplicity. Instead, spherically symmetric fields are used. Particles which represent molecules are assigned rotational and vibrational energies to account for these degrees of freedom.

Collisions between particles are handled statistically using models for the collision cross section and the scattering angle. The collision model which has been used in this study is the Variable Hard Sphere model (VHS) [3]. This model treats the particles as solid spheres of variable diameter which scatter isotropically. The effective particle diameter is proportional to a power of the relative velocity in a collision. The collision cross-section based on this diameter has the following form:

$$\sigma = \pi d^2 \ = \pi {d_{ref}}^2 \left(\frac{g}{g_{ref}} \right)^{-2\omega} \tag{2.2}$$

The parameter ω is obtained from viscosity data for the gas using the following relation:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{\omega + \frac{1}{2}} \tag{2.3}$$

2.1.3 Algorithm

The DSMC technique models a gas by following the trajectories of gas molecules and computes the effects of collisions between particles. One of the fundamental assumptions of the DSMC method is that particle movement can be decoupled from collisional behavior. Particles then move along straight line paths with discrete changes in velocity and energy caused by collisions. In order to accomplish this, the simulation is broken up into time steps the magnitude of which are small compared to the average time between collisions. In this small interval of time, these two processes can realistically be treated as independent.

A complete DSMC simulation is a series of discrete time steps or iterations. During each iteration particles are moved and collisions are computed. The algorithm for a single iteration can be broken down into the following operations:

1. Move particles
2. Perform boundary interactions
3. Sort particles into cells
4. Select collision pairs
5. Calculate collision mechanics
6. Sample macroscopic properties

The first five operations are performed each iteration. The sixth step is performed only when flow properties are desired.

Particle Movement

Particles are moved through space during each iteration according to their individual velocities and the time step. In the absence of external force fields such as gravitational or electromagnetic fields particles follow straight line trajectories. The change in particle position is then given by:

$$\Delta \overline{x} = \overline{v} \Delta t \tag{2.4}$$

This simple vector equation is sufficient for simulations in one, two or three dimensions. Although particles have velocity components in all three directions regardless of the dimensionality of the problem, motion in the homogeneous direction or directions is ignored in two- and one-dimensional simulations.

Flows with axisymmetric symmetry are a special case for which a more complex formulation is necessary to account for the rotational symmetry [4, pages 370–371]. An axisymmetric simulation is considered to take place in a single radial plane of a three-dimensional domain. The presence of an azimuthal velocity component causes particles to move out of this plane when following their physical, three-dimensional trajectories. In order to return particles to the plane of the simulation, final position and velocity components are transformed so as to effectively rotate the particle around the axis of symmetry back into the original plane. This transformation maps a straight line trajectory into a hyperbolic path in the simulation plane.

Boundary Interactions

Particles which cross boundaries of the flow domain are either removed from the simulation or reflected back into the domain. Removal occurs at outflow or inflow interfaces while reflection occurs at surfaces and along lines of symmetry.

Particles striking a surface are affected according to the model used for that surface. Typical models include specular reflection and fully diffuse reflection with thermal accommodation. A particle striking a specular surface reflects back such that the angle of incidence is equal to the angle of departure and with no change in energy. A particle striking a fully diffuse surface is accommodated to the surface temperature. Outgoing velocity components and internal energies are sampled from equilibrium distributions at that temperature. No memory is retained of the incoming properties and the outgoing direction is random. Real surfaces fall somewhere between these two extremes. To simulate this, a surface can be considered partially accommodating, with a fraction of incoming particles reflecting specularly while the rest are diffusely reflected.

Sorting

The operation of the collision and sampling steps requires that the particles be grouped into the cells of a computational grid. This necessitates a sorting step to insure that particles are properly grouped after movement is completed. This operation is typically a computationally expensive operation.

Collision Pair Selection

Collisions between gas particles are the primary means by which the physics of the gas are transmitted through the bulk flow. Calculation of the exact kinematics of a collision — the intersection of particles' trajectories, angles of incidence and impact parameter — would be too costly for a large engineering flow simulation. Instead, collisions are calculated statistically between particles residing in the same computational cell.

Particles are paired in each cell to form a list of possible collision pairs. An accept-reject method is used to determine which pairs, if any, will actually collide. The probability of a collision occurring is dependent on the collision model and selection scheme being employed and is a function of the relative velocity, collision cross section and the gas density in that cell. The probability is also proportional to the time step and the particle weight (see section refsec2:param). The pair is accepted for collision if the probability is greater than a random number.

The number of pairs tested and the collision probabilities are such that the average collision rate is equal to that given by kinetic theory. Several schemes have been proposed for calculating these quantities such as the Time Counter and No Time Counter methods of Bird [4, pages 218–219]. The method used in this study is that of Baganoff and MacDonald [1]. In this scheme the probability of a collision between a particle of species i and a particle of species j is:

$$P = \frac{N_i N_j W_P \, \Delta t \, \sigma g}{SV(1+\delta_{ij})} \left[\frac{2(2-\omega)RT_{ref}}{\mu g^2} \right]^{\omega} \tag{2.5}$$

where N_i and N_j are the number of computational particles of species i and j respectively, S is the total number of collision pairs considered during one time step, V is the cell volume, μ is the reduced mass of the collision pair and g is the relative velocity. The computational parameters W_P (particle weight) and Δt are discussed in section 2.1.4. σ, ω and T_{ref} are parameters of the variable hard sphere collision model.

Collision Mechanics

Once a pair of particles has been selected for collision, momentum and energy are exchanged. The actual trajectories and dynamics of the particles are not considered when determining the mechanics of a collision. Instead, the scattering angle and degree of inelasticity are selected statistically in order to generate post collision properties. Energy and momentum are explicitly conserved in each collision.

Momentum and translational energy transfer is considered in the center of mass frame of reference. The velocity of the center of mass is held constant to conserve momentum while the magnitude of the relative velocity may be affected (increased or decreased) as a result of internal energy transfer (see below). The direction of the post-collision relative velocity vector is selected statistically to represent the physical scattering. Using the VHS collision model the scattering angle is isotropic.

Particles with internal energy may undergo inelastic collisions where energy is exchanged between internal energy modes such as rotational or vibrational energy. Each molecular collision has a chance to exchange internal energy. The probability and amount transferred are selected so that the bulk gas behavior matches macroscopic measurements of relaxation rates. A single collision exchanges energy so as to equilibrate the energy modes. Post-collision energies are sampled from equilibrium distributions according to the Larsen-Borgnakke scheme [5]. The collision probability is the inverse of the collision number, the number of collisions required to equilibrate the energy mode.

Sampling of Macroscopic Properties

The macroscopic properties of a gas are local averages of the properties of the molecules of the gas. Density, velocity and temperature are the zeroth, first and second moments of the velocity distribution function. Since DSMC directly simulates the distribution function, flow field results can be determined by sampling particle properties and calculating the moments of the sampled data.

The computational grid is used to group particles for the purpose of sampling their properties. Flow field properties are calculated in each cell.

Calculation of smooth flow properties requires a large sample size to accurately model the distribution function. The magnitude of statistical fluctuations varies according to the inverse square root of the sample size. A real gas is subject to these fluctuations but the magnitudes are much smaller due to a sample size which is orders of magnitude larger than available in a simulation. The typical number of particles in a cell is too small to provide a reasonable sample. It is generally necessary to sample over a number of iterations to increase the sample size and reduce fluctuations. In a steady flow it is possible to sample over as many iterations as required to obtain smooth results. In time dependent flows it is necessary to perform an ensemble average over many different simulations to obtain an acceptable sample size.

2.1.4 Computational Parameters

Particle Weight (W_P)

Each particle in a DSMC simulation represents a large number of real molecules or atoms. The ratio between the number of simulation particles and real particles is termed the particle weight.

$$W_P = N_R/N_S \tag{2.6}$$

Smaller values of this ratio result in a larger number of computational particles and thus better resolution of flow physics. Increasing the particle weight decreases the number of particles and thus decreases the cost of the simulation. The value of the particle weight is selected so as to obtain a balance between resolution and efficiency. A typical value for a two-dimensional simulation might be 10^{10}.

Time Step

A finite time step is used to decouple the movement of particles from collisions. In order to do this, the time step must be a small fraction of the mean time between collisions. The exact expression is dependent on the collision model employed but is in general a function of density and temperature. For a single species gas using the VHS model the collision time is:

$$t_{col} = \left[\Gamma(2-\omega)\left((2-\omega)\frac{T_{ref}}{T}\right)^{\omega}\pi {d_{ref}}^2 n\sqrt{\frac{16kT}{\pi m}}\right]^{-1} \tag{2.7}$$

The expression for a multi-species flow is more complicated but is proportional to density and temperature to the same exponent.

For purposes of statistical accuracy it is also desirable that a particle not remain in a single cell for a large number of time steps. A high residence time tends to bias the sampling of macroscopic properties. The time step should then be set at a large enough value that the majority of particles moving at the bulk fluid velocity will cross a given cell in a small number of time steps (e.g. 4 or 5 steps). Residence time and mean collision time provide bounds for the selection of a simulation time step.

Grid Cell Size

A computational grid is used in DSMC to group particles for the purposes of calculating collisions and sampling macroscopic properties. The size of the computational cells is limited by the constraints of physical accuracy and computational efficiency.

As previously mentioned, collisions are calculated statistically between pairs of particles residing in the same cell. The two particles participating in a collision may be separated in space by as much as one cell dimension. In a real gas the average distance travelled by a particle between collisions is termed the mean free path and is given by:

$$\lambda = \frac{\overline{C}}{\Theta} \tag{2.8}$$

where $\overline{C}$ is the mean thermal speed of the gas and Θ is the collision frequency ($\Theta = 1/t_{col}$). In a real gas two particles will not collide if they are separated by more than one mean free path. This provides a limit on the size of the computational cell. Ideally, cell dimensions should be smaller than the local mean free path (e.g. $\Delta x \leq 0.3\lambda$) at all points.

The presence of cells which are larger than one mean free path results in reduced resolution of gradients at the macroscopic level. Since the collision process tends to force particles toward equilibrium, calculating a collision between widely separated particles will tend to homogenize the flow at these points. This diminishes physical gradients in the flow which may occur over distances on the order of a mean free path.

Computational efficiency provides a lower bound on the size of cells. In order to accurately resolve collisional effects it is necessary to have a minimum number of particles in each cell. The smaller the cells that are used the larger the number required to fill the domain and consequently the larger the number of particles needed for the simulation. It is often necessary to use cells that are on the order of one or more mean free paths in size for reasons of computational cost.

2.2 DSMC Implementation - MONACO

The work presented in this dissertation was performed using several versions of the DSMC software MONACO developed at Cornell University [13]. MONACO is a general purpose DSMC code designed for workstation architectures. It is designed to be applicable to a wide range of flow problems without requiring problem specific modifications. It can be run on a single machine or in parallel on an arbitrary number of processors. The original version of the code handles two-dimensional and axisymmetric flows.

The design of the software follows an object oriented paradigm with the computational cell as the basic data structure and the particle as a secondary structure. The majority of simulation data are stored local to each cell. In particular, particles are grouped into lists associated with the cell in which they are located. This is different from traditional implementations of DSMC where the majority of data is stored in global arrays.

A cell based, object oriented data structure is very important for high performance on workstation architectures. Processors in these machines have relatively slow main memory access but fast cache access. Large global data structures generally do not use the cache memory efficiently because the cache must be reloaded frequently. Localized data allow the cache memory to be heavily used before being reloaded, resulting in better performance.

The cell data structure provides a great deal of flexibility in terms of the computational grid. It also allows the DSMC algorithm to be very efficiently parallelized using domain decomposition.

MONACO has been designed to be modular. The code is divided into three separate libraries each of which handles different types of tasks associated with the DSMC algorithm. One handles memory management, parallel message passing and coordinates the overall algorithm. Another handles all tasks related to the geometric nature of a simulation such as grid definitions and particle movement. The third performs calculations of all physical models, such as collision mechanics and the physics of boundary interactions.

2.2.1 Computational Grid

The computational grid is used in the DSMC algorithm to group particles for the calculation of macroscopic properties and determination of collision pairs. These are fundamentally local tasks; knowledge of the overall grid structure is not necessary to perform them. Each computational cell can then be considered a separate entity. The grid is stored as a list of distinct cells. Individual cells store the identity of neighboring cells and boundary conditions so that the global grid structure is only known implicitly.

Grid generation is treated as a preprocessing step reflecting the logical separation between grids and the DSMC algorithm. This allows the use of engineering tools designed specifically for grid generation. Grids are specified in a format based on a standard developed by the National Grid Project (NGP) [28]. This format includes boundary conditions as part of the grid definition eliminating the need for any problem specific boundary handling.

The cell based nature allows for great flexibility in the computational mesh. The grid is naturally unstructured so no overall structure is imposed on the grid by the code. As a result, geometric configurations of arbitrary configuration can be handled. Although the data structure is fundamentally unstructured, there is no requirement that cells be triangular as in typical unstructured grids. Traditional structured grids can be easily represented in the NGP format using quadrilateral cells. Cells may, in general, have any number of sides, though typically triangular or quadrilateral cells are employed. The only restriction is that all cells be convex. Hybrid grids consisting of several types of cells can be used.

2.2.2 Particle Movement and Sorting

Particles are moved during each time step along straight line trajectories according to their velocities. The routines which calculate the movement of particles are optimized to take advantage of statement overlapping on modern processors. This is particularly important for axisymmetric flows where particles trajectories are mapped onto hyperbolic paths on the symmetry plane.

A particle must always be stored with its current cell to maintain data locality. Completing each particles movement and then employing an explicit sorting step makes this difficult. Instead a particle tracking scheme is used. As the particle moves, intersections are calculated between its path and the sides of its current cell. When the particle is determined to have crossed one of these sides and left the cell it is passed to the neighboring cell to continue its movement. Movement is thus performed in a series of cell sized steps.

Tracing the actual trajectory is computationally expensive compared to directly calculating the final location, particularly for axisymmetric flows. The added cost is offset by the fact that no sorting step is necessary. At the completion of movement each particle is automatically stored with its final cell. Particle tracing also provides a more accurate method of calculating intersections with boundaries and their effects.

2.2.3 Variable Scaling

MONACO allows the value of the time step and the particle weight to be varied throughout the flow domain. Variation of these computational parameters can significantly improve the performance of a simulation. Using a locally appropriate value everywhere rather than a single global value eliminates wasted computational effort and improves the efficiency of the calculation.

Variation of these parameters is handled by assigning scale factors which modify the reference value for the simulation to each cell. Thus the time step used in a given cell is the product of the reference time step and the time scale factor for the simulation. This implementation allows parameter values for the entire simulation to be modified while holding constant the ratio between individual cells. The generation of scaling factors is treated as a preprocessing step so that the selection scheme can be tailored to each individual problem.

Time Scaling

A finite time step is used to decouple particle movement from collisions. Criteria for setting the time step have been discussed in a previous section. When considering a problem which includes large variations of density and cell size, these constraints may yield an optimal time step which varies considerably across the computational domain. For example, in the calculation of a supersonic nozzle expansion, the time step which is needed near the throat is orders of magnitude smaller than that appropriate at the exit plane. If a single time step is used, the calculation is limited by the minimum value. This results in many times more time steps being calculated at the exit plane than are necessary for resolution of the physics and therefore a significant waste of computational effort.

The problem of widely varying physical time scales can be addressed in the simulation of a steady flow by using time steps appropriate to the local flow conditions. Each computational cell can have a distinct time step. The local time scale affects all aspects of the DSMC algorithm. A particle moves a distance proportional to the local time step and collision probabilities are determined using the local time step. When a particle crosses from one cell into another it is immediately affected by the new time scale. The ratio between the local time step and the reference time step for the simulation is termed the time scale factor for that cell.

When different time steps are used, a single iteration of the DSMC algorithm no longer represents the same amount of physical time in each cell. This disparity in elapsed time is accounted for by effectively weighting all particles by the time scale factor. Particles in cells which have a larger time scale factor then effectively represent a larger number of real gas particles. This affects both macroscopic sampling of number density and collision probabilities which depend on the local number density.

Modifying the effective particle weight to account for time scaling eliminates the need for any form of explicit synchronizing of the cells. Although the time step may be different, each cell is processed normally during each iteration of the DSMC algorithm. The lack of explicit synchronization does, however, prevent the use of time scaling for simulations of unsteady flows.

The rationale for weighting according to the time scale factor can be explained as follows. For simplicity, the argument will be presented in one dimension though it extends to three-dimensional flows. Consider a one dimensional computational cell of length L, time step $\Delta t = \Delta t_{ref}$ and reference particle weight is W_P. A uniform flux of particles crosses the cell with an average of F particles crossing into one side of the cell and F leaving at the other side during each time step. If the average velocity is V and the time step is the reference value then $T = L/(V\Delta t_{ref})$ time steps are required for the average particle to cross the cell. The number of particles in the cell during a single time step is $N = F \times T$. If the number density of the cell is sampled over one time step the following result is obtained:

$$\begin{aligned} n &= \frac{W_P}{L} N \\ &= \frac{W_P F}{V\Delta t_{ref}} \end{aligned} \tag{2.9}$$

Now consider the same cell when a time scale factor of s is applied. The local time step is then $\Delta t' = s \times \Delta t_{ref}$ and the number of time steps required to cross the cell is $T' = L/(V s \Delta t_{ref})$. The number of particles in the cell is then $N' = F \times T'$. Sampling the number density of the cell gives:

$$\begin{aligned} n' &= \frac{W_P}{L} N' \\ &= \frac{W_P F}{sV\Delta t_{ref}} \end{aligned} \tag{2.10}$$

The number density is different by a factor of $1/s$. Changing the time step cannot have an effect on the physical number density. Since F, V and Δt_{ref} are constants the only way to remedy this contradiction is to introduce an effective particle weight which is scaled by the time scale factor s. Thus

$$W'_P = s \times W_P \tag{2.11}$$

and

$$\begin{aligned} n' &= \frac{W'_P}{L} N' \\ n' &= \frac{sW_P F}{sV\Delta t_{ref}} \\ n' &= n \end{aligned} \tag{2.12}$$

One result of time scale factors is to change the number of particles per cell. As in the previous example, a scale factor of s modifies the particle count by the factor $1/s$ (e.g. doubling the time step cuts the count in half).

Particle Weight Scaling

The number of real gas particles represented by a single simulation particle is the particle weight, W_P. It is often desirable for purposes of efficiency to vary this weight throughout the flow field. The best known example of this is the case of radial weight scales in axisymmetric flows. In this case, the particle weight is increased in cells away from the axis to counteract the increasing volume of these cells. More generally, variation in weight scales can be used to improve resolution in portions of the flow field where the number of particles is low or to increase efficiency where there is an excessive number of particles.

The interface between cells with different weights represents a problem of continuity. When a particle crosses between such cells it suddenly represents a different number of real particles creating a discontinuity in mass flux across the interface. This discontinuity can be accounted for by allowing a probability that the particle is either cloned when moving into a cell with a lower particle weight or destroyed for a higher weight.

$$W_P{}^{old} > W_P{}^{new} : P_{clone} \quad = W_P{}^{old}/W_P{}^{new} - 1 \tag{2.13}$$

$$W_P{}^{old} < W_P{}^{new} : P_{destroy} \quad = W_P{}^{old}/W_P{}^{new} \tag{2.14}$$

If the probability of cloning is greater than one more than one new particle can be generated. A cloned particle is identical to the original particle.

The clone-destroy operation conserves mass flux across the interface when many crossings are averaged. It does not, however, explicitly conserve flux except for the case of cloning with an integral ratio of weights. This non-conservative nature introduces an additional random walk to the simulation which can lead to larger statistical fluctuations. Additionally, the identical particles created by cloning do not provide more resolution than the original higher weight particle until their properties are changed by collision or boundary interaction.

2.2.4 Parallel Implementation

The DSMC method is computationally intensive. Improved performance can be obtained through parallel execution. The MONACO software is designed to run efficiently on large parallel machines which are based on workstation architectures, such as the IBM SP2, the Cray T3D or the Intel Paragon.

The DSMC algorithm is readily parallelized through domain decomposition. The cells of the computational grid are distributed among the processors. Each processor executes the DSMC algorithm in serial for all particles and cells in its domain. This decomposition follows naturally from the data structure since particle data is localized with each cell. Parallel communication occurs when particles cross domain boundaries and must be transferred between processors.

High parallel efficiency can only be achieved if communication is minimized and the computational load is evenly distributed between processors. To minimize communication, the boundaries between parallel domains should lie along the streamlines of the flow field. Since the majority of particles will have a velocity which is close to the bulk velocity, this ensures that only a small fraction will cross domain boundaries. Proper load-balancing can, to first order, be achieved by using an equal number of particles on each processor. The number of collisions occurring per processor has a second order effect on computational load; processors which compute a low number of collisions can handle a larger number of particles. The number of cells assigned to each processor does not have a significant effect on efficiency.

Obtaining an efficient decomposition for a new flow problem is not a trivial task. Usual practice is to first run a coarse simulation with a low number of particles. This is done either in serial or using a crude decomposition. The particle distribution resulting from this simulation is then used as a basis for developing an accurate domain decomposition. In this fashion, parallel efficiencies of over 90% often are achieved.

The parallel efficiency of a simulation is calculated during each time step using run time performance data. The parallel efficiency of processor i is defined to be:

$$\eta_i \equiv \frac{t_i^s}{t_i^s + t_i^c} \tag{2.15}$$

t_i^s is the time spent executing the DSMC algorithm on particles located on that processor and is effectively the serial execution time for that portion of the domain. t_i^c is the time used for communication. The sum of t_i^s and t_i^c is the total time required for that time step and as a result of synchronization is equal for all processors.

$$t_{step} = t_i^s + t_i^c \tag{2.16}$$

The definition of processor efficiency can be used to calculate the overall parallel efficiency of a simulation per time step. Parallel efficiency is defined as:

$$\eta \equiv \frac{t_{serial}}{t_{parallel}} \frac{1}{p} \tag{2.17}$$

where t_{serial} is the time required to run the time step using only a single processor and $t_{parallel}$ is the execution time in parallel on p processors. The single processor time is the sum of serial time spent by each processor:

$$t_{serial} = \sum_{i=1}^{p} t_i^s \tag{2.18}$$

Total parallel time is the sum of serial and communication time and is equal to t_{step}:

$$t_{parallel} = t_i^s + t_i^c \tag{2.19}$$

Combining the expressions for serial and parallel time with the definition of processor efficiency gives the following expression for overall efficiency:

$$\eta_t = \frac{1}{p} \sum_{i=1}^{p} \eta_i \tag{2.20}$$

2.2.5 Physical Models

MONACO employs a variety of models for simulating physical gas effects. Collision pairs and probabilities are determined using the Baganoff-McDonald scheme [1]. Collision cross-sections are calculated using the Variable Hard Sphere (VHS) model [3]. Rotational energy exchange probabilities are calculated using an energy dependent model by Boyd [6]. Vibrational energy transfer is computed with a constant probability. Most of the cases investigated in this work have not considered vibrational energy due to freezing of the energy mode. Boundary interactions can be computed with specular or diffuse reflection and either full or partial accommodation.

Chapter 3

Particle Resolution Issues

The DSMC technique uses particles to represent a macroscopic gas. The accuracy of a simulation is directly related to the number of particles per cell. As the number of particles employed increases the resolution of the problem improves both in terms of physical processes and macroscopic properties. The limiting factor is computational cost which is roughly proportional to the total number of particles. When dealing with large flow problems which require many grid cells it is generally necessary to use a small number of particles per cell in order to have a reasonable total for the simulation.

A minimum number of particles is needed to resolve flow physics through collisions. Collisions between particles are discrete events which occur with probabilities selected so that bulk collision rates are maintained when averaging over many events. If the overall collision rate is low the probability of individual collisions will also be low. If the number of collision pairs in a cell is small it is possible to lose low probability events. Typically twenty particles per cell is considered an acceptable number to resolve fluid mechanics. More particles may be required to accurately model certain finite rate processes or species which have small mole fractions.

A numerically efficient calculation is one in which the desired physics are resolved using a minimum number of particles. It is therefore important to consider how the particle count is affected by various computational parameters. The distribution of particles and the computational cost of a simulation can be manipulated through these parameters.

3.1 Scaling with Density

The number of particles per cell is related to the number density by the following relation:

$$N_P = \frac{n\mathcal{V}}{W_P} \tag{3.1}$$

If the density is assumed to be fixed the particle count can be increased by increasing the volume of cells or reducing the (effective) particle weight.

Accurate spatial resolution requires that the computational cells be sized according to the local mean free path. Cell volume is then related to the density by the fact that the mean free path is inversely proportional to the number density. If each dimension of a cell is Δx and this is varied proportionally with the mean free path then this cell size will be inversely proportional to the density:

$$\Delta x \propto \lambda \propto n^{-1} \tag{3.2}$$

The volume of a cell is a function of the cell size. The form of this relation depends on the dimensionality of the problem being simulated.

3.1.1 Two-Dimensional Flows

In a two-dimensional calculation, the cells are planar polygons and the third dimension of the cell is taken to be unity. The cell volume is given by:

$$\begin{aligned}\mathcal{V} &\propto \Delta x \times \Delta x \times 1 \\ &\propto \lambda^2 \\ &\propto n^{-2}\end{aligned} \tag{3.3}$$

Inserting this expression for cell volume into that for number of particles (eq. 3.1) gives the following relation between number of computational particles and flow density (assuming constant particle weight):

$$N_P \propto n^{-1} \tag{3.4}$$

The resolution then decreases linearly as the density increases.

3.1.2 Three-Dimensional Flows

In a three-dimensional calculation, the cells are polyhedrons. The cell volume is thus given by:

$$\begin{aligned}\mathcal{V} &\propto \Delta x \times \Delta x \times \Delta x \\ &\propto \lambda^3 \\ &\propto n^{-3}\end{aligned} \tag{3.5}$$

Using this expression with eq. 3.1 gives the following relation between number of computational particles and flow density (assuming constant particle weight):

$$N_P \propto n^{-2} \tag{3.6}$$

The resolution then decreases quadratically as the density increases.

3.1.3 Axisymmetric Flows

In axisymmetric calculations, the cells are described by polygons in the simulation plane. The cell volume is that of the annular region formed by rotating the bounding polygon through 360 degrees:

$$\mathcal{V} = 2\pi R_C A_P \tag{3.7}$$

where R_C is the radial location of the centroid of the cell and A_P is the planar area.

The planar area of the cell scales as the square of the local cell size. For the majority of cells the radial location of the centroid is independent of cell size so the cell volume has the same scaling as in the two-dimensional case (eq. 3.3). Consequently, the particle count scales as the inverse of the density (eq. 3.4).

Cells in the vicinity of the axis of symmetry have a different scaling. The centroid of a cell adjacent to the axis is located approximately one half of a cell dimension away from the axis. The position of the centroid is then proportional to the cell size in near axis cells. This gives the following scaling for this class of cells

$$\begin{aligned}\mathcal{V} &= 2\pi R_C A_P \\ &\propto \Delta x \times (\Delta x \times \Delta x) \\ &\propto \lambda^3 \\ &\propto n^{-3}\end{aligned} \tag{3.8}$$

This leads to the number of particles scaling with the inverse of the square of the density as in three-dimensional flows (eq. 3.6).

$$N_P \propto n^{-2} \tag{3.9}$$

3.1.4 Resolution Difficulties

The results for two- and three-dimensional flows both show the interesting result that the number of computational particles in a cell varies inversely with the number density at that point. This counter-intuitive result indicates that the resolution of the calculation is highest in the parts of the domain that have the lowest density.

In a typical problem the greatest amount of physics (collisions and chemistry) occur in high density regions. In order to correctly model the flow it is necessary to resolve flow physics occurring in these regions. The particle weight for the calculation must be set to a low number so that there will be a reasonable number of particles in these high density regions. As a result the more rarefied portions of the flow field will have a large number of particles and will be over resolved. More computational time is spent calculating the low density regions than is needed. This results in less efficient calculations.

3.2 Improving Particle Distribution

The optimal distribution of particles from a computational standpoint is one where each cell has approximately the same number of particles. This would ensure that flow physics are properly resolved throughout the flow domain and there is no wasted computational effort due to over-resolution. Due to the scaling of particle count with density, only a flow with uniform number density throughout would naturally have this type of ideal distribution.

There are several ways in which the particle count distribution can be adjusted to approach the optimal case. Variable time step and particle weight scaling each affect the particle count. Cell sizes and shapes can be selected so as to improve the distribution. In each case an estimate of the density field is used to set the parameter of interest. This initial estimate can be based on knowledge of the flow field or a preliminary simulation performed with a reduced number of particles.

3.2.1 Varying Particle Weights

The most direct method of adjusting the particle count in each cell is to vary the particle weight across the computational domain. Then in each cell, simulation particles represent different numbers of physical atoms or molecules. If particle weights are assigned such that they have the correct density dependence, the variation of particle count with density can be eliminated.

Varying the particle weight is the most direct method of improving resolution. Equations 3.4 and 3.6 show that the main difficulty in terms of particle count is the decrease in number of particles with increasing density. Applying particle weights with the following density dependences would counter this effect:

$$\begin{aligned} W_P &\propto n^{-1} && \text{2D} \\ &\propto n^{-2} && \text{3D} \end{aligned} \tag{3.10}$$

Applying an inverse density based particle weight scaling can have significant impact on the statistical accuracy of a simulation. For most flows this type of scheme will result in large variations in weight. This leads to a large amount of cloning and destroying of particles and a significant increase in statistical fluctuations.

This type of scaling is a particular concern for highly compressive flows such as those generated by a hypersonic blunt body. In order to counteract the density rise caused by compression in front of the body, the particle weight must drop rapidly. This produces a significant amount of cloning in the vicinity of the body. The presence of many identical particles in a region which is likely to have an important physical effect on the flow field can seriously degrade the accuracy of the calculation. At best there will be less resolution of this region than expected. In order to avoid this problem, it is suggested that the particle weight generally not decrease in the direction of the main flow.

Under certain conditions, particle weights can be an effective means of improving particle resolution. These flows are typically relatively high density expanding flows. A high density insures collisional variation of cloned particles while expanding flows avoid the problem of significant cloning. This method should be used with caution and only when necessary. It can rarely be used to offset the entire effect of density variations on particle count without causing significant statistical problems.

3.2.2 Varying Time Steps

In flows where there is a large variation in density, there is a commensurate variation in the mean collision time. As has been discussed in a previous section, the use of a single time step for the entire flow domain will result in a waste of computational effort. Many small time steps are calculated in low density regions where the mean collision time is high when a single larger step would be sufficient. The use of variable time scales can eliminate this inefficiency and in the process produce a more uniform particle count throughout the flow field.

The primary restriction on the time step is that it be a small fraction of the mean collision time. A reasonable choice for time steps is a constant fraction K of the local collision time:

$$\Delta t = K t_{col} \tag{3.11}$$

Since the collision time is inversely proportional to the density, the time scale s is given by:

$$s = \frac{\Delta t}{\Delta t_{ref}} \propto n^{-1} \tag{3.12}$$

The variation in time step causes the effective particle weight to be a function of the flow density:

$$\begin{aligned} W_P &= W_{P,ref} \times s \\ &\propto n^{-1} \end{aligned} \tag{3.13}$$

Inserting this expression into that for the number of particles per cell (eq 3.1) gives the following result:

$$\begin{aligned} N_P &= \frac{n\mathcal{V}}{W_P} \\ &\propto n^2\mathcal{V} \end{aligned} \tag{3.14}$$

The volume of cells was found to vary inversely with density. The effect of varying the simulation time step according to this rule is to reduce the dependence of particle count on density. Applying the earlier results for cell volume (equations 3.3 and 3.5) gives the following expressions for particle count:

$$\begin{aligned} W_P &\propto 1 && \text{2D} \\ &\propto n^{-1} && \text{3D} \end{aligned} \tag{3.15}$$

The dependence on density is eliminated for two-dimensional flows and reduced for three-dimensional problems. The behavior of axisymmetric flows is again the same as in two dimensions away from the flow axis and three dimensions near to the axis.

The use of variable time steps thus can reduce or eliminate particle resolution difficulties as well as improving the convergence and efficiency of the calculation. The correct variation of time steps can easily be achieved if the cells in the computational domain are sized proportionally to the mean free path. The cell size will be proportional to the inverse of the density and thus be proportional to the mean collision time. The cell size can then be used to scale the time steps directly. Cell size can be measured by the minimum altitude of the cell. If cell stretching is employed (see section 3.2.3) the cell dimension which is directly scaled to the mean free path should be used to determine the time step.

3.2.3 Grid Manipulation

From the point of view of the physical correctness of the DSMC method, the ideal grid consists of a set of cells whose dimensions are some fraction of the local mean free path. The scale length of flow gradients will be on the order of the mean free path and therefore this size requirement insures that gradients are properly resolved.

In many flows it is clear from the geometry that the primary gradients will be in a specific direction. In these circumstances it is reasonable to relax restrictions on cell dimensions in the directions other than that of the primary gradients. Several mean free paths can be used rather than one or smaller. Significant gains in particle count resolution can be obtained at the cost of some spatial resolution.

Consider a two-dimensional flow with no weight or time scaling which is compressed at a wall. Since the particle count varies as the inverse of the density (eq 3.4), the compression will cause a decrease in particle count at the wall. If cells in the vicinity of the wall are stretched in the direction parallel to the wall by a factor of 10, this increase in cell volume will counteract the effect of the density rising by the same factor. If the flow is such that the properties vary slowly along the wall (e.g. bulk flow normal to the surface) the reduction in spatial resolution will be inconsequential.

In three dimensions, the cells can often be stretched in two directions. Increasing each dimension by an order of magnitude leads to a hundredfold increase in volume, compensating for a two order of magnitude increase in density.

This type of grid manipulation can be particularly important in axisymmetric problems. Cells near the axis tend to suffer resolution difficulties due to three-dimensional type scaling. Stretching cells in the radial direction has a squared effect on particle count due to the resultant movement of the cell centroid away from the axis.

Cells can be of any geometric shape as long as they satisfy the size requirement. From a practical point of view, cells are typically quadrilateral or triangular in two-dimensional and axisymmetric flows and hexahedral or tetrahedral in three dimensions (structured and unstructured grids respectively). While unstructured grids provide the most flexibility for meshing complex domains, structured cells are more easily stretched to increase particle count resolution. In general, a hybrid grid which combines the advantages of both makes the most effective grid.

3.2.4 General Strategies

A combination of grid manipulation and variable time scales will often be the best means of controlling the distribution of particles. Time scaling can eliminate one level of density dependence while not impacting the accuracy of the simulation. Creative use of grid stretching can compensate for a large portion of the remaining dependence. Some spatial resolution is sacrificed but this can be managed so as to not adversely affect the overall macroscopic results.

In certain cases it is not desirable to strictly scale the cells by the mean free path in any case. An example of this is the outer portion of a plume expanding into vacuum. The extremely low density associated with this portion of the flow will have a very large mean free path. Reasonable spatial resolution would suggest using cells many times smaller.

An example at the other extreme is the continuum throat used as an inflow condition for a nozzle expansion. If the flow properties are varying smoothly in this region and the flow has enough collisions to be in local equilibrium resolving the flow to the level of one mean free path may be unnecessary and certainly very expensive.

It is not necessary for an efficient calculation to have a completely uniform distribution. In some cases excessive efforts to obtain this may affect the macroscopic results of the calculation either by reduced spatial or statistical resolution. However, attention to this issue is extremely important when large engineering simulations are performed in order to maintain a reasonable cost.

3.3 Demonstration Results

Particle count scaling and strategies for improving the distribution are demonstrated by considering a representative plume impingement flow. The plume from a nozzle impacts on a surface placed downstream. Figure 3.1 shows a schematic of the configuration. The initial expansion of the plume followed by compression at the surface results in a range of densities which is suitable for examining resolution and scaling.

The plume flow is based on an impingement problem which will be examined in detail in the following chapter. The gas is nitrogen and is assumed to be vibrationally and chemically frozen. Simulation of the plume begins at the exit of the nozzle and extends above and beyond the surface. Conditions at the nozzle exit are taken from the simulation of the heated resistojet nozzle flow (see section 4.2). At the centerline of the nozzle exit plane the flow conditions are a velocity of 1100 m/s, number density of $4.8 \times 10^{21} \mathrm{m}^{-3}$ and temperature of 98 K. The same inflow conditions will be used for both axisymmetric and two-dimensional simulations.

3.3.1 Two-dimensional case

The simplest, interesting case to consider is two-dimensional flow. The problem then involves the plume from a plane nozzle impinging on an infinitely wide plate. Although this configuration is not representative of a real problem, a two-dimensional simulation most clearly shows the effects of density scaling and demonstrates the appropriate strategies.

As a base case the flow is calculated using a fully unstructured (triangulated) grid. Cell sizes are twice the local mean free path as determined by a coarse, preliminary calculation. Two mean free paths are used in order to reduce the computational cost. Figure 3.2 shows the grid for this calculation. Compression of the gas at the surface results in a high density of grid cells while expansion away from the body causes the cells to rapidly grow in size.

Data at the symmetry plane will be examined to consider the effects of particle count scaling. The symmetry plane is also a stagnation plane for the flow so that the maximum compression at the surface occurs here. The wide range of densities occurring on this plane (or line in the simulation space) makes this the most interesting portion of the domain to consider for this study.

Figure 3.3 shows results taken along the stagnation line. Number density and particle count are plotted against distance from the nozzle exit. As expected, the number of particles per cell increases through the expansion region and decreases as the flow compresses in front of the impingement surface. Equation 3.4 indicates that the particle count will be inversely proportional to the density. This relationship is demonstrated in figure 3.4 which plots the product of density and particle count along the stagnation line, normalized by the values at the exit plane. The product is constant on average with fluctuations of 20%. Thus the particle count is inversely proportional to the density to within this variation.

The deviation from inverse proportionality is a result of the variation of cell sizes occurring in the grid. Figure 3.5 plots cell volumes and the square of the local mean free path along the stagnation line. Both profiles are normalized using values at the inflow plane. In an ideal grid, cell dimensions would be exactly proportional to the mean free path leading to volumes which are proportional to the square of the mean free path in a two-dimensional flow. In a

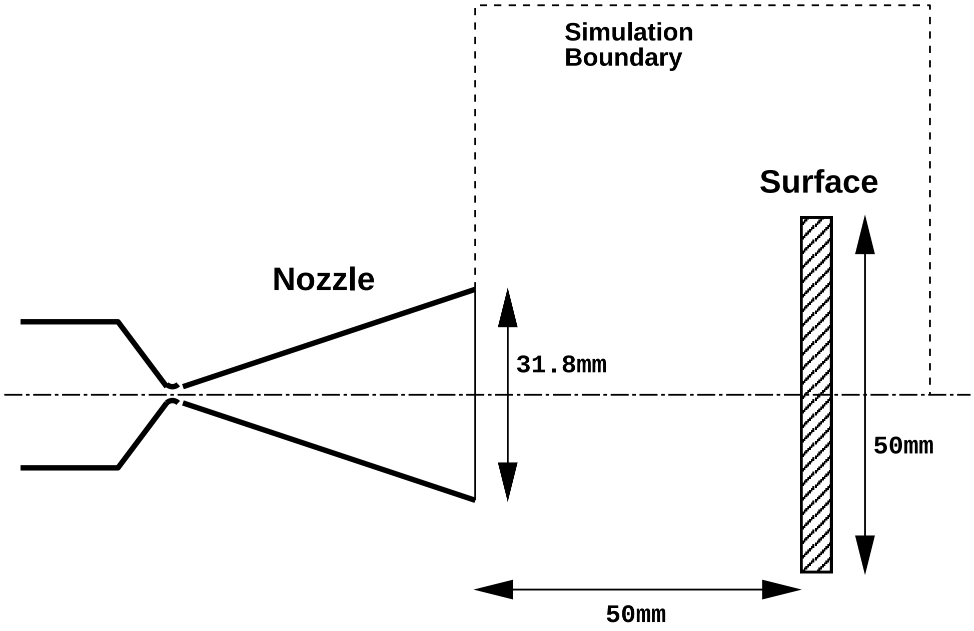

Figure 3.1. Schematic of demonstration impingement case.

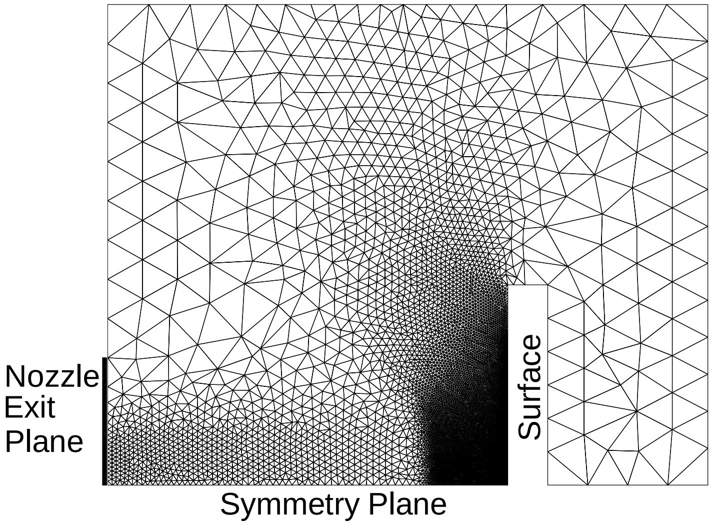

Figure 3.2. Two-dimensional unstructured grid for demonstration impingement case.

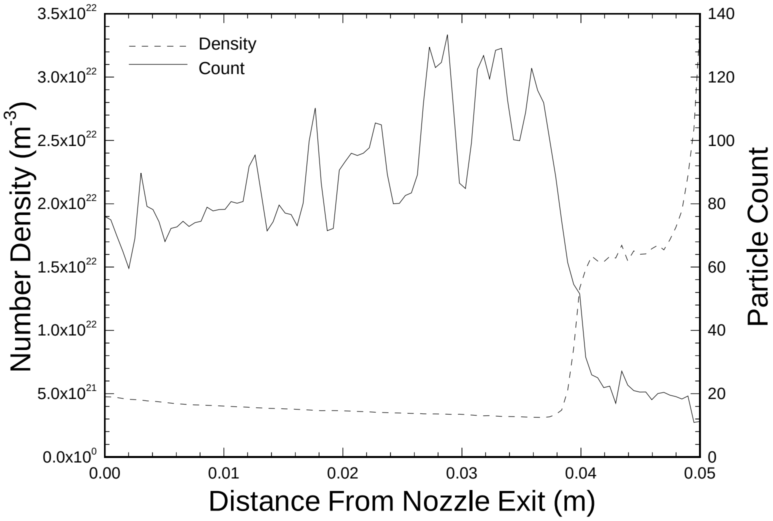

Figure 3.3. Density and particle count along stagnation line for base two-dimensional case.

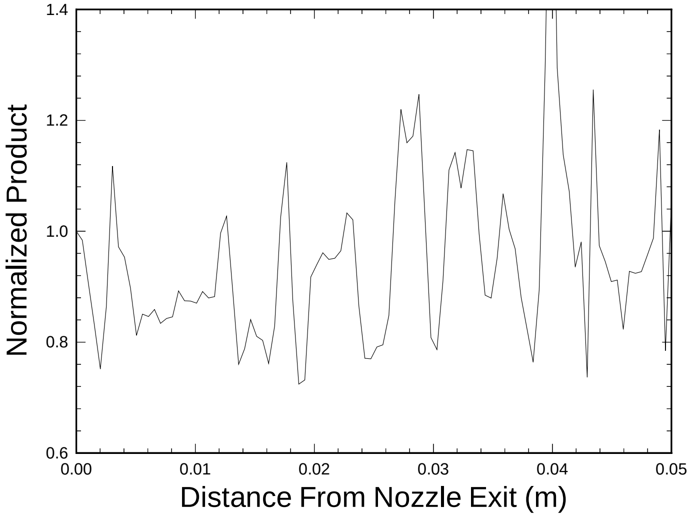

Figure 3.4. Normalized product of density and particle count along stagnation line for base two-dimensional case.

real grid some variation in cell dimensions is necessary to completely fill the domain. This variation in cell size and volume causes the particle count to deviate from the inverse linear relationship specified by equation 3.4.

The calculation uses a total of 750,000 particles. If twenty particles per cell is assumed to be necessary to accurately resolve the flow then the simulation size cannot be made smaller for this problem without improving the particle distribution. Cells immediately at the surface have somewhat less than twenty particles so that a smaller particle weight and larger total number of particles would be needed to satisfy the twenty particle limit throughout.

Embedded Surface Grid

Compression of the gas at the surface causes the particle count to drop rapidly in the vicinity of the surface. The use of an embedded quadrilateral grid with cell stretching parallel to the surface can improve resolution. Figure 3.6 shows a portion of the resulting hybrid grid when such an embedded grid is included. At the surface, cells are ten mean free paths in height, perpendicular to the surface gradients, and two in width. At the outer extent of the embedded grid the aspect ratio of the cells is much closer to unity; this is necessary to have a smooth transition to the unstructured portion of the grid. The embedded grid extends to the edge of the compression region in front of the body (approximately 10 mm). It extends over only 80% of the height of the body. Above this point gradients in the vertical direction start to become important making grid stretching less acceptable. Additionally, the density has fallen off sufficiently at this point so that the unstructured grid can provide adequate resolution.

The effect of the embedded grid is to improve particle resolution at the surface without sacrificing important spatial resolution. Figure 3.7 shows the variation of density and particle count the stagnation line. The number of particles per cell in the compression region is increased by a factor of five, the same factor used to stretch the surface quadrilateral cells. The density profile is unaffected by the change in grid structure. Stretching of the surface grid makes it possible to fully resolve surface conditions using fewer overall particles.

The use of an embedded grid has no effect on the total number of particles in the simulation. Since the same particle weight was employed as in the base two-dimensional case the size of the simulation remains the same (750,000 particles). However, figure 3.7 indicates that the total number of particles could be reduced by a factor of two (by increasing the particle weight) and the nominal twenty particle limit would be satisfied everywhere on the stagnation plane. Since the stagnation plane has the highest densities and is thus the most difficult portion of the flow to resolve this conclusion can be considered to hold throughout the flow field. Thus, using stretched quadrilateral cells at the surface would allow the problem to be accurately calculated using only 375,000 particles and a 50% savings in CPU time.

Variable Time Steps

The scaling of particle count with density can be effectively overcome in two-dimensional flows by utilizing a time step in each cell which is inversely proportional to the density (see section 3.2.2). The fully unstructured grid is employed along with time step variation

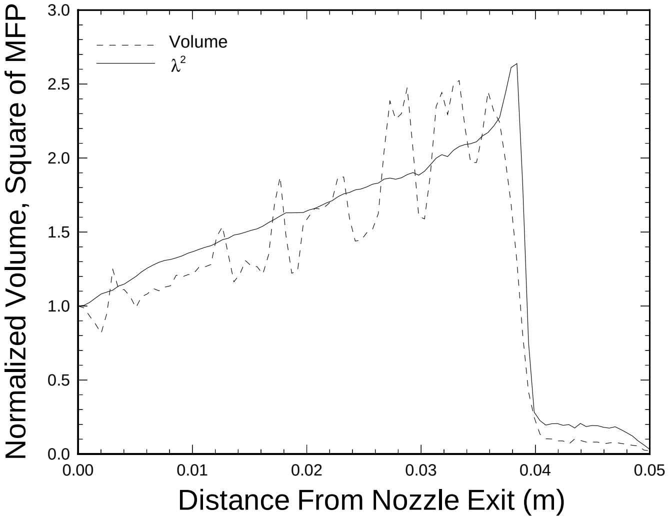

Figure 3.5. Normalized cell volume and square of mean free path along stagnation line for base two-dimensional case.

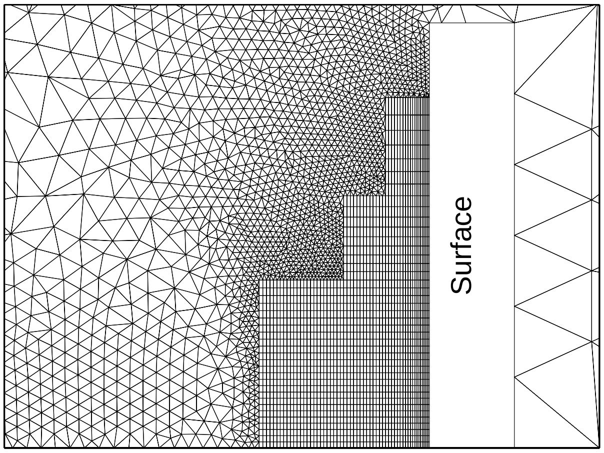

Figure 3.6. Hybrid grid with embedded quadrilateral grid at surface.

according to this rule. The size of cells, measured by the altitude of the triangle, is used to estimate the local mean free path and thus the density.

The use of time scaling according to an inverse density rule should eliminate the variation of particle count with density. Figure 3.8 shows the resulting densities and particle counts along the stagnation line. The particle count shows no correlation with density and is constant on average. Fluctuations can again be attributed to deviation in cell size from the local mean free path. The effect is compounded by the use of cell size in the determination of time scale. A more accurate scheme would directly utilize the density field which was used for grid generation.

The results of this calculation indicate that using variable time steps is the most effective way of obtaining a uniform particle count distribution. The distribution is approximately independent of density throughout the flow field. This provides the most efficient use of computational power.

The use of variable time steps directly affects the number of particles in the simulation through the effective particle weight. As a result the total number of particles used is only 540,000 with the same overall particle weight. Further, figure 3.8 indicates that the total number could be reduced by 40% and still satisfy the requirement of twenty particles per cell. Additionally, the flow converges to a steady state in many fewer iterations using variable time steps. The length of the transient is reduced by a factor of four in this case. The total CPU time savings which can be realized for this problem using variable time steps is on the order of 80%.

3.3.2 Axisymmetric Case

The same impingement flow is used to demonstrate the effects of density scaling in axisymmetric flows. The configuration is now a conical nozzle generating a plume which impacts on a disk. Qualitatively the flow field is the same as in two dimensions. However, the shock is closer to the body and the resulting compression region is smaller in the axisymmetric case.

As a base case the flow is calculated using an unstructured grid using variable time steps. The grid used differs from that used in two dimensions because the density field is different. Time steps are again scaled using cell size. In an axisymmetric flow, the particle count should scale like a two-dimensional flow away from the axis and like a three-dimensional flow close to the axis. Using variable time steps the particle count should then be approximately constant off axis and inversely proportional to density at the axis.

Figure 3.9 shows density and particle count taken at a radius equal to half the nozzle radius. This particle count is approximately constant from nozzle to surface which is expected at this radius. Fluctuations are again a result of variations in cell size. Figure 3.10 shows a similar profile taken at the axis. As expected the particle count varies inversely with the density. The inverse linear relationship is demonstrated in figure 3.11 which plots the normalized product. The magnitude of fluctuations is somewhat larger than seen previously due to the extremely low particle count.

This calculation uses a total of 430,000 particles. The extremely low number of particles at the axis, around 0.25 per cell at the wall, indicates that approximately eighty times more particles would be needed to meet the nominal limit of twenty particles per cell. Although this limit can be relaxed somewhat for cells in the vicinity of the axis, the size of the simulation

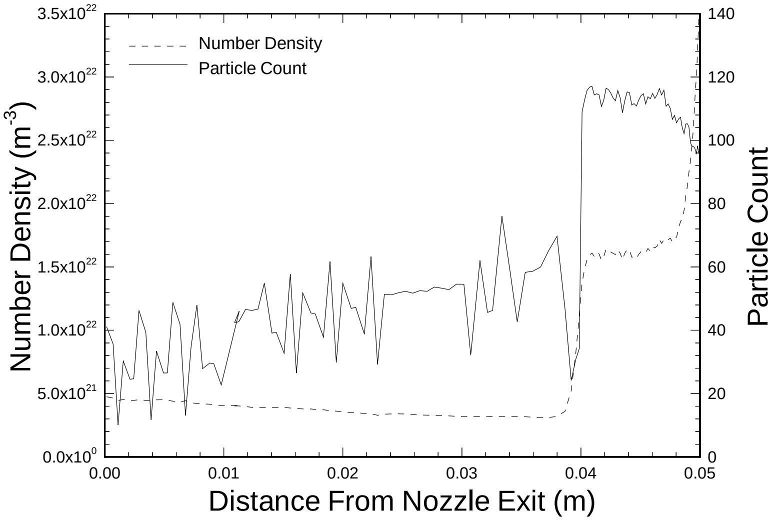

Figure 3.7. Density and particle count along stagnation line for hybrid grid, two-dimensional case.

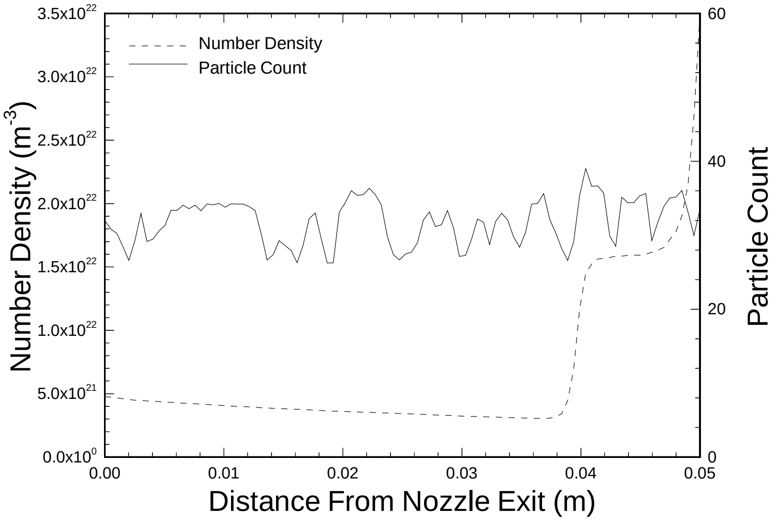

Figure 3.8. Density and particle count along stagnation line for two-dimensional case with variable time steps.

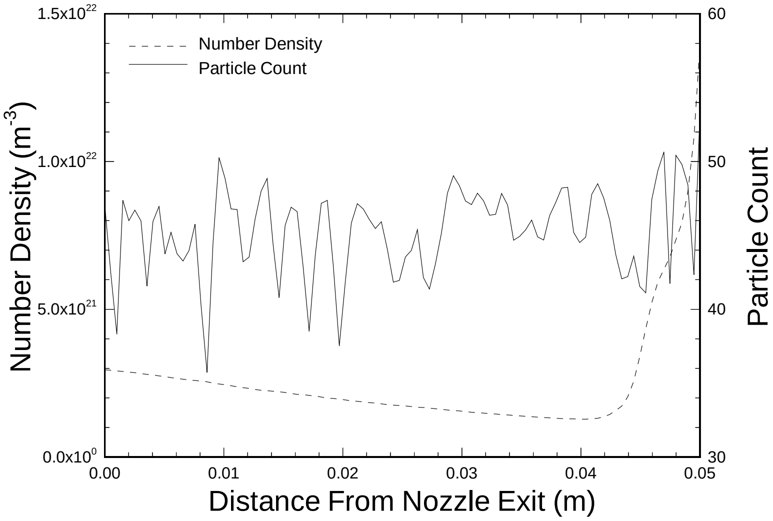

Figure 3.9. Density and particle count at radius equal to half the nozzle radius for axisymmetric case.

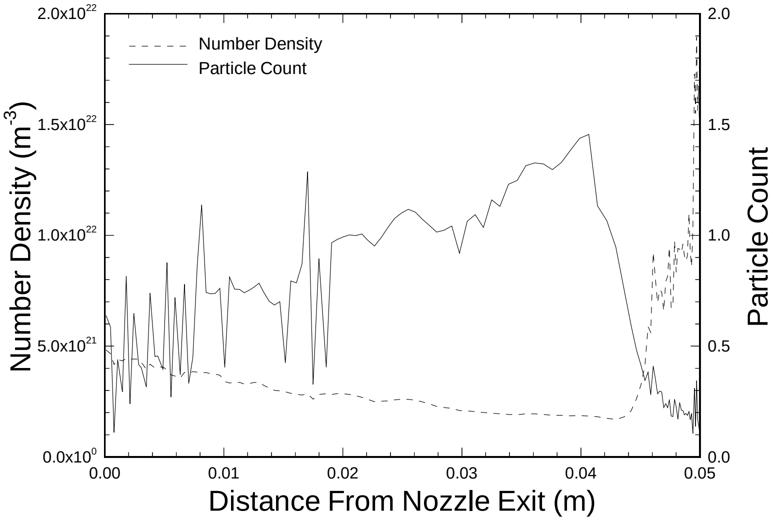

Figure 3.10. Density and particle count along axis for axisymmetric case.

would need to be increased by at least an order of magnitude to achieve reasonable accuracy.

Embedded Axial Grid

The major resolution difficulty with axisymmetric flows is in the vicinity of the axis. The scaling of particle count with radius means that there will be a small number of particles near the axis. This is especially evident in high density regions at the axis due to the additional scaling with density. In order to properly compute any axisymmetric flow this problem must be overcome.

There cannot be gradients in the radial direction at the axis of symmetry. It is thus appropriate to stretch cells in the radial direction in the vicinity of the axis. The use of stretched quadrilateral cells at the axis can have a dramatic effect on cell volumes and the resulting particle counts. Consider a cell which is an equilateral triangle of side length l located at the axis. The volume of this cell is given by:

$$\begin{aligned} \mathcal{V} &= 2\pi R_C A_P \\ &= \frac{1}{2}\pi l^3 \end{aligned} \tag{3.16}$$

A quadrilateral cell whose radial extent is stretched by a factor of 3 to $3l$ has the following volume:

$$\begin{aligned} \mathcal{V} &= 2\pi R_C A_P \\ &= 9\pi l^3 \end{aligned} \tag{3.17}$$

A stretched quadrilateral with a three to one aspect ratio therefore results in an eighteen fold increase in cell volume.

An embedded quadrilateral grid is used to resolve the axis of the impingement flow. Four rows of cells are used. Radial stretching by a factor of three is used for the first row, closest to the axis. A factor of two is used in the second row and the third and fourth rows are unstretched. Within the compression region at the surface stretching by a factor of five is used for all rows to compensate for the rise in density. Figure 3.12 shows a portion of the embedded grid near the front of the impingement surface.

Figure 3.13 shows density and particle count along the axis using the hybrid grid. The number of particles per cell is at an acceptable level across the range and is fairly uniform. The peak occurring at the beginning of the compression region indicates that the stretching is somewhat excessive at this point. This results from the necessity to have smoothly varying cell sizes in the structured grid at the axis.

The total size of the simulation is unaffected by the change in grid structure and remains approximately 430,000 particles. Although the nominal minimum of twenty particles per cell is not reached in most cells, grid stretching results in an order of magnitude increase in particle count and resolution. The cost of accurately computing this flow would then be reduced by an order of magnitude.

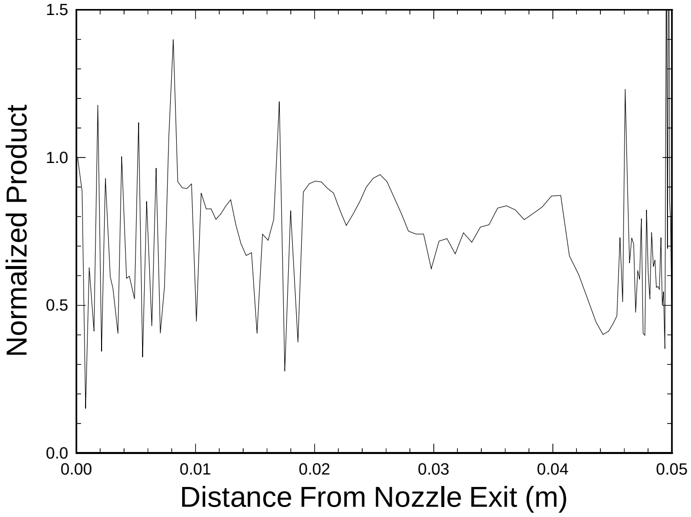

Figure 3.11. Product of density and particle count along axis for axisymmetric case.

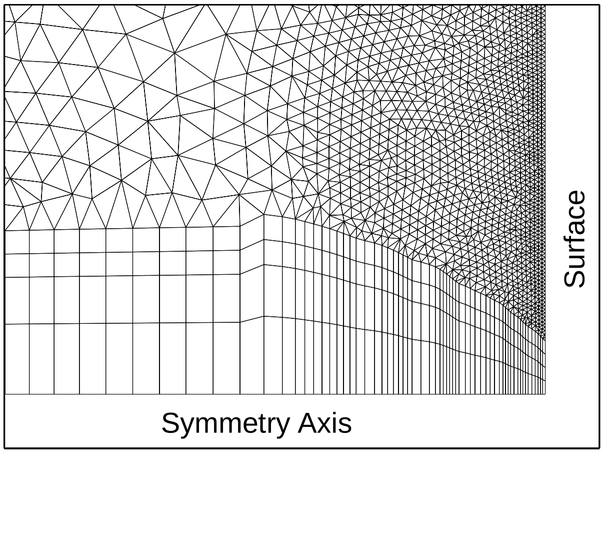

Figure 3.12. Hybrid grid at axis for axisymmetric case.

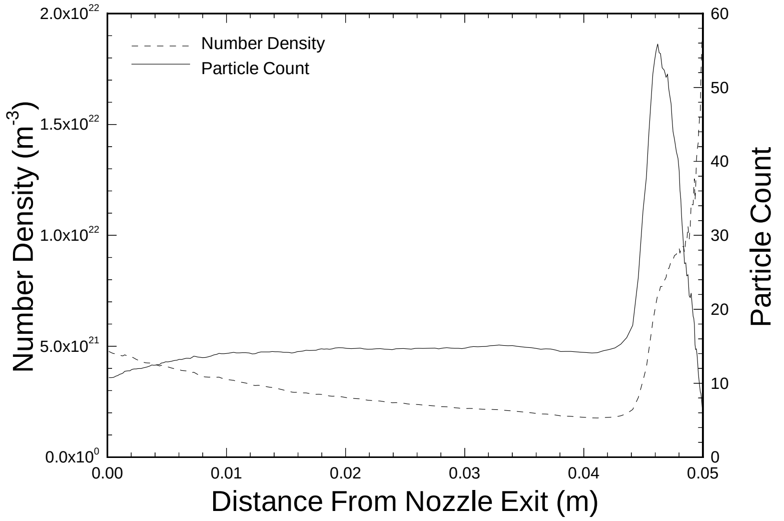

Figure 3.13. Density and particle count along axis for axisymmetric case using hybrid grid.

Chapter 4

Axisymmetric Impingement Flows

Before problems of engineering complexity can be considered the DSMC method and implementation must be properly validated for plume impingement flows. Direct comparison of numerical results with experimental data is necessary. Simple flows which can easily be investigated experimentally are most appropriate for this validation process.

4.1 Description of Problem

The impingement of a plume on an axisymmetric body is considered. Pressures at the surface of the body are compared with experimental measurements. An axisymmetric configuration has no practical application from an engineering standpoint. However, an axisymmetric problem is considerably less expensive computationally and is thus a good problem to examine for validation purposes.

A conical nozzle is used to generate a plume of nitrogen gas. The flow is designed to be representative of the plume from a resistojet thruster. The use of a single species gas allows examination of the fluid mechanics of the plume expansion and impingement without needing to include effects of multiple species transport or chemistry which occur in real electric propulsion devices. The nozzle has an exit to throat area ratio of 100:1 and a half angle of 20 degrees. Both heated and cold flow conditions are considered. Operating conditions are summarized in Table 4.1.

The impingement surface is a conical body located downstream of the nozzle on the axis of the plume. The cone has a half angle of 50 degrees and a base diameter of 50 mm. The axial position of the cone is varied with respect to the nozzle exit plane. Figure 4.1 shows a schematic of the nozzle-cone configuration.

Simulations are performed in two stages. First, flow through the diverging portion of the nozzle is calculated. Conditions at the exit plane of the nozzle are then used as inflow conditions for simulations of the plume. Separation of the nozzle and plume into two calculations decreases the cost of simulating several cone locations as well as simplifying the individual calculations.

Table 4.1. Nozzle operating conditions.

Flow rate	68 mg/s
Stagnation Pressure	6330 Pa
Stagnation Temperature	700 K (heated) 300 K (cold)

4.1.1 Experimental Measurements

Experimental measurements of surface pressure for this configuration were conducted in a vacuum chamber at NASA Lewis Research Center. The facility is described in detail in Ref. [17]. An ambient pressure of 2×10^{-2} Pa was maintained in the chamber.

Pressure measurements were made using four static taps located on the cone. The taps were arranged symmetrically along the cone surface, 10 mm and 20 mm from the cone tip. Figure 4.2 shows the location of the taps. Two pressure values characterize the pressure distribution for each flow condition and cone location. Measurements were taken with the cone located between 50 mm and 400 mm from the nozzle exit.

4.2 Resistojet Nozzle Flow

Simulations of the nozzle expansion begin at a point just beyond the throat and extend into the near plume. The subsonic flow upstream of the throat is continuum in nature and is not necessary to use DSMC. At the throat the Knudsen number based on throat radius is 2.5×10^{-3} for the heated flow and 1.5×10^{-3} for unheated flow. Rapid expansion in the diverging section of the nozzle causes the flow to become rarefied and transitional. The gas plume is assumed to expand into a vacuum.

Conditions at the inlet of the nozzle simulations are generated in two ways. The heated flow calculation uses results from a continuum CFD calculation of the nozzle (see Ref. [10]) to generate an inlet profile. The unheated flow uses isentropic theory and the stagnation conditions to determine the properties of the expansion at a Mach number of 1.1. The resulting velocity, temperature and density are assumed to be constant across the inlet. A uniform profile neglects the boundary layer which occurs at the nozzle wall. However, the boundary layer in the vicinity of the throat is thin and should have only a small effect on the flow. A simulation of the heated flow using isentropic inflow conditions indicates that the use of a uniform profile had negligible effect on the simulations.

The nozzle walls are assumed to be diffusely reflecting with full accommodation. The wall temperature is assumed to be uniform at 500 K for the heated flow and 300 K for the unheated flow. Due to the low density of the problem the flow is assumed to be vibrationally and chemically frozen. Rotational relaxation is considered.

4.2.1 Computational Modelling

The geometry of a diverging nozzle is well suited to structured grids. The primary gradients of the flow are axial due to expansion. Radial gradients are a result of the boundary layer. Under the low density conditions being considered the boundary layer is thick and the gradients are relatively small. As a result spatial resolution is more important in the axial direction than the radial.

Cells dimensions in the axial direction are proportional to the local mean free path. Isentropic theory is used to calculate the expected mean free path distribution in the nozzle. In the radial direction cell spacing is uniform with the exception of the two rows of cells closest to the axis. These cells are stretched by a factor of three and two in order to increase particle resolution. Figure 4.3 shows the grid used for the heated flow calculation.

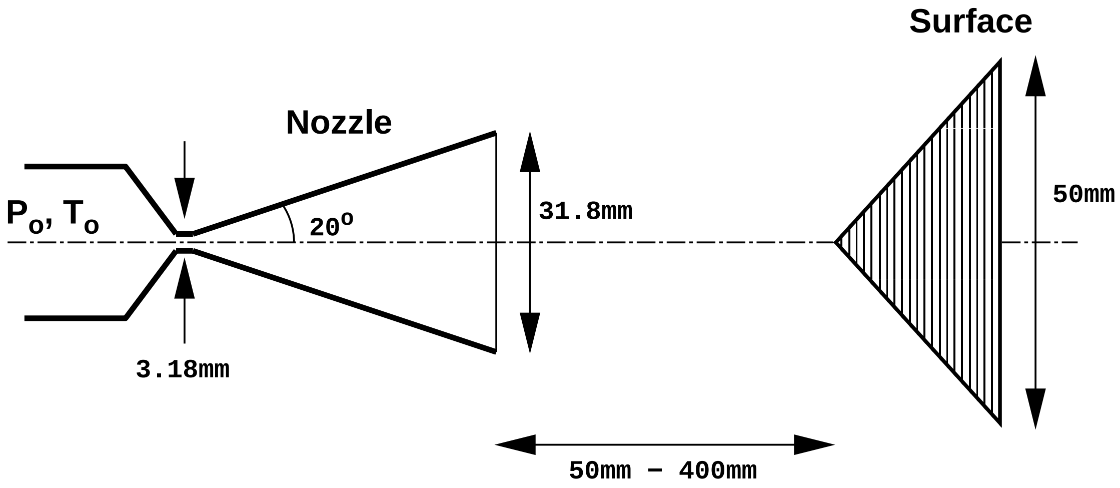

Figure 4.1. Schematic of nozzle-cone configuration.

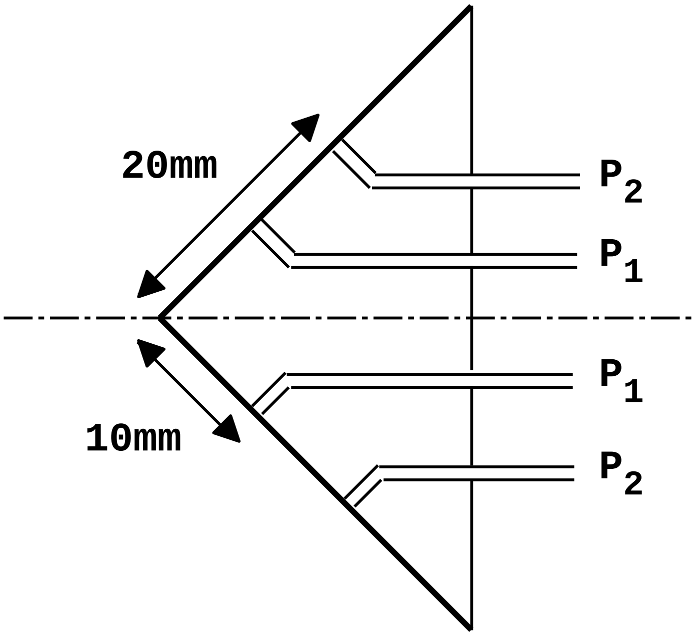

Figure 4.2. Schematic of pressure taps.

Variable time scales are employed to reduce convergence time and improve the particle distribution. Time steps are scaled linearly with the axial size of the cells. Since this dimension is proportional to the mean free path the time step is inversely proportional to the density. The reference time step is set at the nozzle throat.

Variable weight scales are used to counteract the increase in particle count with radial distance due to axisymmetry. A simple scheme is employed which uses different weights in each of four regions. Each region consists of a number of rows of the structured grid extending from throat to outflow boundary. The particle weight of regions increases away from the axis.

The strategy for domain decomposition is to group the cells by row of the structured grid. Each processor is assigned a number of rows of cells. Domain boundaries are approximately parallel to streamlines using this simple breakdown. The number of rows assigned to each processor is chosen to load balance the processors based on the results of a low particle count, preliminary simulation.

4.2.2 Computational Cost

Table 4.2 summarizes the size and cost of the nozzle calculations. The simulations were performed using four processors of an IBM SP-2.

The parallel efficiencies per time step obtained in these calculations are summarized in Table 4.3. The decomposition scheme is designed to be optimal at steady state. As a result, the efficiency begins at a minimum value and then increases as the transient progresses. Efficiencies comparable to steady state levels are reached after approximately 5,000 time steps or one quarter of the transient. Steady state efficiencies are high due to the highly localized nature of the DSMC algorithm and the structure of the nozzle flow which allows a simple decomposition scheme to achieve low communication.

4.2.3 Flow field Results

Heated Flow

The nozzle flow field for the heated case is shown in figure 4.4. Contours of velocity (top) and number density (bottom) display the expansion and acceleration of the gas which occurs through the nozzle. The boundary layer which develops at the nozzle wall is thick due to the low density flow.

For purposes of calculating plume impingement the exit plane properties are of particular interest. Data from this plane is used as inflow for the subsequent plume calculations. A previous study compared DSMC simulations of the heated nozzle flow with experimental measurements of pitot pressure at the exit plane. The DSMC results were found to agree with experimental measurements to within 3%. The results of this previous study are used to validate the current calculations.

Figure 4.5 compares velocity components in the exit plane from the current simulation and the previous study. Figure 4.6 shows a similar comparison of number density profiles. Excellent agreement is found between the two data sets across the nozzle exit.

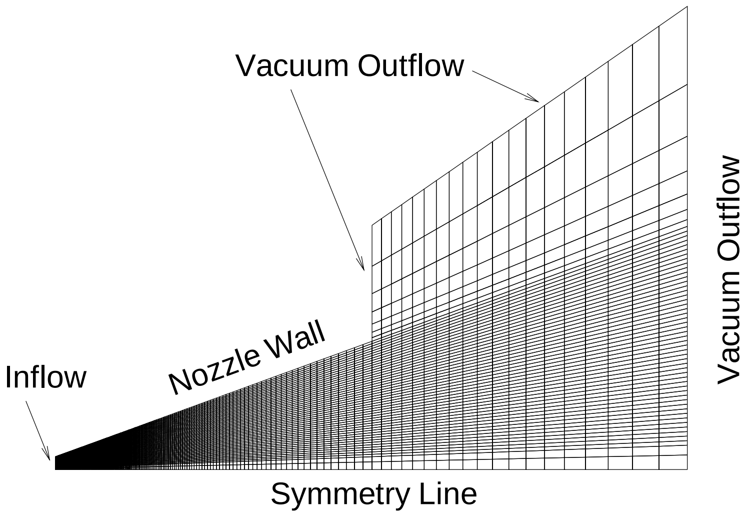

Figure 4.3. Computational grid for nozzle simulation with heated flow.

Table 4.2. Computational data for nozzle simulations.

	Heated Flow	Unheated Flow
Grid Size	340 × 58	420 × 58
Number of Particles	320,000	525,000
Transient Time	20,000 steps	20,000 steps
Sampling Steps	5,000 steps	5,000 steps
Calculation Time	5 hours	7 hours

Table 4.3. Parallel efficiencies for nozzle simulations.

	Heated Flow	Unheated Flow
Initial η_t	75 %	75 %
Initial $\eta_{i,min}$	67 %	55 %
Steady State η_t	95 %	97 %
Steady State $\eta_{i,min}$	92 %	94 %

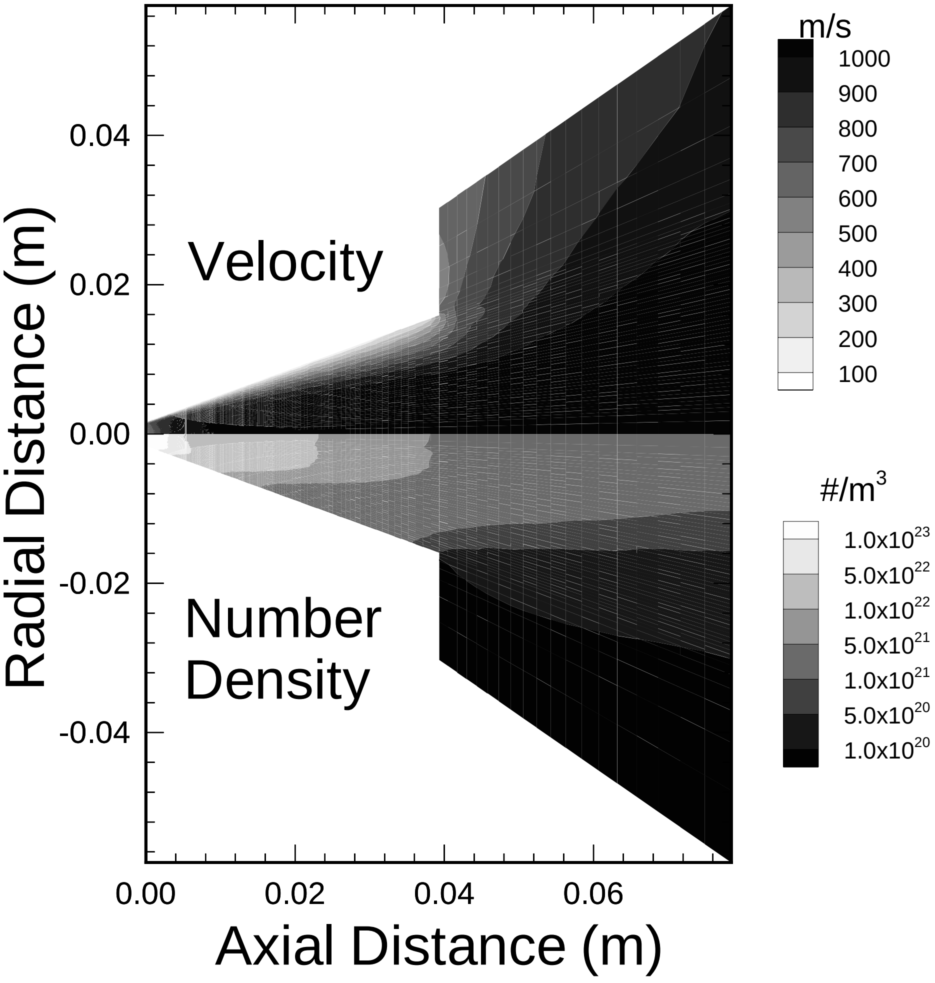

Figure 4.4. Contours of velocity magnitude and density for heated flow.

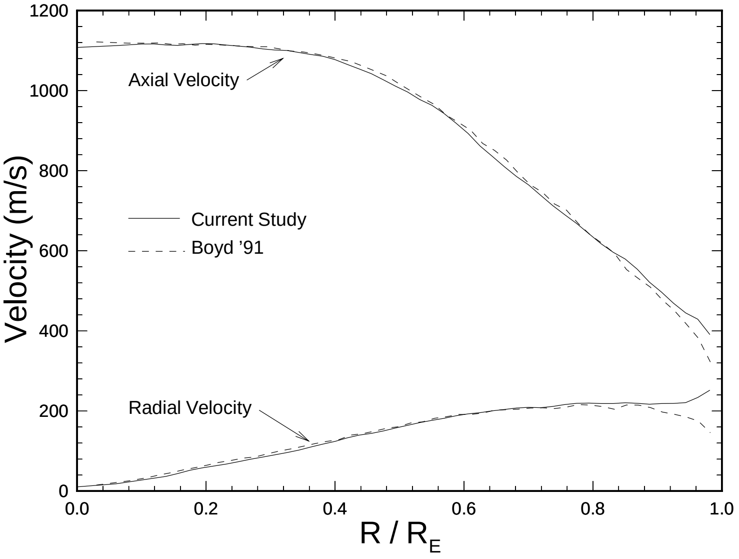

Figure 4.5. Exit plane profiles of velocity components for heated flow.

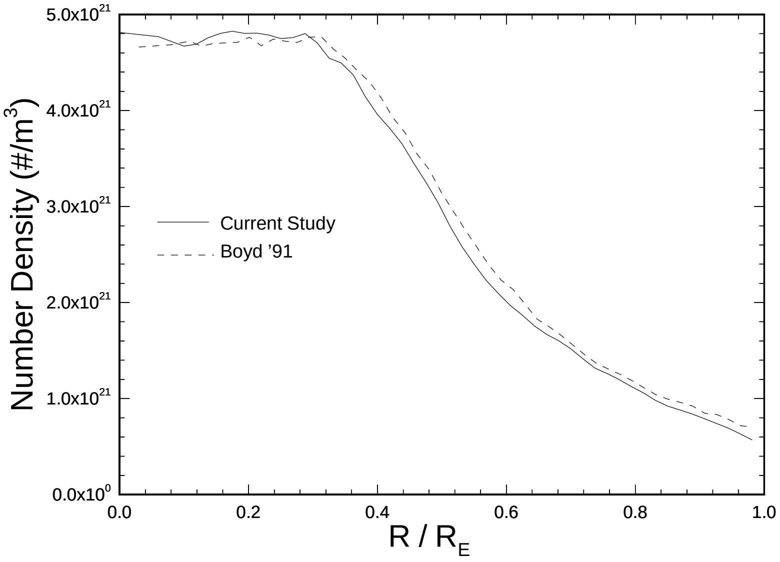

Figure 4.6. Exit plane profiles of number density for heated flow.

Translational and rotational temperature profiles at the exit plane are plotted in figure 4.7. Again good agreement is found between the current simulation and the previous work. The relatively small difference between translational and rotational temperatures are an indication that the flow is not significantly transitional at this point. As the flow becomes rarefied and the number of collisions drops energy is locked into rotational modes so that the rotational temperature is larger that translational temperature.

The breakdown parameter defined by Bird [4] measures the level of nonequilibrium of an expanding flow. In steady flow it is given by:

$$P = \frac{\sqrt{\pi}}{2} \frac{U}{\sqrt{2RT}} \frac{\lambda}{n} \left| \frac{dn}{dx} \right| \tag{4.1}$$

where the spatial derivative $\frac{dn}{dx}$ is evaluated along streamlines. Initial breakdown of continuum behavior has been found to occur when the breakdown parameter is approximately equal to 0.02. Using streamlines in the simulated flow, it is possible to evaluate the breakdown parameter at the exit plane for the nozzle flow. Figure 4.8 shows the breakdown parameter as a function of normalized radius in the exit plane. From this plot it can be seen that the flow is nonequilibrium across the entire exit plane. The degree of nonequilibrium is relatively small in the central core flow and increases rapidly in the boundary layer as the density falls.

The use of properties from a nonequilibrium portion of the flow to generate inflow conditions for another simulation represents a potential source of error. The usual DSMC inflow boundary condition assumes that the gas can be described by a Maxwellian velocity distribution and Boltzmann energy distributions. The magnitude of the breakdown parameter indicates that this is not the case at the exit plane. The use of equilibrium inflow distributions will not correctly sample the properties of a particle leaving the nozzle. This potential source of error is considered to be acceptable for this particular problem, however. The core flow is most important for determining impingement properties on a body located at the flow axis. Since the breakdown parameter in the core flow is close to the breakdown point it is likely that the distributions vary only a small amount from equilibrium. Therefore, using exit plane properties as inflow to the plume simulation appears reasonable.

Unheated Flow

No experimental measurements or previous computational studies are available for comparison of results for the unheated nozzle flow. Consequently, a qualitative analysis of the simulation results follows.

Exit plane profiles of velocity components, number density and temperatures are shown in figures 4.9, 4.10, and 4.11 respectively. The profiles show the same qualitative structure as seen in the heated flow profiles. The effect of lower temperature stagnation condition is seen in lower temperatures, lower velocities and higher densities.

The breakdown parameter at the exit plane is plotted in figure 4.12. The core flow is again close to the breakdown limit of P = 0.02 while the flow becomes significantly nonequilibrium as the boundary layer is traversed. As a result of higher densities the unheated flow is closer to the breakdown point in the core flow and thus is more acceptable for sampling inflow conditions.

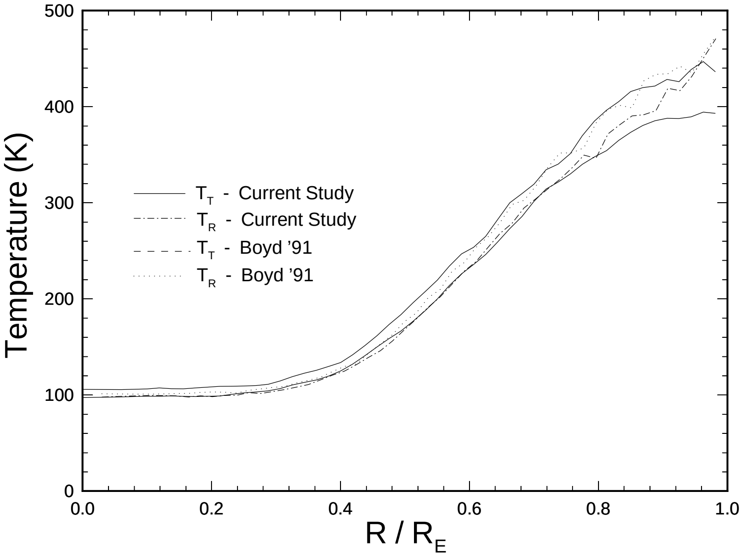

Figure 4.7. Exit plane profiles of temperature for heated flow.

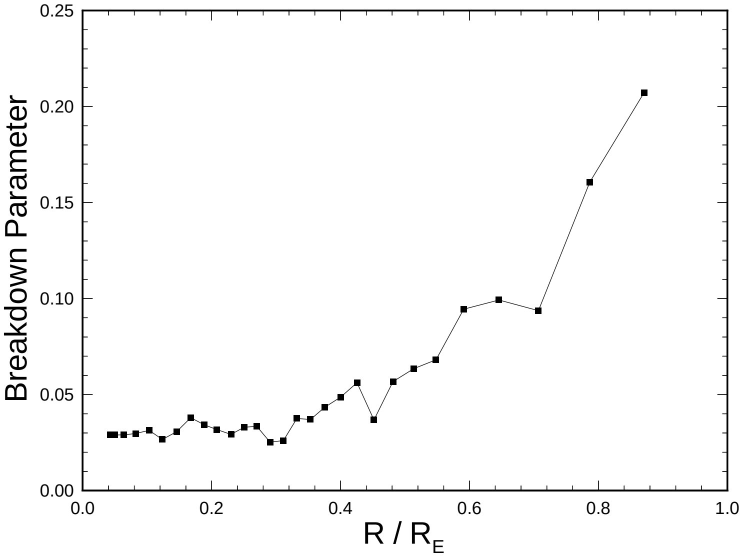

Figure 4.8. Breakdown parameter at exit plane for heated flow.

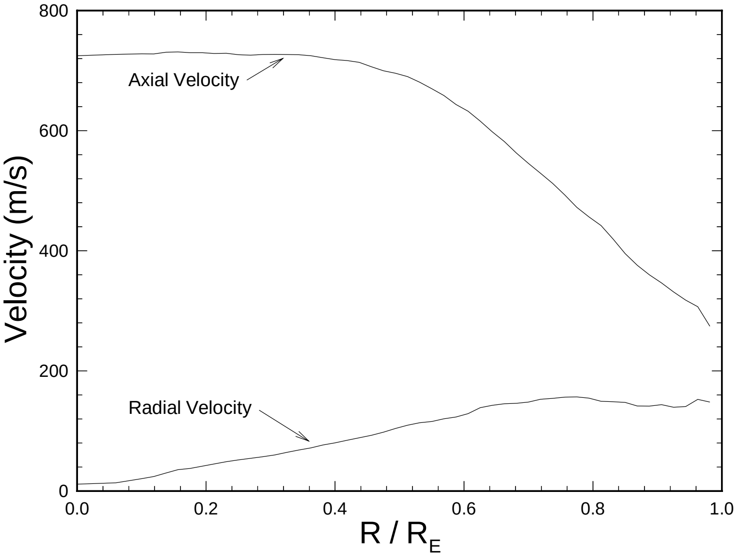

Figure 4.9. Exit plane profiles of velocity components for cold flow.

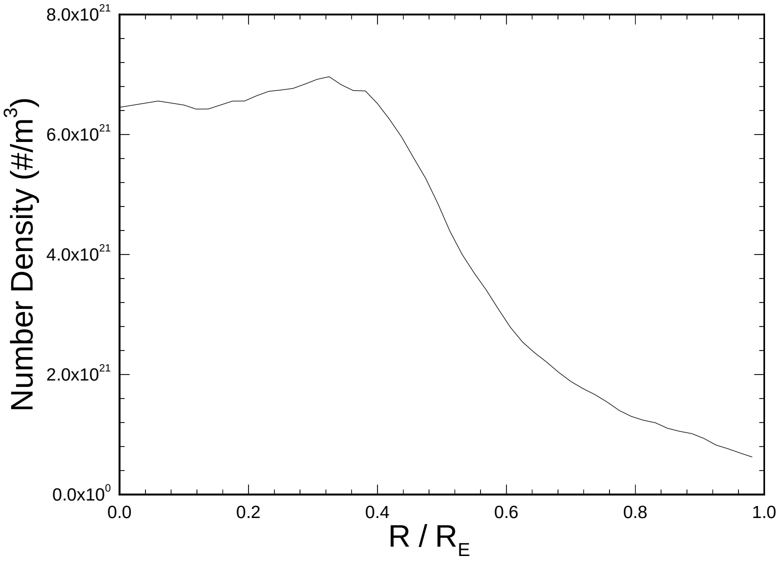

Figure 4.10. Exit plane profiles of number density for cold flow.

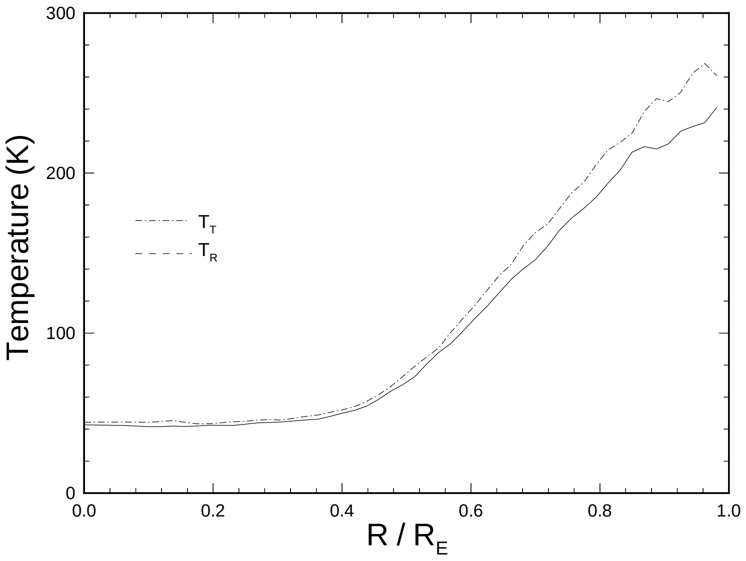

Figure 4.11. Exit plane profiles of temperature for cold flow.

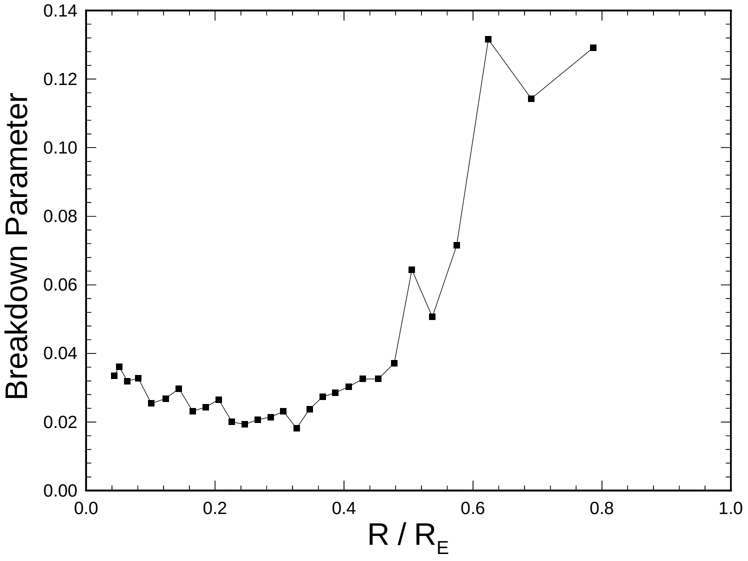

Figure 4.12. Breakdown parameter at exit plane for cold flow.

4.3 Plume Flow

The gas plume from the nozzle and its interaction with a body located downstream is simulated in order to determine impingement effects. The impingement cone is located on the axis of the plume at distances from the nozzle exit plane ranging from 50 mm to 400 mm. Conditions at the surface of the impingement body are of primary importance from an engineering standpoint. Surface pressure calculated by DSMC is compared with experimental data.

Simulations of the plume begin at the exit plane of the nozzle. Macroscopic exit plane properties (density, temperature, and velocity) from simulations of the nozzle are used as inflow conditions for both heated and unheated flows. Equilibrium distributions are assumed at the inflow even though the flow is transitional at this point. The degree of nonequilibrium is sufficiently small that employing equilibrium distributions should have minimal impact on the calculation.

The plume is assumed to expand into a vacuum. The ambient back pressure of the experimental facility is sufficiently low (2×10^{-2} Pa) that it has not been included in the simulations. The axisymmetric computational domain extends beyond the body in the radial and axial directions. The impingement surface is assumed to be diffusely reflecting with full accommodation at 300 K.

4.3.1 Computational Modelling

A hybrid grid is necessary to provide reasonable resolution and computational efficiency. An unstructured mesh based on one mean free path per cell is used to fill the majority of the domain. However, the shock interaction in front of the body and the surface properties depend on resolution of the stagnation streamline at the axis. As a result of axisymmetric particle count scaling the axis of the flow is difficult to resolve. Quadrilateral cells which are stretched in the radial direction are employed to improve resolution.

When the impingement surface is close to the nozzle (up to 200mm) compression of the gas causes resolution problems at the surface. Quadrilateral cells which are stretched parallel to the surface are employed to improve particle resolution at the surface at the cost of some spatial resolution. When the body is located far from the nozzle (300 mm, 400 mm) the gas does not compress to any significant amount at the surface. In these cases, cells which are on the order of one third of a mean free path are employed at the surface in order to obtain sufficient spatial resolution of surface properties. An exception is the cell at the tip of the cone where the cell must be on the order of one mean free path to generate a sufficient particle count due to axisymmetric scaling effects.

Grid development strategies are the same for both heated and unheated plume simulations. The unheated simulations require a larger number of cells due to higher densities and a greater amount of radial spreading of the plume.

Time steps are scaled linearly with a characteristic length of the cells. For triangular cells this length is the average side length. For quadrilateral cells the cell length in the axial direction is used — this dimension is in the direction of the primary flow gradients and is scaled with mean free path.

Determination of an optimal domain decomposition for the plume flow is not a simple

Table 4.4. Summary of computational data for plume simulations.

	Heated Flow	Unheated Flow
Grid Size	5,000–7,000 cells	7,000–16,000 cells
Number of Particles	200,000–400,000	300,000–800,000
Transient Time	3,000 steps	2,500–3,000 steps
Sampling Steps	5,000 steps	5,000 steps

task. The plume flow field is more complex than the nozzle flow field with expansion and compression regions and a more complicated streamline structure. Rather than develop a decomposition which follows streamlines, a simpler scheme is used which groups cells according to their radial position. The number of cells assigned to each processor is chosen to distribute particles evenly among processors.

4.3.2 Computational Cost

A number of different plume simulations are calculated to cover the range of body locations for both heated and unheated flows. Table 4.4 summarizes computational costs for these calculations. For the same body location an unheated plume calculation is larger and more expensive because the plume spreads radially to a larger degree.

Load balancing is performed periodically during the transient stage to optimize parallel efficiency. Every five hundred steps the simulation is stopped and a new decomposition is calculated based on grouping by radial position. Table 4.5 shows typical parallel efficiencies for a plume calculation. The minimum processor efficiency is initially zero as a result of the decomposition scheme. Since particles enter the domain at the axis, the processor handling cells furthest from the axis initially has zero particles and essentially zero work. This inequity is quickly resolved as particles fill the domain.

4.3.3 Results

The flow fields of representative plume cases are shown in figures 4.13 and figure 4.14. Contours of velocity and number density are shown in figure 4.13 for heated flow at 100 mm while figure 4.14 shows unheated flow at 300 mm. In both cases, the plume expands from the nozzle exit with compression occurring at the surface of the body.

It is not entirely clear whether the static pressure taps used for the experimental study measure pressure in the gas or impact pressure at the surface. It is likely, however, that the taps measure gas pressure by establishing an equilibrium between gas in the tap and at the surface. Both methods will be used to calculate surface pressure from the DSMC simulations. Flow field pressure is evaluated at nodes adjacent to the surface and calculated as $p = nkT_T$. Impact pressure is evaluated at the surface and is momentum transferred to the surface per unit area per unit time.

Table 4.5. Typical parallel efficiencies for plume simulations.

Initial η_t	50 %
Initial $\eta_{i,min}$	0 %
Steady State η_t	94 %
Steady State $\eta_{i,min}$	89 %

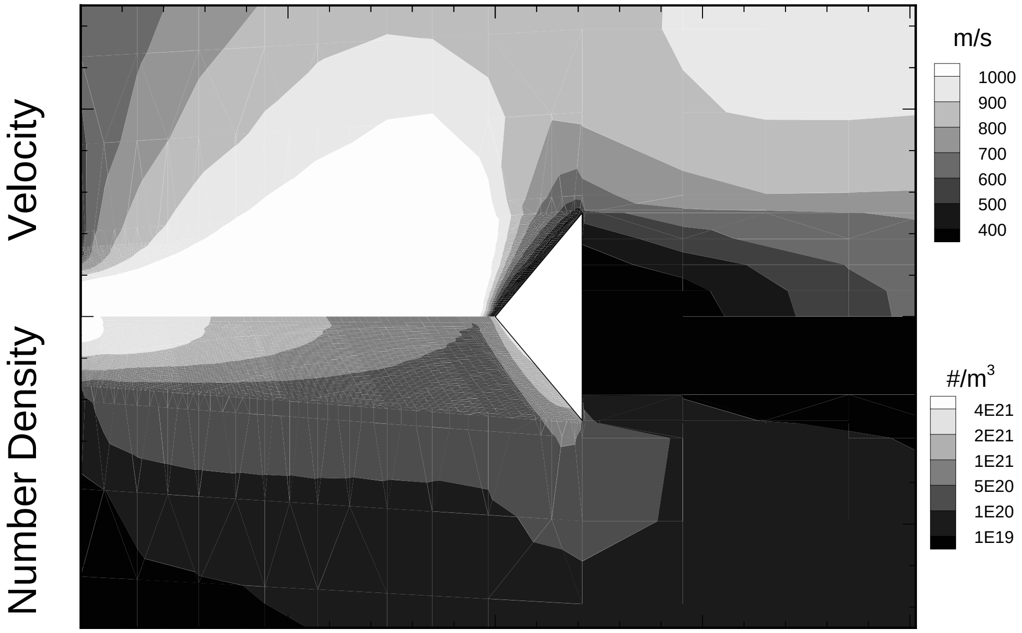

Figure 4.13. Contours of velocity and number density for heated flow at 100 mm,

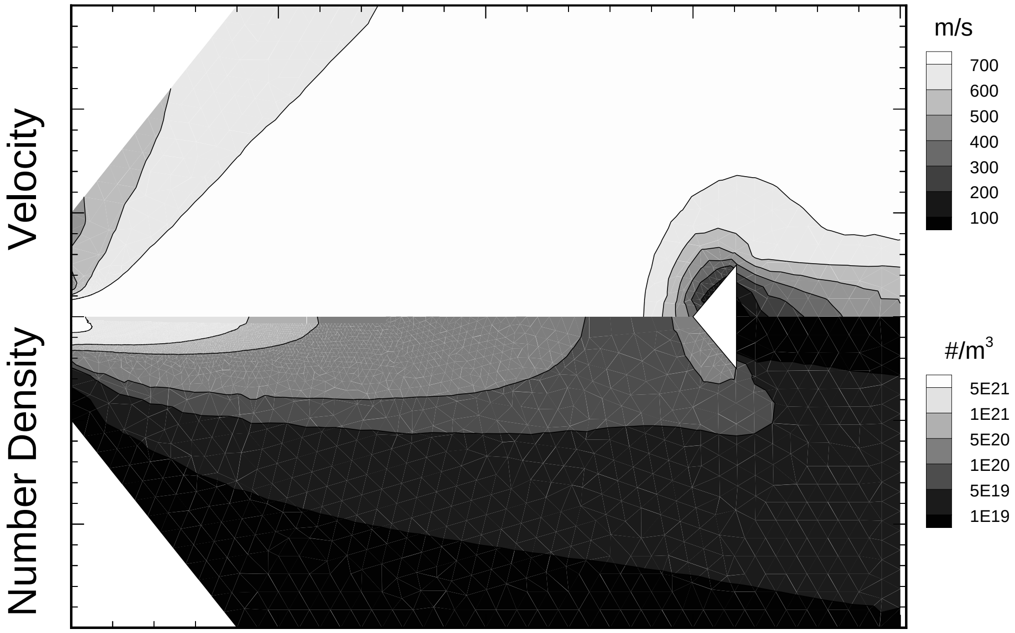

Figure 4.14. Contours of velocity and number density for cold flow, at 300 mm.

Heated Flow

Simulated profiles of pressure along the front surface of the cone are considered at 100 mm in figure 4.15 and at 400 mm in figure 4.16. In the 100 mm case, where the shock is strongest and the gas the most dense the pressure is highest at the tip of the cone and falls toward the base. Away from the axis both pressures are primarily due to the presence of a boundary layer behind the weak bow shock. Consequently, the two pressures are nearly equal. As the cone tip is approached a larger number of directly incident particles cross the shock and strike the surface causing an increase in impact pressure.

As the cone is moved away from the nozzle, the incident flow becomes more uniform and rarefied. As the density of the gas drops, the strength of the bow shock decreases and the amount of gas held in front of the body decreases. Consequently, the surface pressure decreases and the distribution becomes more uniform. A local minimum of flow field pressure appears at the axis because the point of maximum density moves back from the cone tip leaving a small pressure drop. The impact pressure becomes larger than the flow field pressure due to contributions from a larger number of directly incident particles which cross through the weak shock.

Surface pressure profiles are also compared with experimental data in figures 4.15 and 4.16. At 100 mm, both DSMC pressure profiles, impact and flow field, show good agreement with the experimental data. Both the magnitude and the trend are captured. At 400 mm, both DSMC pressures capture the trend of the data fairly well, being relatively uniform across the middle of the cone. The small rise in pressure moving back along the cone indicated by the measurements is not seen in either simulation profile. The flow field pressure shows better agreement in terms of magnitude than the impact pressure. Similar agreement is found across the range of axial positions.

The variation of surface pressure with cone location is summarized in figures 4.17 and 4.18. Figure 4.17 shows pressures taken at the location of the first experimental pressure tap (10 mm from the tip) while figure 4.18 shows pressures at the second tap (20 mm). As expected, the pressure falls as the distance from the nozzle increases. The various pressures are nearly proportional to the inverse of the square of the distance far from the nozzle. This is consistent with a point source model of the far field plume. The separation of impact pressure and flow field pressure with increasing distance is apparent.

There is good agreement between the simulation results and data from both pressure taps. While both methods of calculating pressure match the trend with distance, better quantitative agreement is found with the flow field pressure for large distances. This supports the contention that the static pressure taps measure gas pressure.

Unheated Flow

Surface pressure profiles and comparison with experimental data for unheated flows at 100 mm and 400 mm are shown in figures 4.19 and 4.20 respectively. The shape of the simulated pressure distributions are similar to those for the heated flow though at lower pressure levels. The simulated pressures show qualitative agreement with the data. However, both pressure methods significantly under predict the experimental measurements for both cone locations.

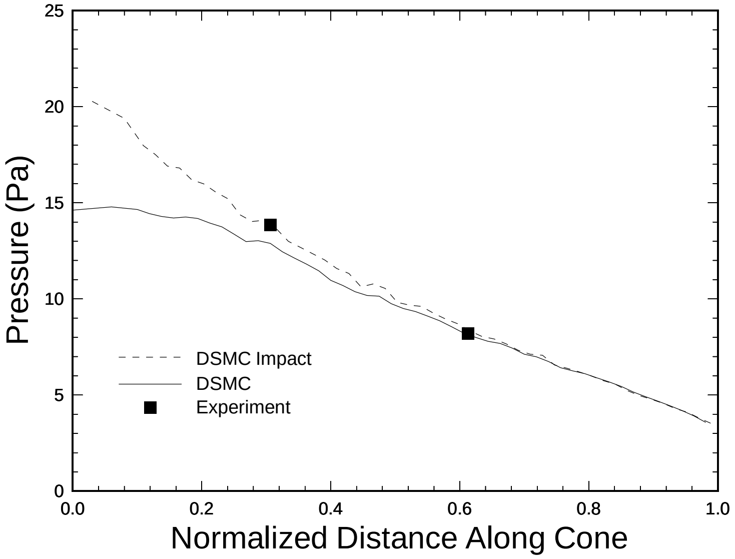

Figure 4.15. Surface pressure distributions for heated flow, 100 mm,

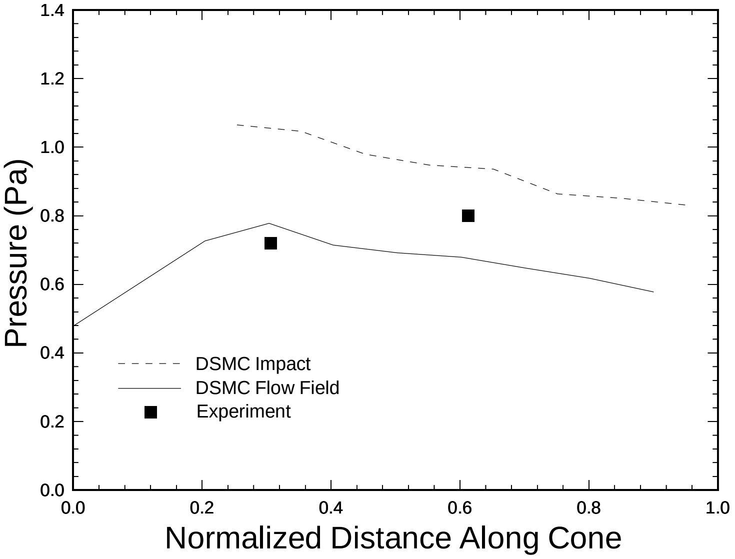

Figure 4.16. Surface pressure distributions for heated flow, 400 mm,

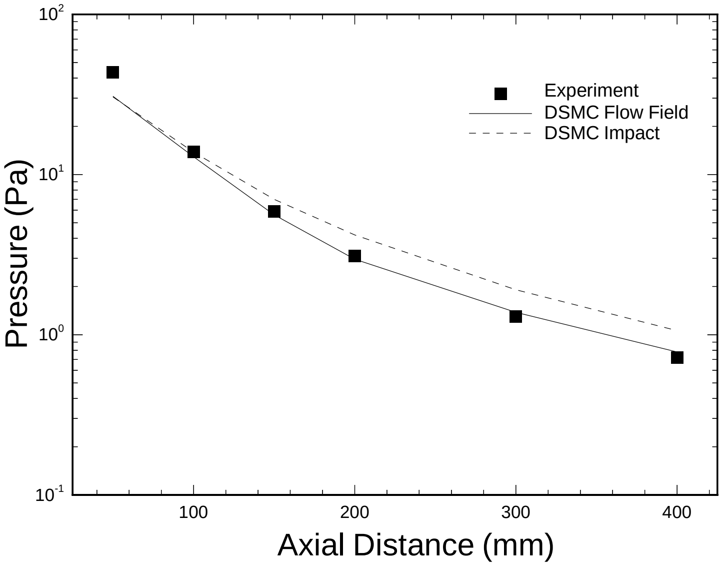

Figure 4.17. Variation of surface pressure with cone location. Data for heated flow at 10 mm from cone tip.

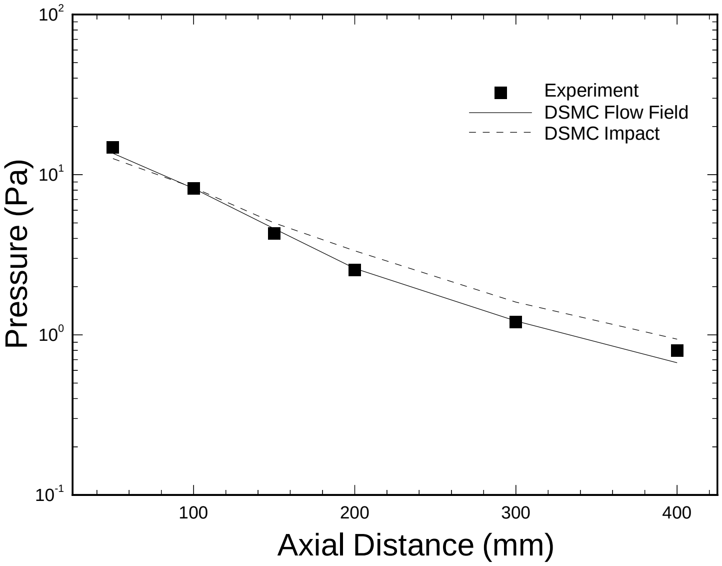

Figure 4.18. Variation of surface pressure with cone location, Data for heated flow at 20 mm from cone tip.

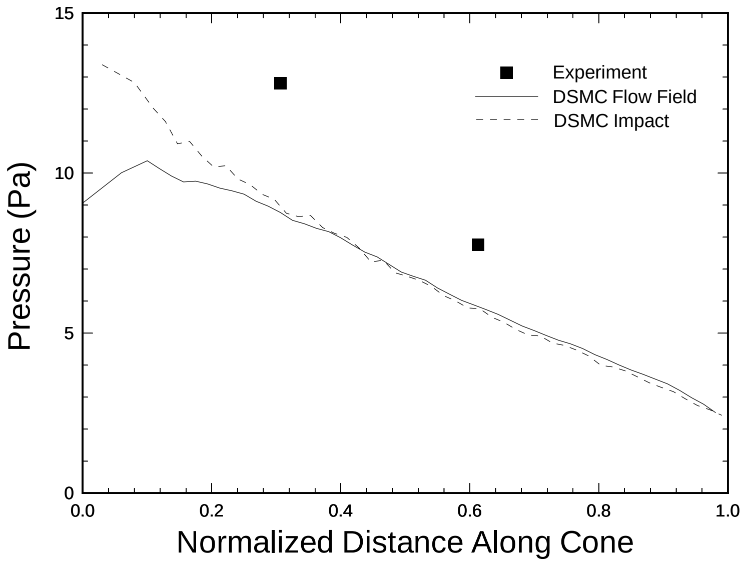

Figure 4.19. Surface pressure distributions for cold flow, 100 mm,

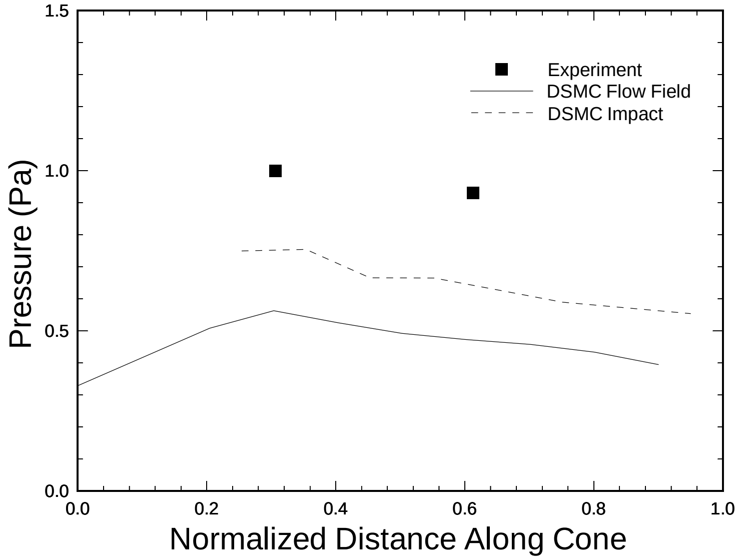

Figure 4.20. Surface pressure distributions for cold flow, 400 mm,

The variation of pressure with cone location at the two pressure taps is summarized in figures 4.21 and 4.22. The DSMC pressures consistently under predict the measurements across the range of locations. The trend is captured by the simulations as well as the inverse quadratic variation with distance in the far field.

In comparison with the heated flow simulations and experimental data, the cold flow, DSMC pressure values from both methods seem consistent. The lower plume velocities associated with the lower stagnation temperature should lead to a reduced momentum flux and lower pressures at the surface. This is observed in the DSMC results but not in the experimental data. Figure 4.23 compares experimental measurements for heated and unheated flows at the first pressure tap. Measured pressures for the cold flow are higher than the heated flow at large distances. This indicates that under prediction of the cold flow measurements may be a result of problems with the data.

4.4 Combined Nozzle/Plume Simulation

In order to ascertain the effect of splitting the impingement problem into two parts, the nozzle and plume are simulated simultaneously for one case. Heated flow with the body located at 100 mm is chosen as a representative case. The grid and scaling used for this calculation are a superposition of those used for the separate calculations.

Nozzle exit plane properties for the nozzle and combined simualtions are compared in figures 4.24 and 4.25. The profiles of velocity and density are essentially the same indicating that the separation has no effect on the flow at the exit plane. Pressures at the cone surface are compared in figure 4.26. Results from the combined simulation show good agreement with the separate plume simulation for both flow field and impact pressures. This demonstrates that performing the calculation in two stages has minimal effect on impingement properties.

Accuracy of the separate simulations requires continuity of flow properties across the interface. Figure 4.27 plots velocity along the axis in the vicinity of the nozzle exit plane for three simulations — nozzle, plume and combined. Similarly, figures 4.28 and 4.29 shows number density and translational temperature. The exit plane is located at $x = 0$ in these plots. The nozzle and plume simulations show virtually no discontinuity at the interface for any of the three properties. The combined simulation shows only small variation from the other two. This level of agreement further indicates that is reasonable to calculate the impingement problem in two parts. The agreement is expected considering the magnitude of the breakdown parameter at this point indicates that the flow is just beginning to become transitional.

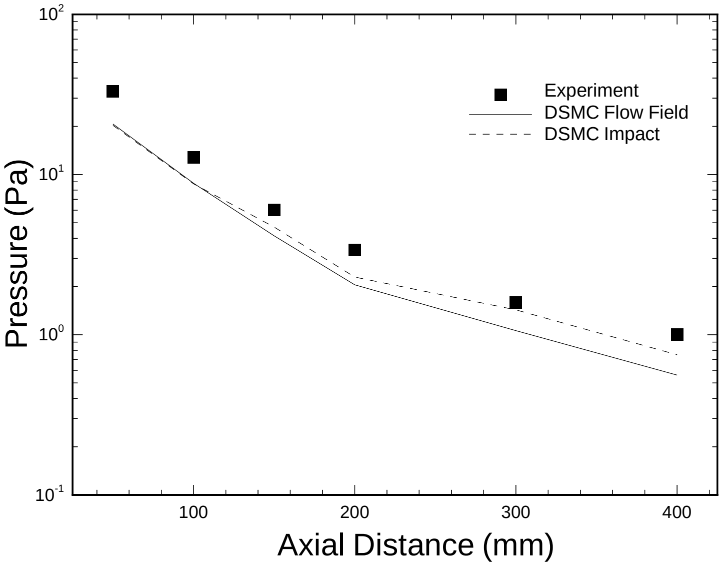

Figure 4.21. Variation of surface pressure with cone location. Data for cold flow at 10 mm from cone tip.

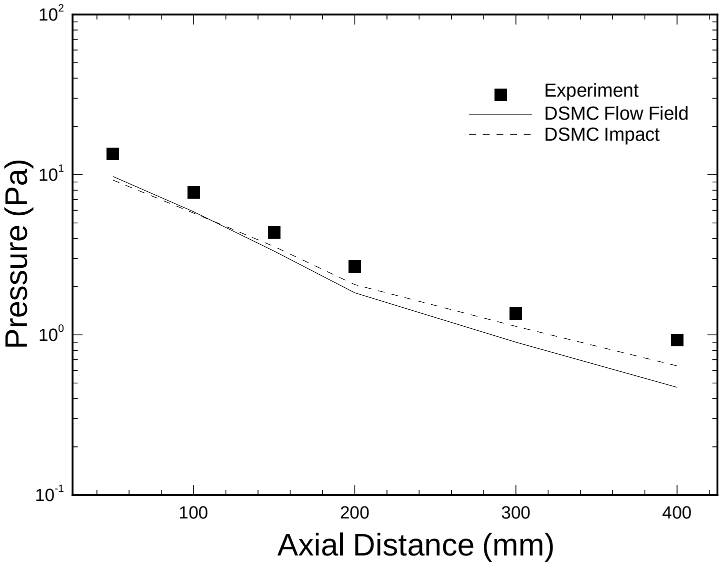

Figure 4.22. Variation of surface pressure with cone location. Data for cold flow at 20 mm from cone tip.

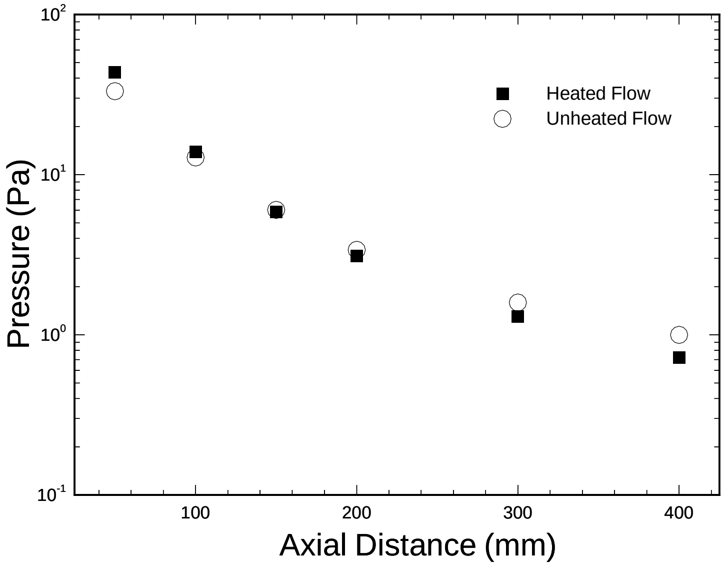

Figure 4.23. Comparison of heated and unheated experimental data at 100 mm.

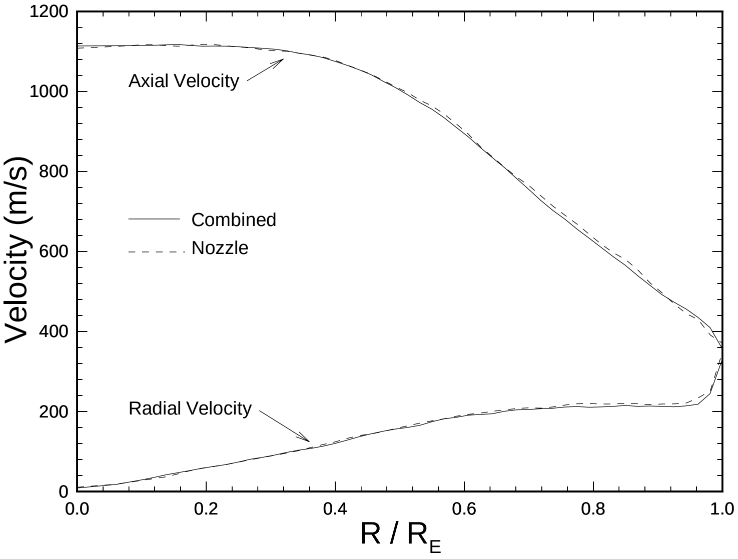

Figure 4.24. Comparison of exit plane velocities between nozzle simulation and combined nozzle-plume simulation.

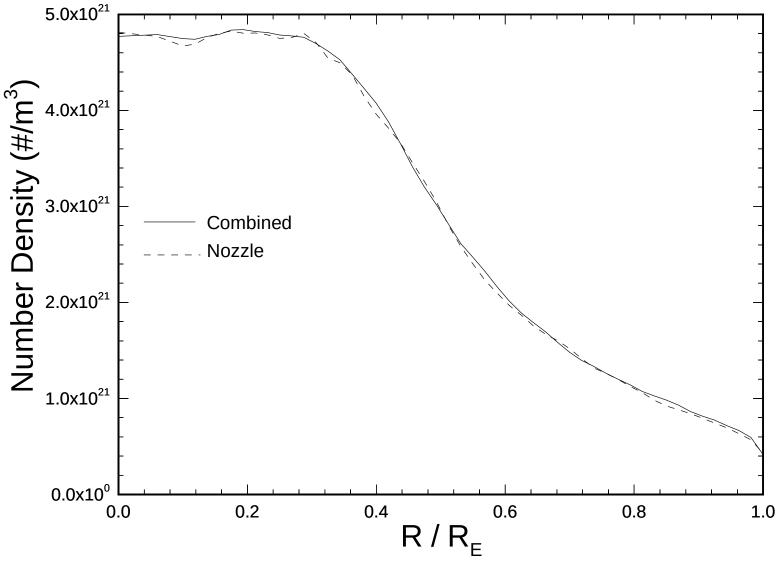

Figure 4.25. Comparison of exit plane number densities between nozzle simulation and combined nozzle-plume simulation.

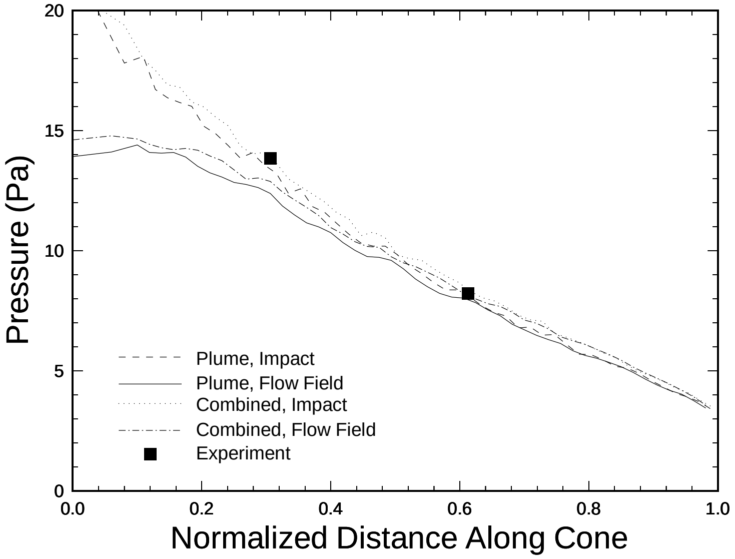

Figure 4.26. Comparison of surface pressure between plume simulation and combined nozzle-plume simulation.

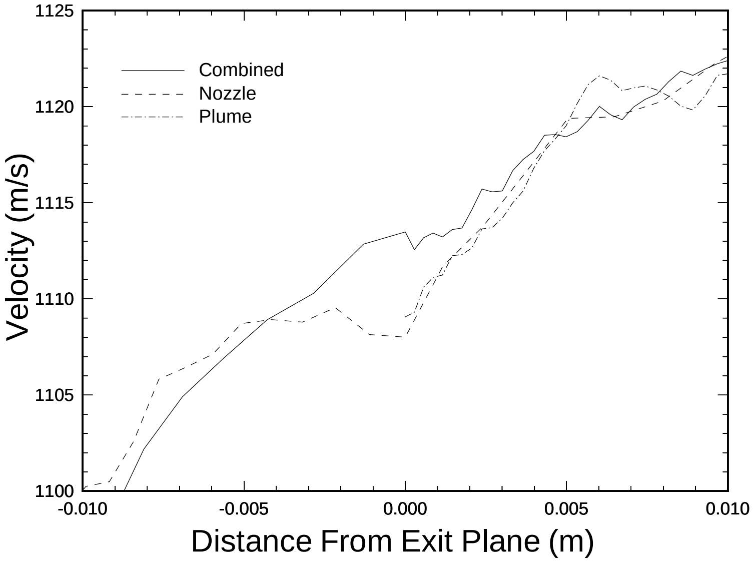

Figure 4.27. Comparison stagnation line velocities for nozzle, plume and combined simulations.

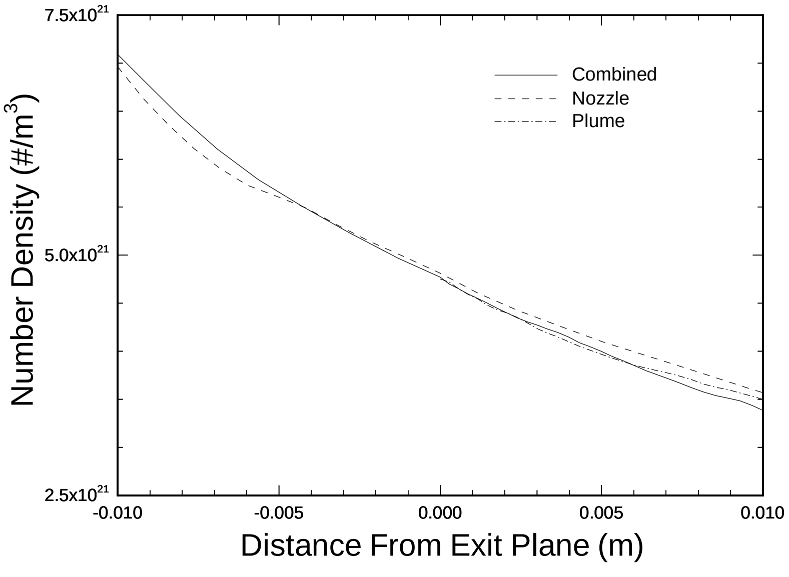

Figure 4.28. Comparison of stagnation line number densities for nozzle, plume and combined simulations.

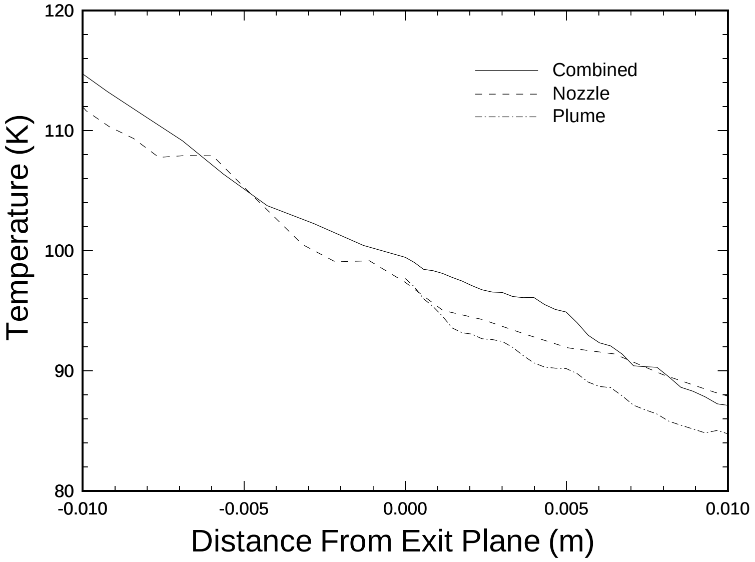

Figure 4.29. Comparison of stagnation line temperatures for nozzle, plume and combined simulations.

Chapter 5

Design and Development of a 3D Code

Impingement problems of engineering interest involve flows with complex three-dimensional configurations. Calculations of three-dimensional flows typically will be large and can quickly become impractical if not performed efficiently. A robust three-dimensional implementation of the DSMC method is required to accurately simulate flows of this type. A three-dimensional version of the MONACO software is developed to meet this need.

5.1 Design Requirements

Of primary importance in the development of a three-dimensional version is the retention of all important features of the original code. The design of the base MONACO localizes most elements related to the dimensionality of a problem within the geometry library. This library can then be replaced with a three-dimensional version to produce a three-dimensional code while retaining the memory and parallel management features and physical models of the base code. The important features of the geometry library are the computational grid and cell structure, particle movement and placement of new particles.

Two principles guide the design of the three-dimensional geometry library. First, the implementation should be as flexible in terms of grid type and structure as the original code. This facilitates the meshing of complex domains and allows the user to tailor the grid to suit the problem. Second, the routines other than initialization should be efficient to allow reasonable computation times for large three-dimensional calculations.

5.2 Grid Definition

The cell centered data structure and the requirement of completely flexible grids indicate the need for general unstructured grids. In three dimensions this is a computational domain composed of n-sided, convex, polyhedral cells. The sides of a polyhedral cell are termed faces. Cells with different numbers of faces or faces with different numbers of sides can be used together within the same grid. The only requirement is that each cell face can have exactly one neigboring cell.

Definition of general unstructured grids in 3-space is a complex task. Specification of the node locations is straightforward. However, unlike two-dimensional, unstructured grids, a cell is not completely specified by a series of nodes. Additional information about how the nodes are connected is necessary to completely specify the geometry of a cell. Rather than define cells explicitly, the most general, efficient method of specifying the grid is to define all of the faces in the grid. The cells are then defined implicitly.

A grid face is defined by a series of nodes that form a planar polygon. Any number of

Table 5.1. Summary of general 3D grid format.

Grid header:

```
    (number of nodes), (number of cells), (number of faces)
```

For each node:

```
    index, X, Y, Z
```

For each face:

```
    index, node-1, node-2, ... node-n
    index, (left neighbor), (right neighbor), (Boundary condition)
```

nodes is possible provided each falls within the plane. The requirement that cells be convex indicates that faces must also be planar and convex. Each face connects two cells or one cell and a boundary condition. The indices of the neighboring cells are part of the face definition. Cells can be constructed through this implicit definition using all faces which comprise the cell.

A complete grid definition is a list of nodes followed by a list of face specifications. Boundary conditions are included as part of the face definitions. The grid format is summarized in table 5.1.

The general, unstructured grid format does add complexity to the grid specification when compared to pure tetrahedral or structured grids. However, the added flexibility can be very useful when meshing complex flow domains. The type of cell used can be tailored to the flow problem or to portions of the domain. Hybrid grids consisting of prismatic and tetrahedral cells can be used to improve particle resolution as described in section 3.2.3. Rotated, axisymmetric grids can be used in regions with such symmetry. Particular types of grids may be appropriate for different portions of the domain. Using a grid format which allows a wide variety of cell types provides needed flexibility for complex problems. The additional work required to define the grid implicitly is either a preprocessing step or part of the initialization. As a result it does not reduce code performance. Defining the grid in terms of faces actually simplifies the initialization of parameters needed by the movement routine.

5.3 Particle Tracing

Particles in a DSMC simulation travel along linear trajectories between collisions. In three dimensions such a trajectory is defined in vector form by

$$\overline{x} = \overline{x}_0 + \overline{v}\, t \tag{5.1}$$

When a particle is moved during a time step it must be grouped with its destination cell. In order to avoid using an explicit sorting step, a particle tracing scheme is employed (see section 2.2.2). This requires calculation of the particle trajectories and their intersections with cell boundaries.

During a single time step a particle can either remain within its original cell or move out of that cell into a new one. If a particle stays within the original cell the task of movement is routine. If it leaves the current cell it becomes necessary to determine which cell face is intersected by the trajectory. A cell is defined by a series of arbitrarily oriented faces which together form a convex polyhedron. The faces are themselves defined by planes in three-dimensional space. Thus, the task is to calculate the first intersection between a line and a series of planes. Since movement routines constitute a significant fraction of the computational cost of a DSMC simulation, it is extremely important that this be done efficiently.

Faces in a grid must be planar and are defined by the equation of that plane. In vector form this is

$$\overline{n} \cdot \overline{x} = k \tag{5.2}$$

where $\overline{n}$ is a unit vector normal to the plane and pointing into the cell and k is the signed, perpendicular distance between the plane and the origin. This distance k is positive if the normal vector $\overline{n}$ is directed away from the origin and negative if $\overline{n}$ is directed towards the origin.

The first task when moving a particle is determining whether the particle will leave the current cell during the time step. This is done by first calculating the final position $\overline{x}_1$ of the particle.

$$\overline{x}_1 = \overline{x}_0 + \overline{v}\,\Delta t \tag{5.3}$$

This final position is checked against each face of the current cell to determine whether it lies on the side of the face that is inside the cell or outside the cell. The position $\overline{x}_1$ is "inside" if the following condition is met

$$\overline{n} \cdot \overline{x}_1 - k > 0 \tag{5.4}$$

In order to be within the current cell this condition must be met for each face that comprises the cell.

The proof of this method is as follows. Consider a plane defined in the standard way and a set of orthonormal vectors $\overline{n}$, $\overline{t}_x$ and $\overline{t}_y$ where $\overline{n}$ is the normal vector to the plane, $\overline{t}_x$ and $\overline{t}_y$ are vectors within the plane. These three vectors form a complete basis for 3-space. Now consider a point $\overline{y}$ defined by

$$\overline{y} = \overline{x} + C_1\,\overline{n} + C_2\,\overline{t}_x + C_3\,\overline{t}_y \tag{5.5}$$

where $\overline{x}$ is some point in the plane. Since the normal vector $\overline{n}$ is defined to point into the cell a positive value of C_1 indicates that the point $\overline{y}$ is on the side of this face that is within the cell.

To determine whether $\overline{y}$ lies within the cell bounded by this plane calculate

$$\begin{aligned} \overline{n} \cdot \overline{y} &= \overline{n} \cdot (\overline{x} + C_1\,\overline{n} + C_2\,\overline{t}_x + C_3\,\overline{t}_y) \\ &= \overline{n} \cdot \overline{x} + C_1\,\overline{n} \cdot \overline{n} + C_2\,\overline{n} \cdot \overline{t}_x + C_3\,\overline{n} \cdot \overline{t}_y \\ &= k + C_1 \end{aligned} \tag{5.6}$$

Since positive values of C_1 indicate a point within the cell, the condition for being within the cell can be expressed as

$$\overline{n} \cdot \overline{x}_1 - k = C_1 > 0 \tag{5.7}$$

as stated previously.

Once it has been determined that a particle will leave the current cell during this time step the next task is to determine which face of the current cell the particle will exit through. This amounts to finding the face which the particle trajectory intersects.

The intersection between the trajectory and a face is calculated by solving these equations simultaneously for the time t. In vector form

$$\overline{n} \cdot (\overline{x}_0 + \overline{v}\, t) = k$$

$$t = \frac{k - (\overline{n} \cdot \overline{x}_0)}{\overline{n} \cdot \overline{v}} \tag{5.8}$$

This can be calculated for each face comprising the cell. The face that results in the smallest positive time is the intersection face.

Faces which will have intersections in negative time can be skipped without explicitly performing the intersection calculation. The normal vectors $\overline{n}$ for each face are chosen to point into the current cell. Thus an intersection in positive time can only occur if

$$\overline{n} \cdot \overline{v} < 0 \tag{5.9}$$

Faces which do not meet this condition are skipped.

No information about the size of a face is used in the intersection time calculation. Only the plane which contains the face is considered. Thus the point of intersection could be outside the bounds of the face. If cells are convex this is not a concern as the intersection with minimum time will always be located within the bounds of the face.

The operations used in calculating trajectory-face intersections are primarily vector inner products. These operations involve a series of additions and multiplications which overlap well when computed on workstation processors. As a result, particle movement can be calculated with high efficiency.

The calculation of intersection time requires one division operation per face. In order to reduce the number of these expensive operations, the following method is used. For each face, define

$$\begin{aligned} A &= \overline{n} \cdot \overline{v} && (5.10) \\ B &= k - (\overline{n} \cdot \overline{x}_0) && (5.11) \end{aligned}$$

The face has a lower intersection time than the current minimum t_{min} if

$$A\, t_{min} > B \tag{5.12}$$

If this condition is satisfied reset the minimum time

$$t_{min} = \frac{B}{A} \tag{5.13}$$

In the worst case this method involves the same number of division operations as the direct approach. Under most circumstances, however, there will be a reduction in the number of divisions.

5.4 Placement of New Particles

Particles enter a simulation through inflow boundaries. The number of particles generated during each time step is determined by the velocity, density and temperature specified at that boundary. Each new particle is assigned a position, velocity and appropriate internal energies. Velocities and internal energies are sampled from equilibrium distributions based on the inflow conditions. Sampling these properties is unaffected by the dimensionality of the problem. The task of determining a position for each new particle is, however, significantly more complex in three dimensions.

When a particle enters the simulation its position is chosen at random from within the boundary area. The method for selecting this location must be such that the distribution of positions generated is uniform across the boundary. In two dimensions this amounts to choosing a point along a line segment. In three dimensions a point must be chosen from within a cell face. This is a polygonal area located on a plane of arbitrary orientation.

The simplest scheme for uniformly sampling points from a polygon in two-dimensional space is an accept-reject scheme. The minimum size rectangle which bounds the polygon is determined. Points are selected from within this rectangle using two random numbers. A point is accepted if it lies within the polygon and rejected if it does not.

When considering a planar polygon in three-dimensional space it is convenient to first transform the plane through rotation so that it lies parallel to the xy-plane. If the face is defined by equation 5.2, each vertex, $\overline{x}_i$ of the polygon is transformed to its position in the xy-plane, $\overline{x}'_i$, by the following operation

$$\overline{x}'_i = \overline{\overline{T}} \cdot \overline{x}_i - k\hat{e}_3$$

$\overline{\overline{T}}$ is a rotation matrix defined by

$$\overline{\overline{T}} = \begin{bmatrix} \overline{t}_x \\ \overline{t}_y \\ \overline{n} \end{bmatrix}$$

where $\overline{t}_x$ and $\overline{t}_y$ are orthonormal vectors within the plane of the face. Positions can then be selected in transformed space using the accept-reject method. The particle position is then transformed back to the grid coordinate system using the inverse transformation.

Chapter 6

Flat Plate Impingement

Impingement of a jet onto an inclined flat plate represents one of the simplest possible three-dimensional impingement problems. The relatively simple geometry allows it to be readily investigated both numerically and experimentally while still retaining three-dimensional effects. This problem is used to validate the accuracy and performance of the three-dimensional version of MONACO.

Two steps are involved in the verification of the three-dimensional code. First, normal impingement of the jet onto the plate is examined. Since this represents an axisymmetric problem, the results are compared with a calculation using the axisymmetric code. Second, simulations of the plate impingement are compared with experimental data for several orientations of the plate.

Free molecular analysis can be used to predict surface properties in rarefied plume flows. In order to assess the utility of this simple engineering model, surface properties from the DSMC simulations are compared with the results from free molecular analysis.

6.1 Description of Problem

The problem under consideration is a free jet impacting on a flat plate. A plume of molecular nitrogen is generated by a sonic orifice. The orientation of the plate is varied relative to the axis of the plume. Figure 6.1 shows a schematic of the configuration.

This study will consider one particular set of orifice inlet conditions — unheated flow ($T_0 = T_W = 300$ K) expanding from a stagnation pressure of 1000 Pa. The orifice itself is circular in cross section and 1 mm in radius and the flow is assumed to be sonic at the exit. The Knudsen number based on orifice radius is 8×10^{-3} at the orifice exit. These conditions ensure a rarefied plume flow suitable for calculation using the DSMC method. The impingement surface will be located a constant distance of 40 mm from the orifice measured perpendicular to the surface (distance L in figure 6.1).

6.2 Experimental Study

The plate impingement configuration was investigated experimentally at DLR in Germany. Measurements of impingement pressure and shear stress were taken by Legge [20] and measurements of heat flux were made by Döring [14]. A variety of stagnation pressures, plate orientations and separations were considered.

Pressure and shear stress data were obtained using a pressure balance which directly measured the force on a floating element. Heat flux data were obtained by measuring the rate of change of surface temperature using thermocouples. Data is taken on the surface along a line which is coplanar with the plume axis.

The experiments were performed in the high vacuum facility in Göttingen. The background pressure of the facility was 0.045 Pa during pressure and shear measurements with the stagnation pressure considered in the present study (1000 Pa). The background pressure was twice this value (0.090 Pa) during the heat transfer experiment.

Pressure and shear stress data were normalized in order to eliminate the effect of stagnation pressure and plate separation L. The following normalizations were used.

$$\hat{p} = \frac{p}{p_0}\left(\frac{L}{r^*}\right)^2 \tag{6.1}$$

$$\hat{\tau} = \frac{\tau}{p_0}\left(\frac{L}{r^*}\right)^2 \tag{6.2}$$

While this normalization does reduce the data to a significant degree some dependence on the stagnation pressure is still observed. A reduction in stagnation pressure results in an increase in normalized pressure and shear stress due to the increasing rarefaction of the plume. For the present study, data obtained using a stagnation pressure of 1000 Pa will be used whenever possible.

6.3 Free Molecular Analysis

Free molecular theory provides a limiting case for the surface properties at the plate. Under the flow conditions considered in this study it is likely that conditions at the surface will approach free molecular. If the plume density is assumed to decrease with the square of the distance from the exit and a corresponding increase in mean free path is considered, the Knudsen number at the surface will be on the order of 10. Thus free molecular theory is likely to provide a good estimation of surface properties.

Under the assumption of free molecular flow, surface properties are a function only of the incoming influx of particles. Particles scattering off the surface do not have any further interaction with the surface or the incoming gas. The pressure experienced at a point on the surface can be obtained by integrating normal momentum flux over the incoming and outgoing particle distribution functions [25]. If the incoming gas is assumed to have an equilibrium velocity distribution and the surface is assumed to be fully accommodating the pressure is given by:

$$\begin{aligned} p = \frac{\rho U^2}{2S^2}\Bigg[&\left(\frac{1}{\sqrt{\pi}}S\sin\alpha + \frac{1}{2}\sqrt{\frac{T_W}{T}}\right)e^{-(S\sin\alpha)^2} + \\ &\left((S\sin\alpha)^2 + \frac{1}{2} + \frac{1}{2}\sqrt{\frac{\pi T_W}{T}}(S\sin\alpha)\right)\left[1 + \mathrm{erf}(S\sin\alpha)\right]\Bigg] \end{aligned} \tag{6.3}$$

where α is the angle of incidence of the mean flow and S is the molecular speed ratio $U/sqrt2RT$. In the hypersonic limit for which $S \to \infty$ this becomes

$$p = \frac{1}{2}\rho U^2\left[2(\sin\alpha)^2 + \sqrt{\pi\frac{\gamma-1}{\gamma}}\sqrt{\frac{T_W}{T_0}}\sin\alpha\right] \tag{6.4}$$

The density of the gas can be modeled using the plume model of Boynton and Simons [26]. This model describes the density as a function of the throat density and location in the plume expressed in polar coordinates measured from the center of the exit plane. If boundary layer effects are neglected the density is given by:

$$\frac{\rho(r,\theta)}{\rho^\star} = A_P\left(\frac{r^\star}{r}\right)^2 f(\theta) \tag{6.5}$$

$$f(\theta) = \left[\cos\left(\frac{\pi}{2}\frac{\theta}{\theta_{lim}}\right)\right]^{2/(\gamma-1)} \tag{6.6}$$

where $\rho^\star$ refers to the exit plane density and $r^\star$ to the exit radius refer (the throat is used for a nozzle flow). A_P is a plume constant and θ_{lim} is the maximum turning angle of a streamline at the exit for inviscid flow. These parameters can be calculated analytically as a function of the stagnation conditions, gas properties and source geometry (see Ref. [21]). The flow conditions of the current study yield a plume constant $A_P = 0.617$ and a turning angle of 130.5°.

If the impingement surface is located sufficiently far from the orifice the gas can be considered fully expanded to the hypersonic limit. In this case the fluid velocity is approximated by the limiting result from isentropic theory, $U_{lim} = \sqrt{2C_pT_0}$.

The relations for plume density and velocity can be used with equation 6.4 to obtain an expression for pressure in terms of distance from the orifice, location in the plume, and the angle of incidence, α:

$$p = p_0 A_P f(\theta)\left(\frac{r^\star}{r}\right)^2\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}}\frac{\gamma}{\gamma-1}\left[2(\sin\alpha)^2+\sqrt{\pi\frac{\gamma-1}{\gamma}\frac{T_W}{T_0}}\sin\alpha\right] \tag{6.7}$$

If this is then normalized to eliminate the effects of stagnation pressure and orifice-plate separation according to equation 6.1 the following result is obtained:

$$\begin{aligned}\frac{p}{p_0}\left(\frac{L}{r^*}\right)^2 &= A_P f(\theta)(\sin\alpha)^2\frac{\gamma}{\gamma-1}\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}} \\ &\times \left[2(\sin\alpha)^2+\sqrt{\pi\frac{\gamma-1}{\gamma}\frac{T_W}{T_0}}\sin\alpha\right]\end{aligned} \tag{6.8}$$

The shear stress at the surface can be obtained through a similar development. In this case, tangential momentum is integrated over the incoming and outgoing particle distributions. The following expression is obtained:

$$\tau = 2p_0 A_P f(\theta)\left(\frac{r^\star}{r}\right)^2\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}}\left(\frac{\gamma}{\gamma-1}\right)\cos\alpha\sin\alpha \tag{6.9}$$

Normalizing this using equation 6.2 gives:

$$\frac{\tau}{p_0}\left(\frac{L}{r^*}\right)^2 = 2\cos\alpha(\sin\alpha)^3 A_P f(\theta)\left(\frac{\gamma}{\gamma-1}\right)\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}} \tag{6.10}$$

The energy transferred to the surface can be calculated as the difference between the integral of incoming and outgoing particle energies. The net heat flux for a fully accommodating surface is given by:

$$\begin{aligned} q'' &= \sqrt{2RT_0}p_0 A_P f(\theta)\left(\frac{r^\star}{r}\right)^2 \sin\alpha \left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}} \\ &\times \sqrt{\frac{\gamma}{\gamma-1}}\left[\frac{\gamma}{\gamma-1} - \frac{\gamma+1}{2(\gamma-1)}\frac{T_W}{T_0}\right] \end{aligned} \tag{6.11}$$

6.4 Simulations

6.4.1 Physical Modeling

The inlet flow at the sonic orifice is modeled as a macroscopically uniform stream of molecular nitrogen directed along the axis of the plume. The effects of a boundary layer at the exit are assumed to be small and are neglected. Inlet properties are calculated from isentropic theory assuming sonic conditions at the exit and stagnation conditions of 1000 Pa and 300 K. This corresponds to a velocity of 323 m/s, temperature of 252.2 K and number density of 1.53×10^{23} m^{-3}.

The effects of finite tank pressure in the experimental facility is simulated using a background gas pressure. The flow field is initially filled with a gas of uniform density and temperature representative of the tank. The background is maintained by applying these conditions as an inflow on the outer boundaries of the domain. The background density is based on a temperature of 300 K and a pressure of 0.045 Pa. This is equal to that measured in the experimental facility during impingement pressure and shear stress measurements. The higher back pressure observed during heat transfer measurements (0.09 Pa) should have minimal effect on heat flux to the surface.

The impingement surface is modeled assuming diffuse reflection with full accommodation to a constant surface temperature. The surface temperature is equal to the stagnation temperature of the plume (300 K). Diffuse reflection should adequately model a real surface particularly for large angles of attack.

6.4.2 Computational Modeling

3D calculations

Fully unstructured grids using tetrahedral cells are used. Grids are generated using an advancing front package called FELISA. Cell sizes are approximately scaled to the local mean free path. Some variation is present due to the lack of a direct adaption capability in the package. Figure 6.2 shows an example of a grid used for this problem showing only the surface mesh. The grid is that for the case with plume axis parallel to the surface ($\beta = 0°$). The majority of cells are in the vicinity of the nozzle where the density is highest. Compression at the surface is relatively small due to the low densities involved. Symmetry planes are employed to reduce the size of the domain. Two planes are used in the normal impingement ($\beta = 90°$) case and one each in the other cases.

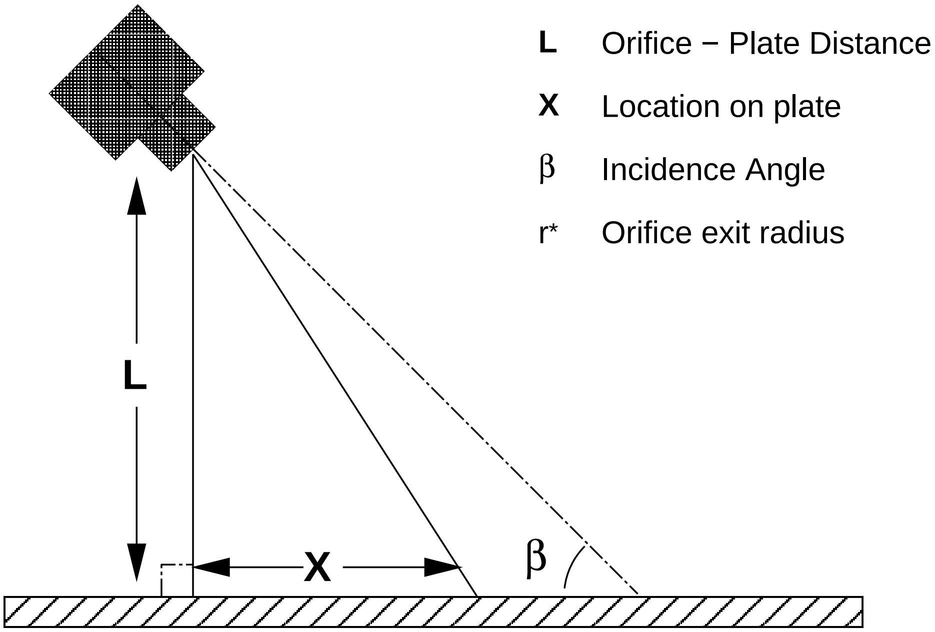

Figure 6.1. Schematic of flat plate impingement.

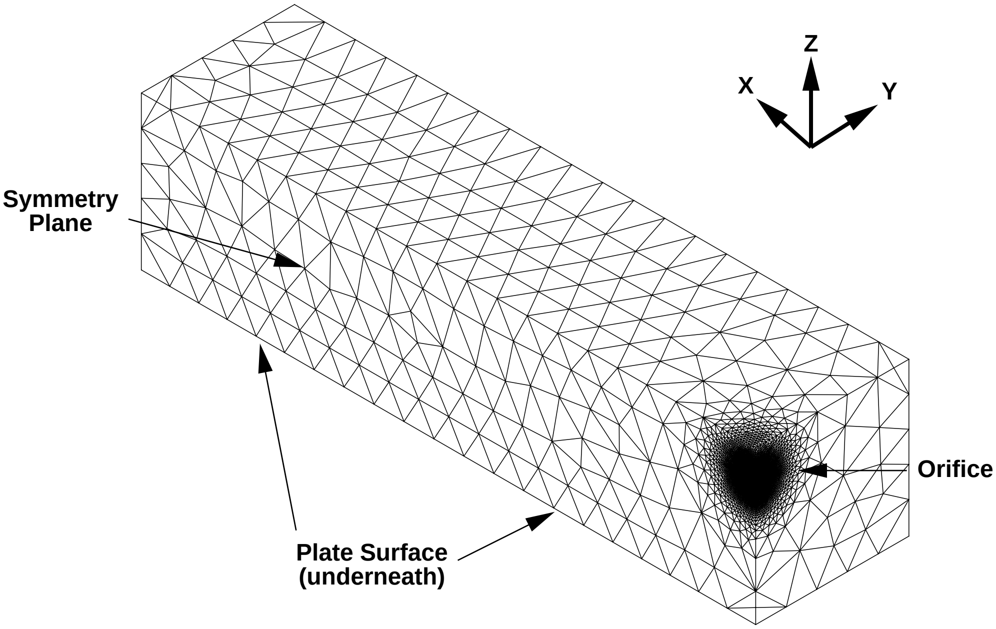

Figure 6.2. Surface grid for $\beta = 0°$ case.

Table 6.1. Computational cost parameters for plate impingement.

Grid Cells	300,000
Number of Particles	3.5 million
Transient Steps	22,500
Sampling Steps	5,000
Calculation Time	14 hours
Steady State η_t	92 %

A hybrid grid would provide some grid flexibility in the vicinity of the inflow. Compression at the surface is sufficiently small that there are no resolution difficulties at the surface. The grid generation package did not allow the use of more complex grid structure.

Variable time scaling is employed using cell size which is taken to be the cube root of the volume. For cells with reasonable aspect ratio this is approximately proportional to the size of the cell.

Simulations are load balanced using a simple scheme similar to those employed for the axisymmetric impingement problems in chapter 4. The "Z" coordinate of a cell's geometric center is used for grouping.

Axisymmetric Calculation

An axisymmetric simulation is made of the normal impingement case ($\beta = 90°$). The grid generation, scaling and load balancing are similar to that used for the axisymmetric plume flows considered in chapter 4.

6.4.3 Computational Cost

Typical cost parameters for the three-dimensional plume simulations are summarized in table 6.1. The calculations are performed using 16 nodes of an IBM SP-2.

6.5 Validation Against Axisymmetric Code

In order to test the accuracy of the three-dimensional code, comparison is made between a three-dimensional simulation and an axisymmetric simulation performed using a well validated version of the DSMC code. The $\beta = 90°$ case is investigated because the plume axis is normal to the plate and the flow is axisymmetric.

When considering an axisymmetric flow, the three-dimensional code should produce the same flow field as the axisymmetric code to within statistical variations. Statistical fluctuations may be larger in the three-dimensional case, particularly away from the axis, because the number of particles per cell is smaller. This is a result of larger volumes of axisymmetric cells which span the entire azimuthal angle.

Data along the plume axis is used to quantitatively compare the three-dimensional and axisymmetric results in the flow field. Figure 6.3 compares densities and velocities; figure 6.4

compares translational temperature. Very good agreement is seen between the two simulations. These results indicate that the two simulations are in fact generating the same flow field.

Of primary importance to this type of problem is the ability to correctly calculate surface properties. A comparison between pressure and shear stress results for the two simulations are shown in figure 6.5. The three-dimensional results show excellent agreement with the axisymmetric both qualitatively and quantitatively. Figure 6.6 shows a similar comparison for heat flux. Good agreement is again found between the simulations. The three-dimensional results show a larger amount of statistical scatter due to smaller sample sizes. These comparisons indicate that the two codes produce essentially the same surface properties for an axisymmetric flow.

6.6 Comparison of Surface Properties

The experimental data from DLR are used to verify the accuracy of the plate simulations. Measurements of plume flow field properties are not available for this problem and scarce for any three-dimensional problem. Accurate prediction of surface properties should indicate that the plume is being modeled correctly.

Free molecular analysis is used to calculate the surface properties for each of the plate orientations considered. Comparison with the simulation results provides a estimate of the accuracy of this theory for simple impingement flows of this type. The theory also provides a qualitative basis for comparison with the simulation at the far ends of the surface where experimental data are not available.

In the case of normal impingement of the plume ($\beta = 90°$), the experimental data are taken along a radial line extending out from the axis. The simulation data set includes data across the entire surface presented as a function of distance from the axis. In the other cases both the experimental and simulation data sets are taken along a line on the surface which is coplanar with the plume axis.

6.6.1 Normal Impingement ($\beta = 90°$)

Figure 6.7 compares normalized surface pressures for simulation, experiment and free molecular theory. Good agreement is found between the data sets. The profiles show the same general shape, pressure falling from a maximum at the axis. The DSMC and theoretical profiles show an asymptote at the background pressure at large distances from the axis, where the plume has minimal effect on the surface.

The simulation under predicts the experimental values by a significant amount (as much as 60%) at the far extent of the experimental data set. The smooth asymptotic shape seen in the DSMC and theoretical profiles is not as apparent in the measurements far from the axis. The slower decrease in pressure with distance from the axis may indicate difficulties in measuring pressures at the low densities away from the axis. It may also indicate a higher tank pressure in the facility than reported.

Normalized shear stress is considered in figure 6.8. Good agreement is observed between the simulation, experiment and theory. The profiles each show the same shape with stress

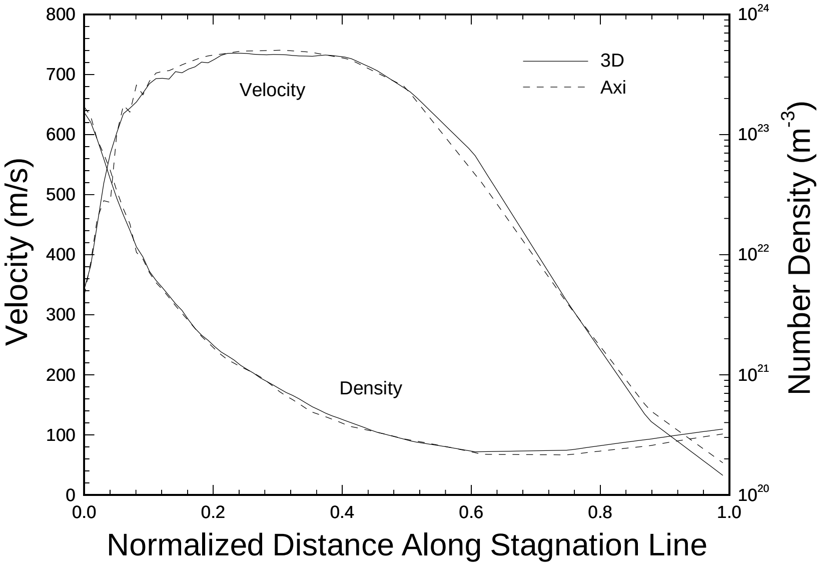

Figure 6.3. Comparison of velocities and densities along stagnation line for three-dimensional and axisymmetric simulations of normal impingement case.

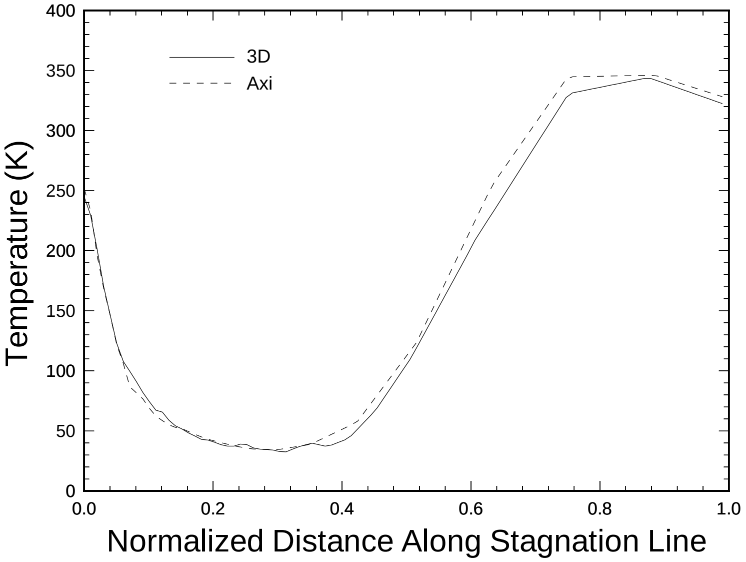

Figure 6.4. Comparison of temperature along stagnation line for three-dimensional and axisymmetric simulations of normal impingement case.

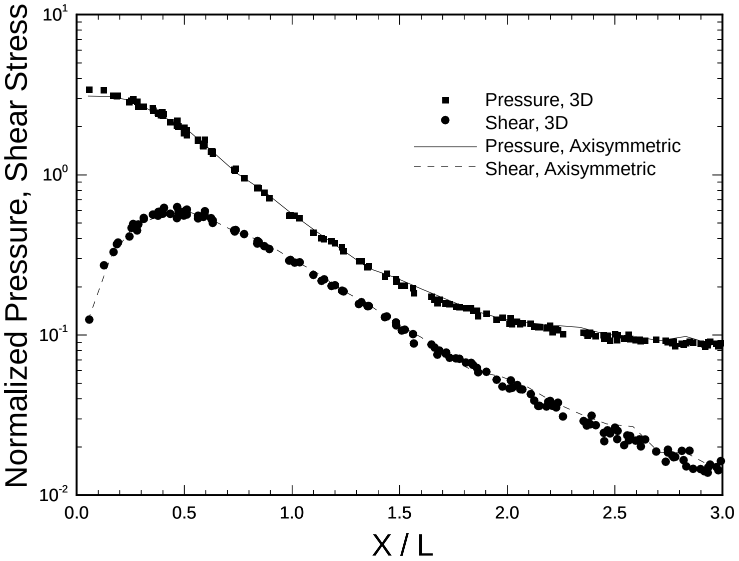

Figure 6.5. Comparison of surface pressure and shear stress for three-dimensional and axisymmetric simulations of normal impingement case.

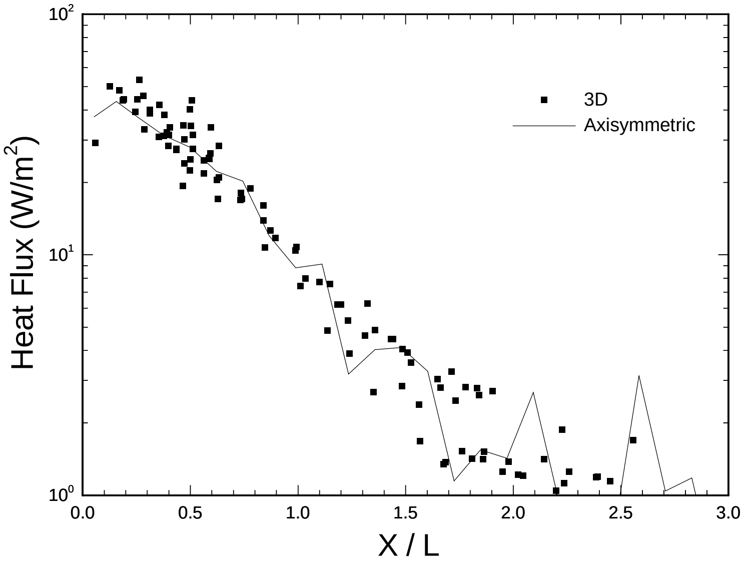

Figure 6.6. Comparison of surface heat flux for three-dimensional and axisymmetric simulations of normal impingement case.

rising from zero at the plume axis to a maximum and then falling with decreasing gas pressure. DSMC stresses exceed the free molecular values far from the axis. This is likely a result of particles which strike the surface, undergo collisions with the incoming plume and are scattered back to the surface at a high angle of attack.

Surface heat flux is shown in figure 6.9. Significant fluctuations appear in both the simulation and experimental data sets, particularly at large distances from the axis. The statistical fluctuations in the DSMC simulation are magnified in the calculation of energy transfer which involves the second moment of the particle velocities. In the experiment, the fluctuations result from the difficulty of measuring heat flux in extremely rarefied flows. There is good agreement between the two data sets despite the fluctuations. The simulation does over predict the heat flux at the axis significantly, however.

The theoretical curve has the same shape as the DSMC result but shows a larger magnitude of heat transfer. The difference increases as the axis is approached. Gas densities are larger near the axis and consequently the mean free path and Knudsen number are smaller. The assumption of free molecular flow is weakest near the axis and collisional effects are more significant. This tends to lower the energy transfer below the free molecular value. The deviation is greater in terms of heat transfer than in pressure or shear stress since energy is a second moment of the particle distribution function.

6.6.2 $\beta = 45°$ Case

Figure 6.10 shows surface pressure profiles for the $\beta = 45°$ case. Good agreement is again found between the simulation and experimental data sets as well as free molecular theory. The point of maximum pressure is just downstream of the point directly below the orifice where the effects of density and angle of attack are high. The pressure drops off asymptotically to the background value to both sides of this maximum, more rapidly behind the orifice.

The experimental data are again higher than the DSMC and theoretical values at both ends of the profile. The magnitude of the pressures at these points are comparable to that seen at the end of the $\beta = 90°$ profile. This points toward uncertainty in pressure measurements at low densities or a higher back pressure in the facility.

Shear stress profiles are reported in figure 6.11. The DSMC stresses show excellent agreement with the experimental data. The features of the experimental profile are captured by the simulation, particularly the rapid decrease which occurs directly below the orifice.

While the qualitative structure of the free molecular profile agrees with those of the simulation and experiment, the theoretical profile differs in terms of magnitude. In front of the orifice ($X/L > 0$), the free molecular stresses over predict the simulation and measured values. Behind the orifice ($X/L < 0$), the free molecular stresses under predict the actual stresses. This is likely due to the presence of backscattered particles which strike the surface behind the orifice.

Heat transfer profiles are reported in figure 6.12. Although significant fluctuations are again seen in the simulation and experimental results there is generally good agreement between the data sets. Free molecular theory over predicts heat transfer across most of the range considered due to collisional effects. This effect is countered to some extent behind the orifice by backscattered particles which tend to increase the transport of energy to the

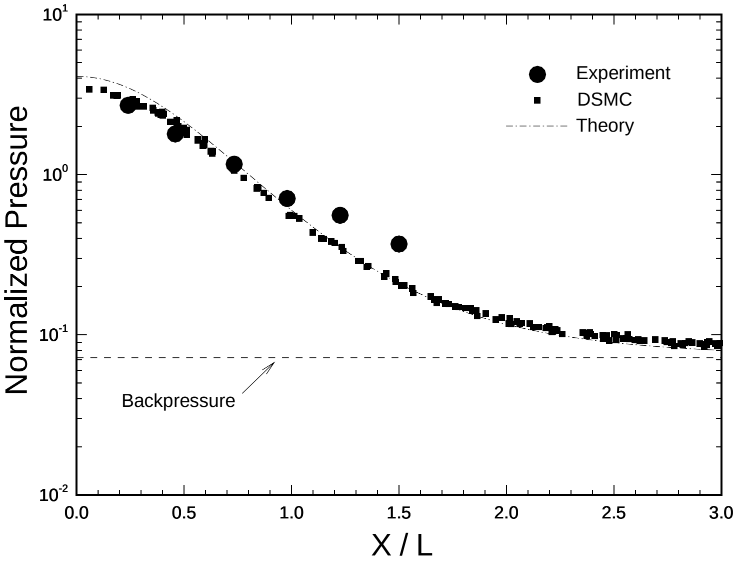

Figure 6.7. Comparison between DSMC and measured surface pressures at $\beta = 90°$.

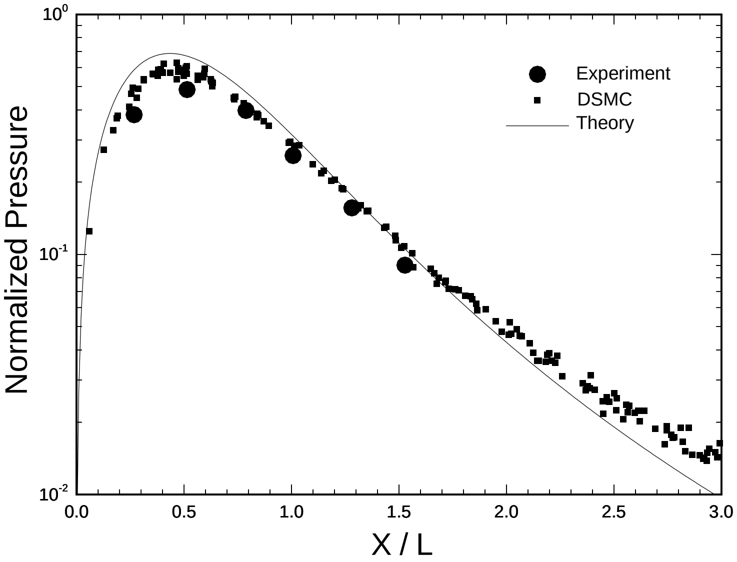

Figure 6.8. Surface shear stress for $\beta = 90°$. Comparison between DSMC, experiment and free molecular theory.

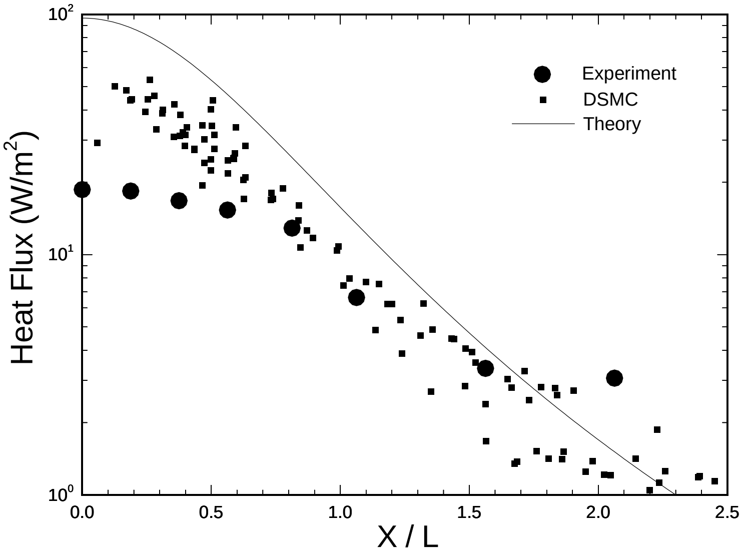

Figure 6.9. Surface heat flux for $\beta = 90°$. Comparison between DSMC, experiment and free molecular theory.

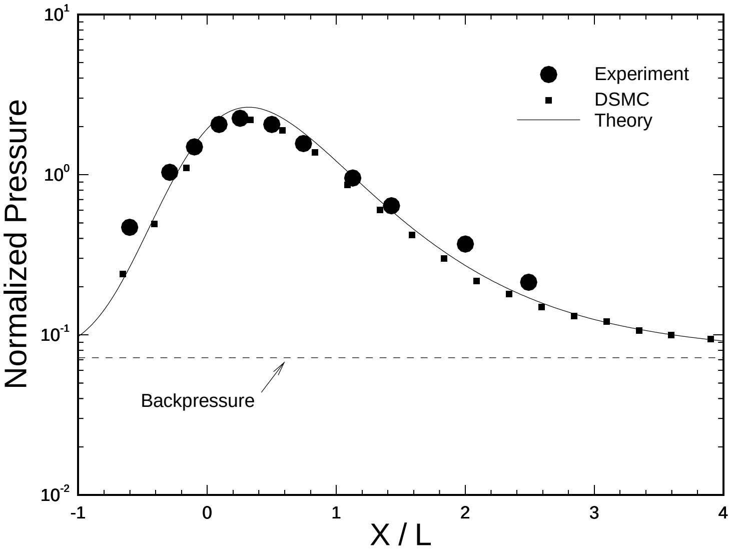

Figure 6.10. Surface pressure for $\beta = 45°$. Comparison between DSMC, experiment and free molecular theory.

surface.

6.6.3 Parallel Impingement ($\beta = 0°$)

Before presenting comparisons with experimental data it is important to note that the DLR study by Legge does not report pressure or shear stress data for the parallel impingement case with a stagnation pressure of 1000 Pa. The lowest stagnation pressure for which $\beta = 0°$ pressure and shear data are reported is 4000 Pa. The study by Döring does present heat flux data for a stagnation pressure of 1000 Pa. The DSMC simulations use a stagnation pressure of 1000 Pa. Although the data are normalized by the stagnation pressure, Legge reports some effect of flow rarefaction on the normalized surface properties, particularly shear stress.

The difference in stagnation pressures complicates the modeling of tank pressure effects. The ratio of background to stagnation pressure is not a constant in the experimental facility, generally falling with increasing stagnation pressure. Consequently, the parallel impingement case has been simulated without backpressure and the theoretical pressure profile is for expansion into a vacuum.

Surface pressures for the parallel impingement case are shown in figure 6.13. The profiles show good qualitative agreement. Without a backpressure the simulation and theoretical profiles drop toward zero at the ends of the plate. No asymptote is seen in the experimental data because the normalized facility pressure (not shown) is significantly lower with the higher stagnation pressure. The lower experimental pressures are a result of a higher stagnation pressure and lower Knudsen number at the surface.

Shear stress data are shown in figure 6.14. The profiles again show qualitative agreement. The experimental values are lower due to higher densities. The effect on shear is stronger than that on pressure. This was previously reported by Legge. The effect of rarefaction can be seen to a much smaller degree in a comparison between simulation and free molecular results. The DSMC values are slightly lower due to collisional effects.

Heat transfer results are presented in figure 6.15. Experimental heat flux data are based on a stagnation pressure of 1000 Pa, equal to that used in the DSMC simulation. Statistical fluctuations are quite large at the low densities involved in the parallel impingement case. However, reasonably good agreement is seen between the experimental and simulation data. Surprisingly, the free molecular results do not significantly over predict the heat transfer as was seen in the other cases.

6.7 Conclusions

The DSMC method provides reasonably accurate modeling of the simple plate impingement flow. Good agreement with experimental data is found for different plate orientations. The results of this work indicate that the plume and impingement effects are modeled correctly.

A comparison of axisymmetric and three-dimensional simulations of the normal impingement case result in effectively identical results both in the flow field and at the surface. This verifies that the three-dimensional version of MONACO maintains the accuracy of the previously tested two-dimensional version.

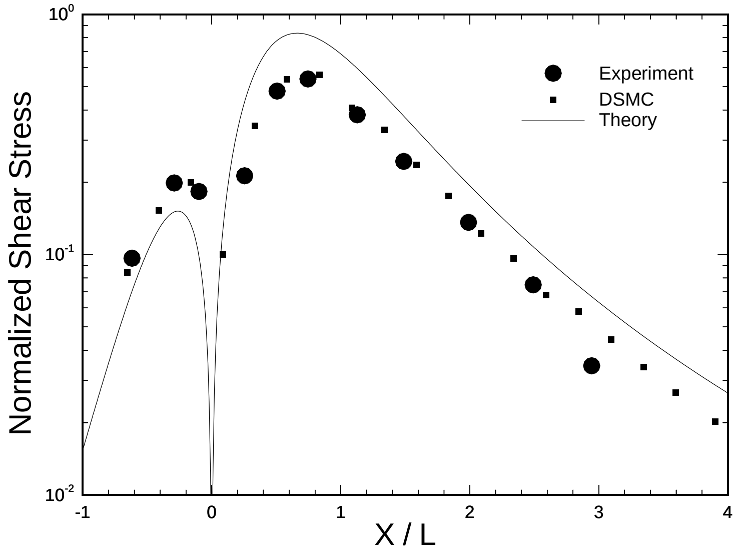

Figure 6.11. Surface shear stress for $\beta = 45°$. Comparison between DSMC, experiment and free molecular theory.

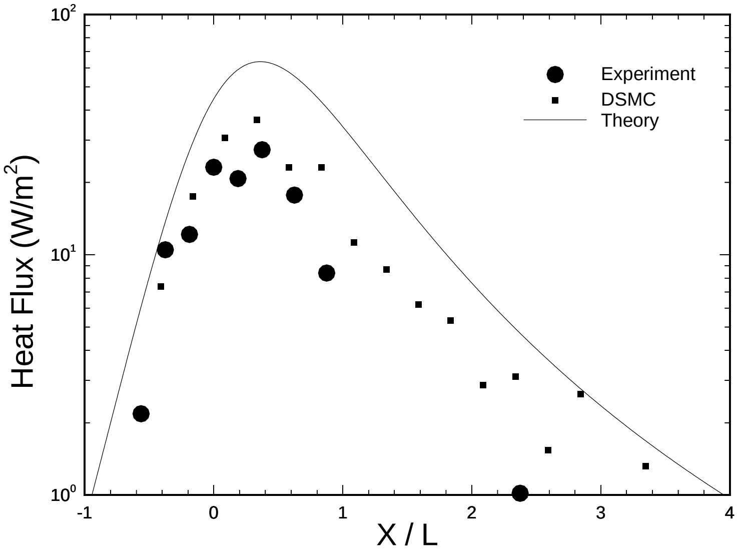

Figure 6.12. Surface heat flux for $\beta = 45°$. Comparison between DSMC, experiment and free molecular theory.

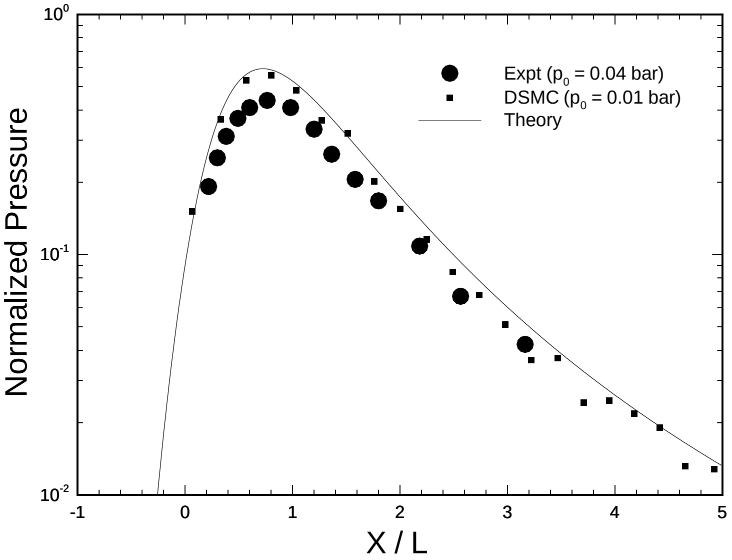

Figure 6.13. Surface pressure for $\beta = 0°$. Comparison between DSMC, experiment and free molecular theory.

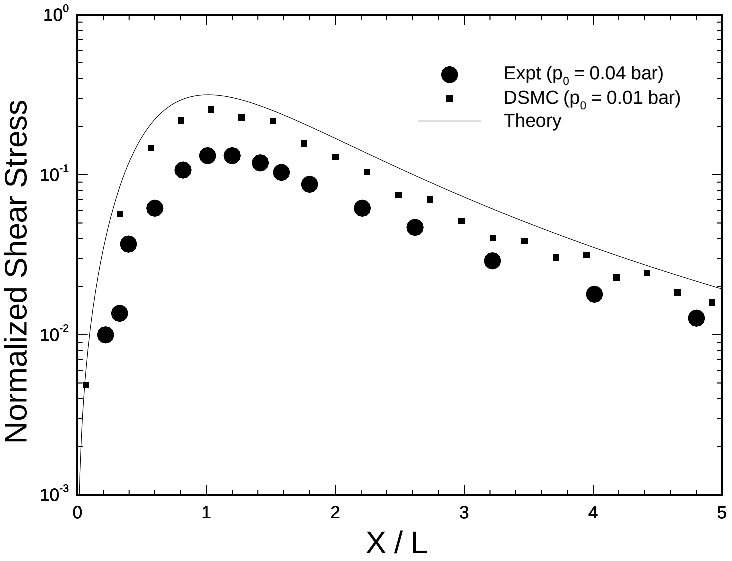

Figure 6.14. Surface shear stress for $\beta = 0°$. Comparison between DSMC, experiment and free molecular theory.

Free molecular theory provides a reasonable estimate of surface properties at high Knudsen numbers. The analysis tends to over predict the values, however, with heat transfer and shear stress being more sensitive to the degree of rarefaction than pressure.

Flows with more complex geometries may not readily lend themselves to a simple free molecular analysis. Surfaces which are not in direct line of sight from the source will not be modeled correctly. In particular, flow behind the orifice was not predicted very accurately due to back scattering particles.

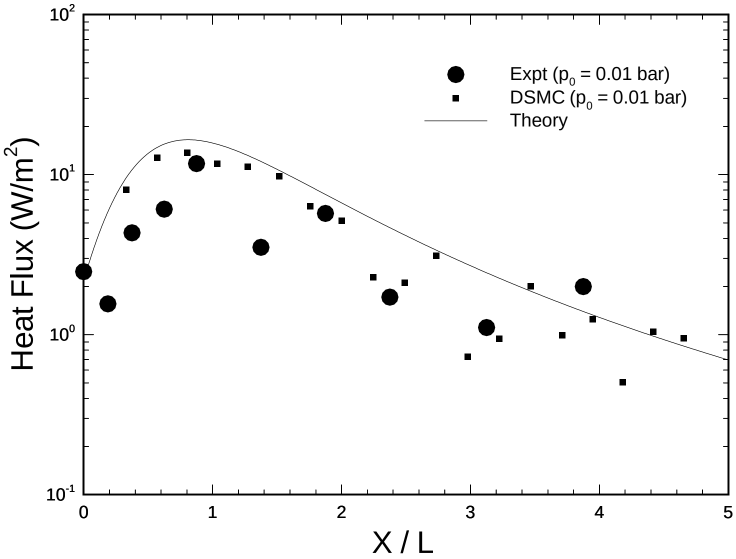

Figure 6.15. Surface heat flux for $\beta = 0^\circ$. Comparison between DSMC, experiment and free molecular theory.

Chapter 7

Demonstration Results

The goal of this work is to develop the capability to simulate plume flows around complex satellite geometries. The ability to calculate simple impingement flows using DSMC has been demonstrated in the preceding chapter of this thesis. It is now necessary to apply the current code to a more realistic configuration.

A representative satellite geometry consisting of a spacecraft bus and solar arrays is used as a model configuration. A plume is generated by a hydrazine control thruster mounted at the corner of one side of the spacecraft bus. Figure 7.1 shows a schematic of the problem. The spacecraft geometry is a 1.5 m cube of which half is simulated. The array section is 3.25 m long, 2.5 m wide and 0.12 m thick and is deployed 0.75 m from the spacecraft. Expansion of the plume and its impingement on the solar array panel are modeled. Two orientations of the array with respect to the bus are considered.

The thruster is modeled after an operating hydrazine control thruster, the MR-103-series 0.2 lbf REA manufactured by Primex Aerospace Company. This thruster was originally developed for the Voyager 1 and 2 spacecraft and is also in use on such programs as GPS, Iridium, Cassini, and various communications satellites. This study will consider an operating mode at 60% of the rated thrust. Table 7.1 shows relevant data for the thruster [12].

7.1 Physical Modeling

The gas plume is composed of a mixture of nitrogen, hydrogen and ammonia molecules. Simulations begin at the exit plane of thruster using flow properties based on a numerical calculation of flow in the thruster [12]. The inflow profiles are constructed with piecewise linear fits of the exit plane data. Figures 7.2 and 7.3 show profiles of exit plane properties. Number densities in the exit plane are scaled down by a factor of 1.52 in order to insure consistency with the measured mass flow of the device.

Due to the relatively low temperature of the plume it can be considered chemically frozen. Vibrational energy modes are likewise frozen.

All spacecraft surfaces are modeled as diffuse reflectors with full accommodation. Surfaces are maintained at 273 K. While this temperature is relatively high for a spacecraft operating in the near vacuum of space it is not unreasonable due to the heat dissipation of spacecraft electrical components.

Surface chemistry and adsorption are not considered in these calculations. The intent is to isolate the fluid mechanics of the plume impingement problem. More advanced surface models and effects can be included at a future time.

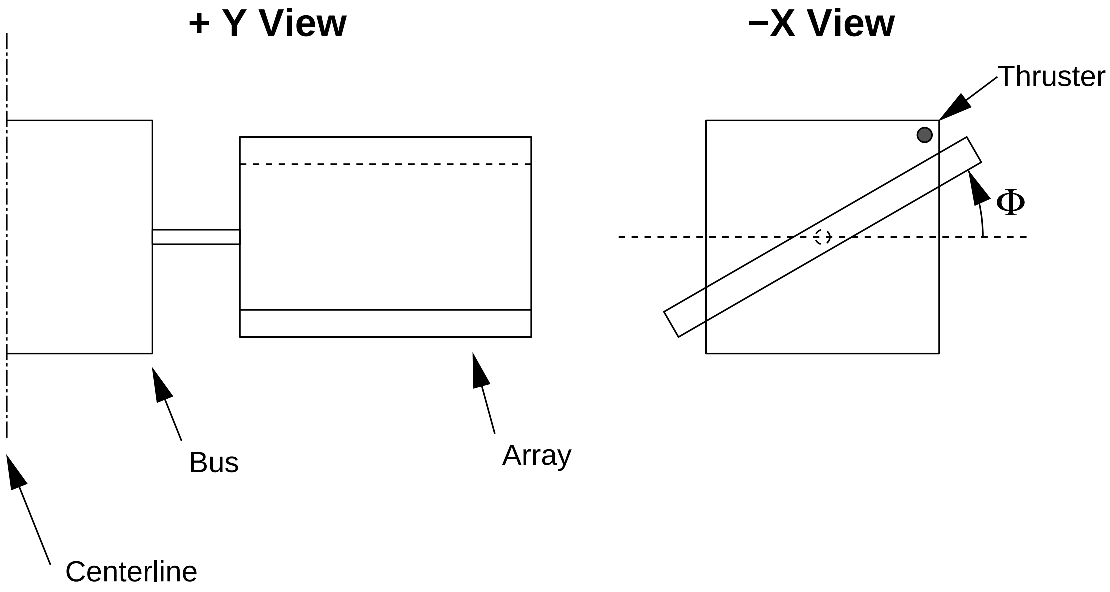

Figure 7.1. Schematic of model satellite configuration.

Table 7.1. Operating conditions and parameters for hydrazine thruster.

Nominal Thrust	0.55 N
Expansion Ratio	100:1
Half Angle	15°
Exit radius	2.921 mm
Chamber Pressure	1.25 MPa
Stagnation Temperature	1167 K
Flow rate	2.62×10^{-4} kg/sec

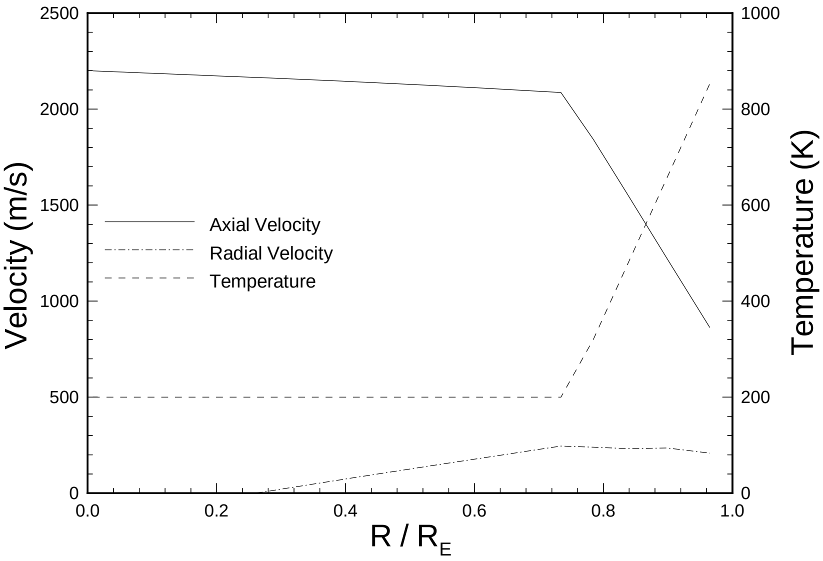

Figure 7.2. Hydrazine thruster exit plane velocity and temperature.

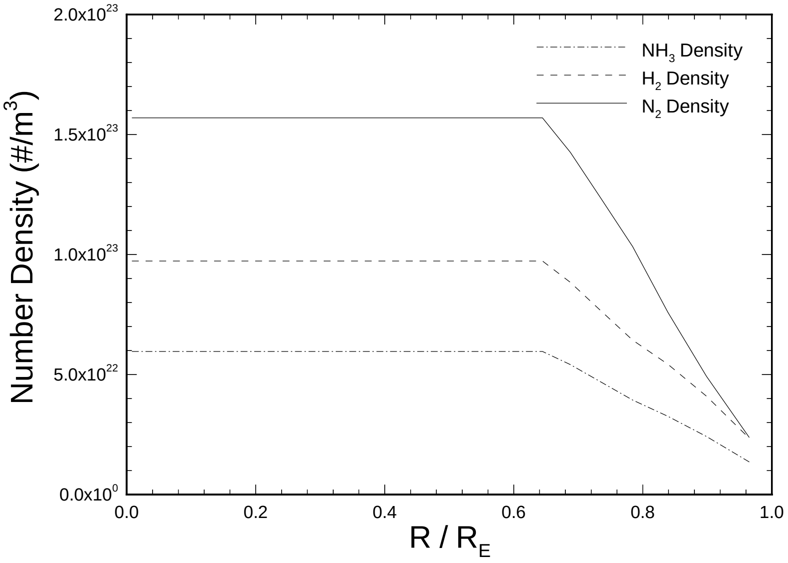

Figure 7.3. Hydrazine thruster exit plane number densities.

7.2 Computational Modeling

Simulations are performed in two parts. First, an axisymmetric calculation of the near field plume in the vicinity of the thruster is performed. Data from this calculation is then used as input for the three-dimensional calculation of the plume far field and impingement. Separation of the calculation in this manner greatly reduces the cost of the overall calculation. The number of cells required for the three-dimensional plume simulation is significantly reduced when the high density region near the nozzle exit is not included. An axisymmetric simulation of this region requires far fewer cells and particles and thus less computational expense. Additionally, different orientations of the array can be investigated without recalculating the vicinity of the nozzle.

7.2.1 Near Field

The near field expansion of the plume is considered to be an axisymmetric problem which is independent of the surrounding geometry. The flow domain is hemispherical and extends more than ten exit radii out from the center of the thruster exit. The grid is an unstructured mesh with cell sizes and time steps scaled according to the local mean free path.

Inflow conditions for the far field calculation are taken at the breakdown surface. This is the locus of points for which the breakdown parameter (see eq, 4.1 and Ref. [4]) equals the threshold value $P = 0.02$. In order to more readily use this as a geometric interface between the near and far field simulations the surface is simplified to a capped cylinder with uniform cross section. Density in the near field is shown in figure 7.4 along with the location of the interface surface.

7.2.2 Far Field and Impingement

The far field simulation considers the whole area surrounding the spacecraft, including the solar array. The computational domain is an orthogonal parallelepiped which extends beyond the spacecraft surface by at least 0.25 m in each coordinate direction. Only one half of the spacecraft is simulated. The boundaries of the computational domain are shown in figure 7.5. The solar array is shown in the nominal, unrotated configuration ($\Phi = 0°$).

An unstructured mesh consisting of tetrahedral cells is employed. The mesh is generated using the grid generation package FELISA. Cell sizes are roughly scaled according to local density. The ability to adapt the grid directly to the local properties was not available, thus there is some variation of cell sizes. Variable time steps are employed with cell size being used as a scaling factor.

It has previously been noted that the number of particles per cell scales as the inverse of the square of the density in three-dimensional flows (see section 3.1.2). Scaling of time steps according to cell size was shown to reduce the dependence on density by one power. Thus, a three-dimensional problem with variable time steps will see particle counts inversely proportional to the density. In an expanding flow such as the model plume impingement flow this can cause significant inefficiencies due to the large range of densities involved

In order to counteract the remaining scaling with density, a simple scheme of particle weight scaling is employed. Cells in the immediate vicinity of the inflow (interface) surface

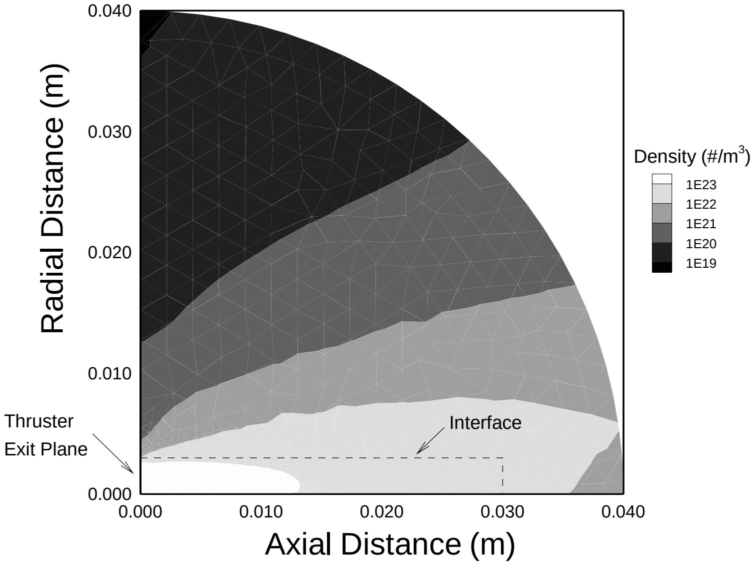

Figure 7.4. Contours of number density for the axisymmetric near field simulation. The interface surface is indicated.

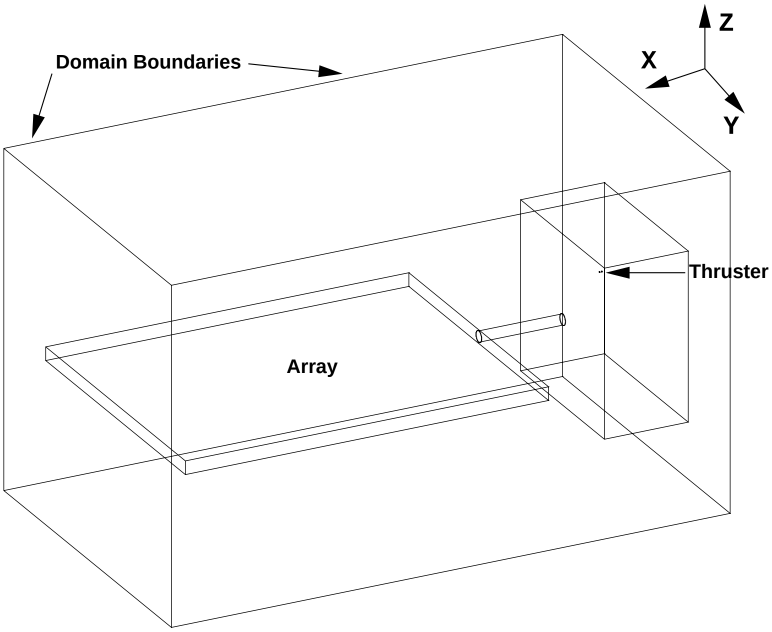

Figure 7.5. Boundaries of the computational domain. Shown for unrotated array case, $\Phi = 0°$.

are assigned a low particle weight, 10% of the reference value for the simulation. The band of cells immediately outside this surface are assigned a weight of 40% of the reference value. The remaining cells are assigned the reference weight. Figure 7.6 shows this weight scheme on a plane cut taken along the axis of the cylindrical inflow surface. This scaling roughly follows the density variation in close proximity to the inflow. Since there is only a small number of particles moving toward the inflow in this highly supersonic flow, there is a minimal amount of cloning involved. The weight scheme allows reasonable resolution of the high density region without affecting the accuracy of the calculation.

7.3 Computational Cost

The computational cost and related parameters for the near and far field plume calculations are summarized in table 7.2. The expense of the three-dimensional calculation is primarily due to the large number of small cells in the vicinity of the interface surface. For instance, a hemispherical interface with radius 33% larger than the cylinder height, the number of particles required would drop by an order of magnitude. This would require a more advanced boundary condition which would include nonequilibrium distribution functions at the interface.

7.4 Results

Impingement simulations were performed for two orientations of the solar array, $\Phi = 0°$ and $\Phi = +30°$. The flow fields for these calculations are shown in figures 7.7 and 7.8. Each figure shows a plane cut through the domain which is parallel to the beam axis and contains the plume axis. Figure 7.7 shows number density contours for the $\Phi = 0°$ case and figure 7.8 shows velocity contours for the $\Phi = +30°$ case. Both cases show the expected rapid expansion of the plume gas out of the nozzle and subsequent compression at the surface of the array.

Of primary importance are the mass, momentum and energy imparted to the array by the plume. Figures 7.9 and 7.10 show distributions of surface pressure for the 0°and 30°cases. Figures 7.11 and 7.12 show contours of heat transfer to the array. The view is in the plane of the array in both cases. Both sets of contour plots show maximum values at the near end of the array toward the +Y side, closest to the thruster. In the $\Phi = 0°$ case the maximum pressure and heat transfer are downstream of the leading edge due to the influence of the nozzle boundary layer. As expected the impingement properties show larger values in the $\Phi = +30°$ case where the array is tilted toward the thruster.

Ammonia molecules are the major contamination concern with hydrazine thrusters. Figures 7.13 and 7.14 show distributions of ammonia flux on the array. The qualitative behavior of this property is very similar to that observed with momentum and energy transfer.

Data extracted along a line on the array surface is used to quantitatively examine the surface properties. Data is taken on the line which intersects the array top surface and the plane parallel to the array axis which contains the plume axis. This plane is the one used to display flow field contours.

Comparisons of the DSMC results are made with predictions from free molecular theory. Since the thruster nozzle has a significant boundary layer, the plume density model needs to

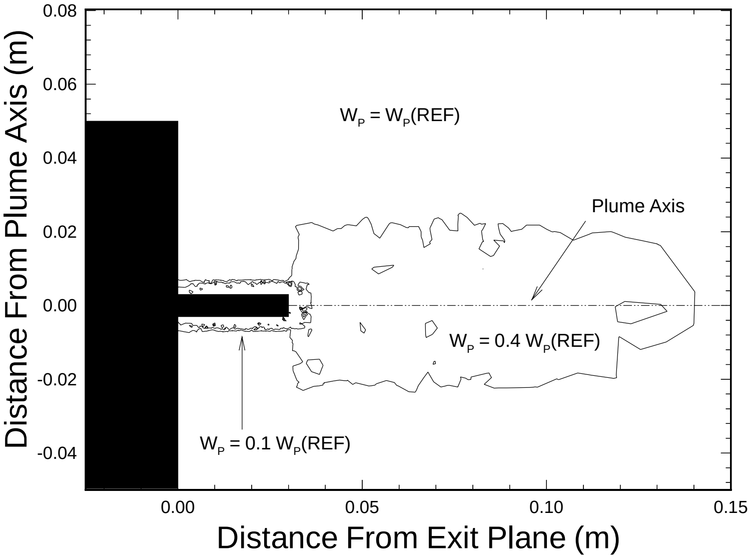

Figure 7.6. Particle weight scaling in the vicinity of the inflow surface.

Table 7.2. Computational cost parameters for model satellite configuration.

	Near Field	Far Field
Grid Size	7,600 cells	290,000 cells
Number of Particles	425,000	2,900,000
Transient Steps	12,500	18,000
Sampling Steps	10,000	10,000
Calculation Time	5 hours	11 hours
Number of Processors	4	16
Parallel Efficiency	97%	91%

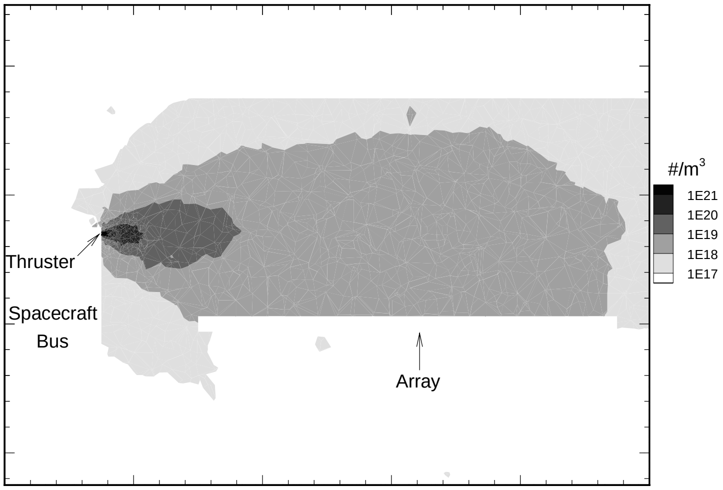

Figure 7.7. Flow field contours of number density for 0°array case. Data on a plane slice parallel to the plume axis.

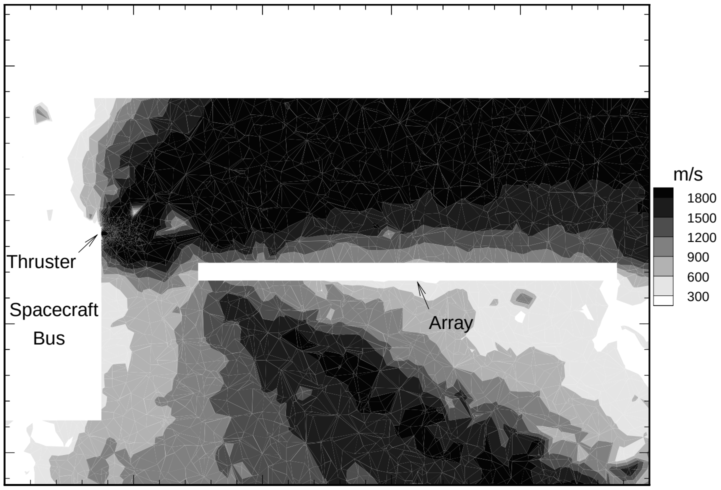

Figure 7.8. Contours of velocity for 30°array case. Data on a plane slice through the plume axis.

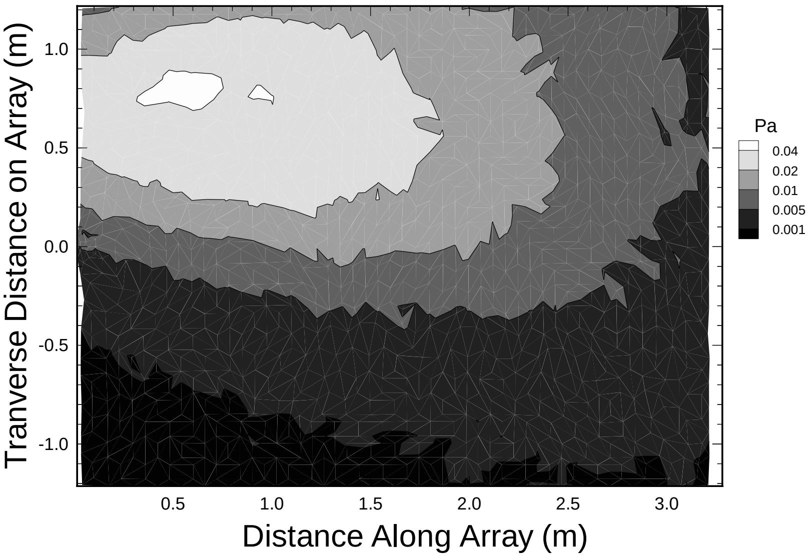

Figure 7.9. Contours of impingement pressure at the array surface for the 0°array case.

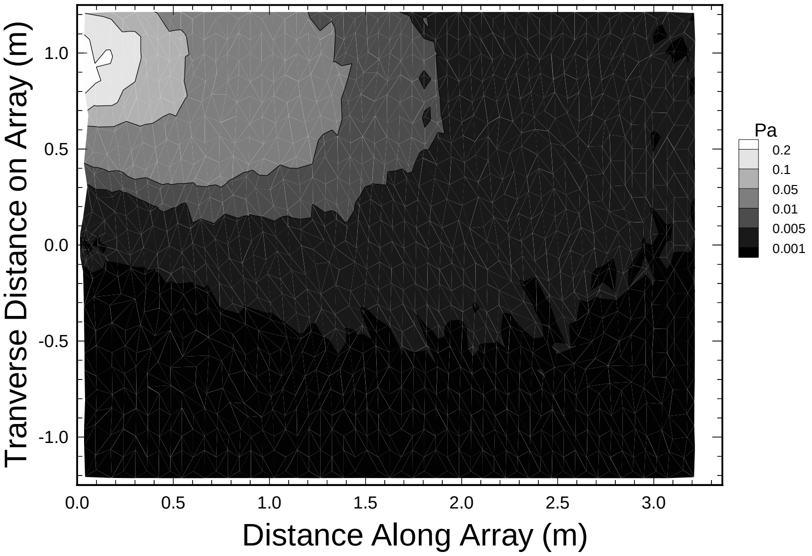

Figure 7.10. Contours of impingement pressure at the array surface for the 30°array case.

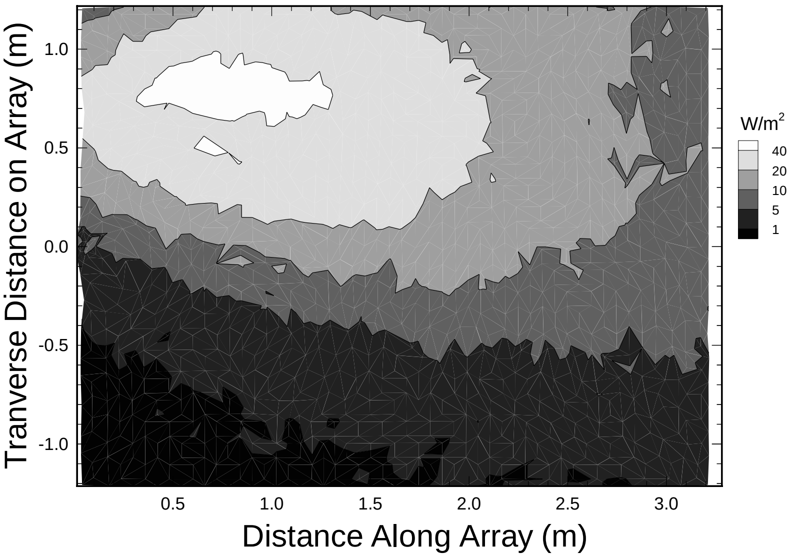

Figure 7.11. Contours of heat flux at the array surface for the 0°array case.

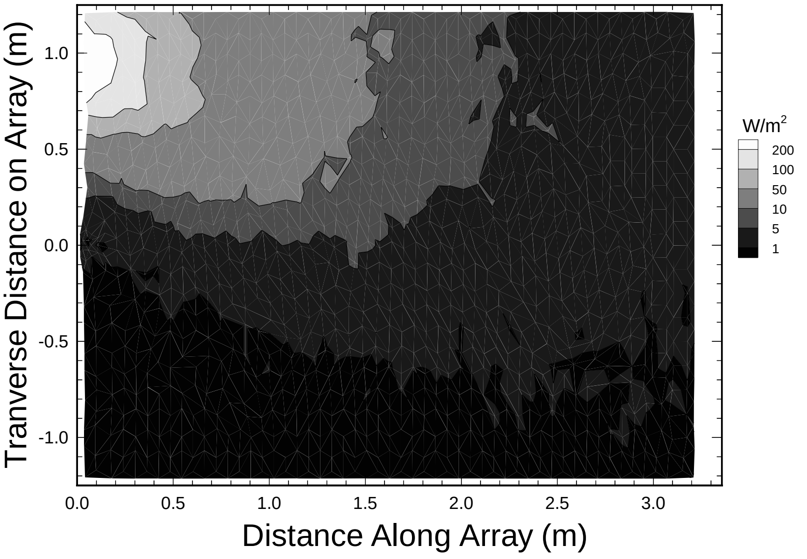

Figure 7.12. Contours of heat flux at the array surface for the 30°array case.

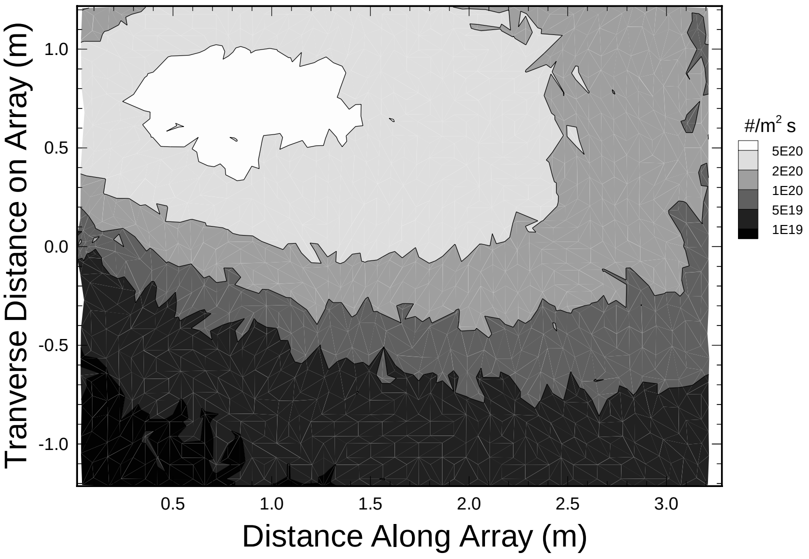

Figure 7.13. Contours of NH_3 flux at the array surface for the 0°array case.

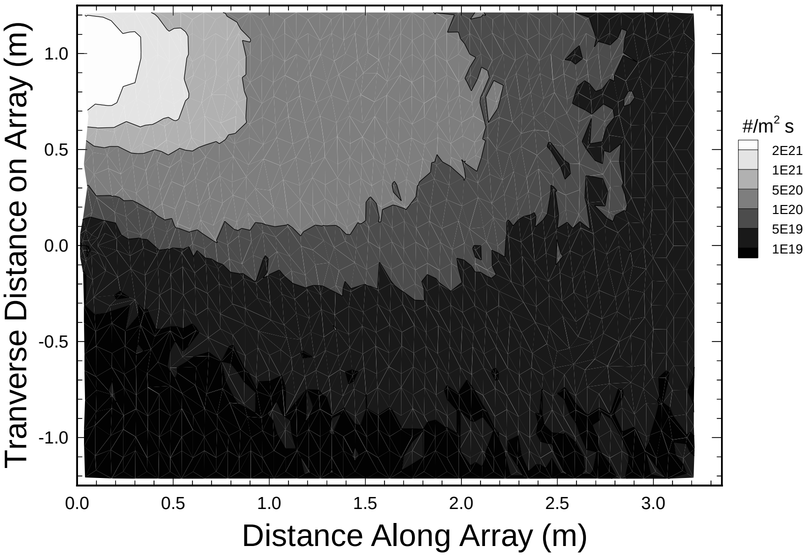

Figure 7.14. Contours of NH_3 flux at the array surface for the 30°array case.

take this into account. The angular distribution of density in the Simons plume model (eq. 6.6) has the following form when boundary layer effects are included:

$$\begin{aligned} f(\theta) &= \left[\cos\left(\frac{\pi}{2}\frac{\theta}{\theta_{lim}}\right)\right]^{2/(\gamma-1)}, && \theta \leq \theta_0 \qquad (7.1) \\ &= f(\theta=\theta_0)\exp\left[-C(\theta-\theta_0)\right], && \theta_0 < \theta < \theta_{lim} \end{aligned}$$

θ_0 is the angle between the plume axis and the streamline separating the boundary layer from the isentropic core. The constant C is given by:

$$C = \frac{3}{2}A_P\sqrt{\frac{\gamma+1}{\gamma-1}}\left(\frac{R_E}{2\delta_E}\right)^{\frac{\gamma-1}{\gamma+1}} \qquad (7.2)$$

where R_E is the exit radius of the nozzle and δ_E is the boundary layer thickness at the exit.

Number density flux to the surface is in the hypersonic, free molecular limit given by:

$$\begin{aligned} F &= nU \qquad (7.3) \\ &= \frac{p_0}{m}A_P f(\theta)\left(\frac{r^\star}{r}\right)^2\left(\frac{2}{\gamma+1}\right)^{\frac{1}{\gamma-1}}\sqrt{\frac{2}{RT_0}\frac{\gamma}{\gamma-1}} \end{aligned}$$

The operating conditions for the hydrazine thruster can be used to calculate the plume parameters, A_P, θ_{lim}, θ_0 and C. There is some uncertainty involved in this calculation when a multi-species gas is considered. To be most useful from an engineering standpoint, the free molecular results should be based on stagnation conditions and the geometry of the nozzle. The expressions for surface properties (equations 6.7–6.11 and 7.3) depend on gas properties R and γ which are difficult to determine a priori for a rarefied gas in which individual species will have varying mole fractions and velocities. In this study, the gas has been assumed to have a constant molecular weight and ratio of specific heats using values reported for the exit plane ($M = 13$ kg/mol, $\gamma = 1.4$). These parameters yield a plume constant, $A_P = 3.62$, boundary layer constant, $C = 12.6$, and limiting angle, $\theta_0 = 19.2°$ and $\theta_{lim} = 58.7°$.

Figure 7.15 shows impingement pressure profiles for the 0°case. Reasonable agreement is found on the far end of the array but at the near end the free molecular result falls well below the simulation. The noticeable drop in the free molecular profile on the upstream side of the array is a result of boundary layer effects. The drop in pressure can be seen in the simulation as well but is not as dramatic. The sharp decline indicates that the density model falls too rapidly in the boundary layer regime.

The 30°case is shown in figure 7.16. In this case the array is rotated towards the thruster and as a result the profile being considered is significantly closer to the thruster. The linear profile is within the core of the plume and consequently there is no drop in pressure on the upstream side of the plate as a result of the boundary layer. Both the simulation and analytical profiles primarily show an inverse squared drop in pressure with distance which is consistent with the density model. The DSMC results predict a 70% higher pressure across the length of the array. This may indicate a low estimate of the plume constant, A_P.

Figures 7.17 and 7.18 show heat transfer profiles. In both cases the behavior is very similar to that of surface pressure. The free molecular profile for 0°shows an extreme drop

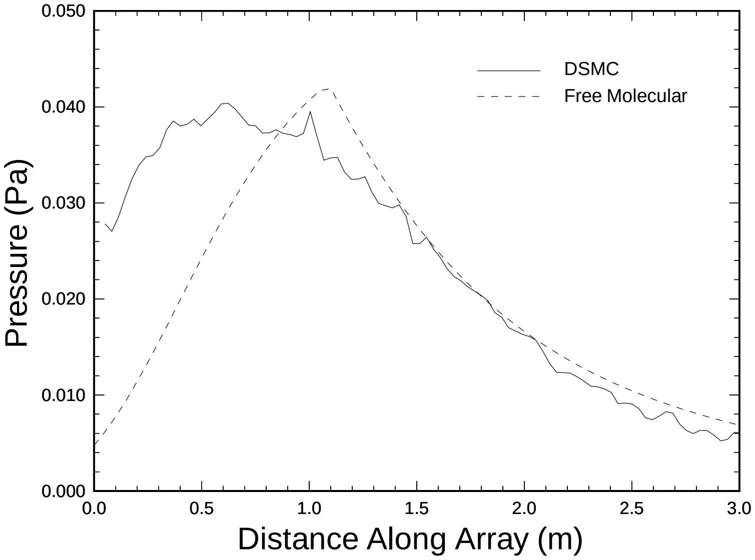

Figure 7.15. Comparison of simulation and free molecular pressures on array surface for 0°array case.

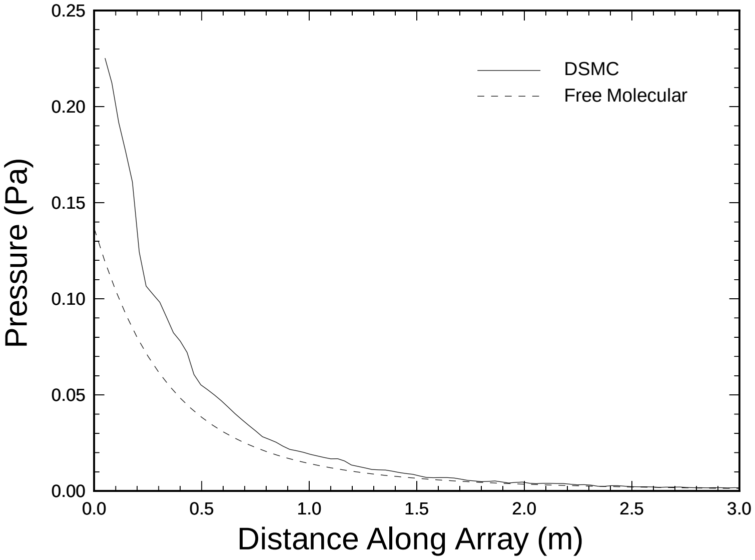

Figure 7.16. Comparison of simulation and free molecular pressures on array surface for 30°array case.

on the upstream end while otherwise matching the shape of the DSMC profile. In the 30°case the DSMC and theoretical profiles are qualitatively similar. In both cases the free molecular values tend to exceed the simulation heat transfer. Over prediction of heat flux is consistent with the results for flat plate impingement.

The number density flux of ammonia striking the array is shown in figures 7.19 and 7.20. Two free molecular profiles are shown for each case. The first (labeled A) assumes a constant species composition in the plume model and a uniform limiting velocity based on the average molecular weight. The second (B) calculates the ammonia flux as if the plume was composed entirely of ammonia with the correct number density. The two methods produce essentially the same fluxes.

Qualitatively the flux profiles show the same behavior for both pressure and heat transfer. Both angle cases show significantly higher DSMC fluxes. This indicates the difficulty of applying the free molecular model to a multi-species, rarefied gas.

The problems with the free molecular model for flux are illustrated by a simulation of the thruster plume flow field without any impingement surfaces. This plume flow should give a reasonable estimate of what the impingement surfaces see, since the boundary layer caused by the impingement is small. Figure 7.21 plots ammonia density for the freely expanding plume at the same spatial location as the linear profiles shown for the 0°case in figure 7.19. The DSMC results are compared with density predicted by the Simons plume model (assuming an ammonia plume — method B above).

Figure 7.22 compares simulated ammonia velocities and the limiting velocity employed in the free molecular analysis ($\sqrt{2C_pT_0}$). Downstream of where the leading edge of the solar array would be located, within the isentropic core of the plume, DSMC densities are on average two times higher than the Simons model predicts. DSMC velocities are also 5–8% larger than the analytical model. Since the number density flux in free molecular flow is the product of density and velocity (eq. 7.3), these factors combine to produce the significantly larger fluxes seen in the DSMC results shown in figure 7.19.

In the boundary layer region, the relationship between plume properties and impingement flux is less clear. However, near the upstream edge, the simulation again produces significantly higher ammonia densities which again would lead to larger flux. The more complex density profile seen in the DSMC result, likely due to differing mole fractions and velocity slip between the species again indicates the limitations of the analytical plume model for complex gas flows.

The total forces and torques imparted to the array are important for spacecraft design. The total amount of a contaminant such as ammonia striking the surface is also important. Tables 7.3 and 7.4 summarize these integrated properties for the two simulations. Integrated properties calculated from free molecular theory are also included.

It is worthy to note that in both cases the net force exerted on the array is on the order of 20% of the nominal 0.55 N thrust of the thruster. This indicates that a significant proportion of the propulsion energy is wasted. There is also little net difference between the two array orientations. Although part of the array is closer to the thruster in the 30°case, the average angle of attack over the array is higher which reduces the impingement properties. The rotation causes a portion of the array to be significantly farther from the thruster, reducing the effective area over which force is exerted.

Comparison between the simulation and free molecular values follows the same pattern

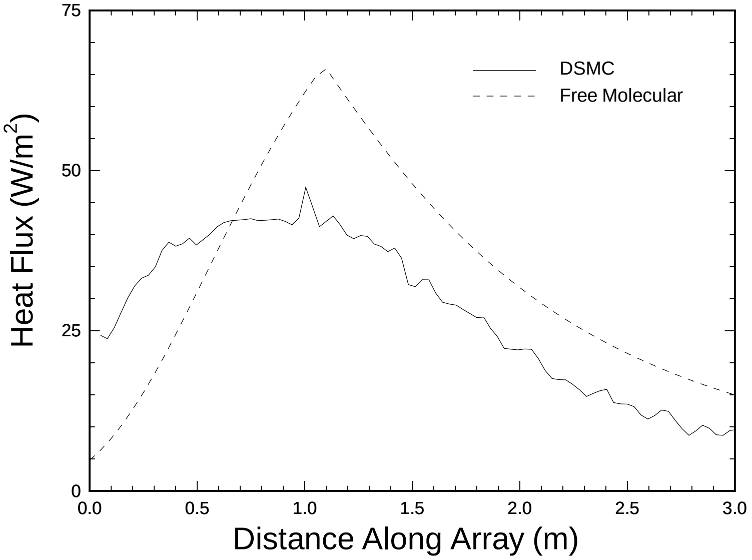

Figure 7.17. Comparison of simulation and free molecular heat flux on array surface for 0°array case.

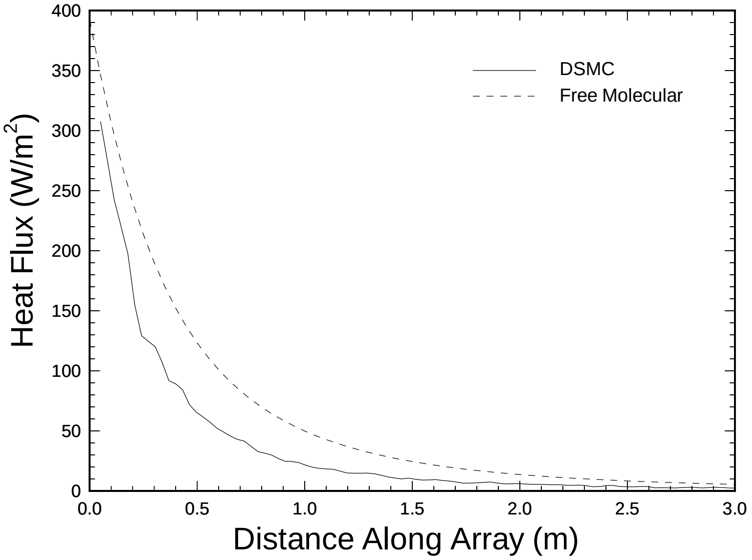

Figure 7.18. Comparison of simulation and free molecular heat flux on array surface for 30°array case.

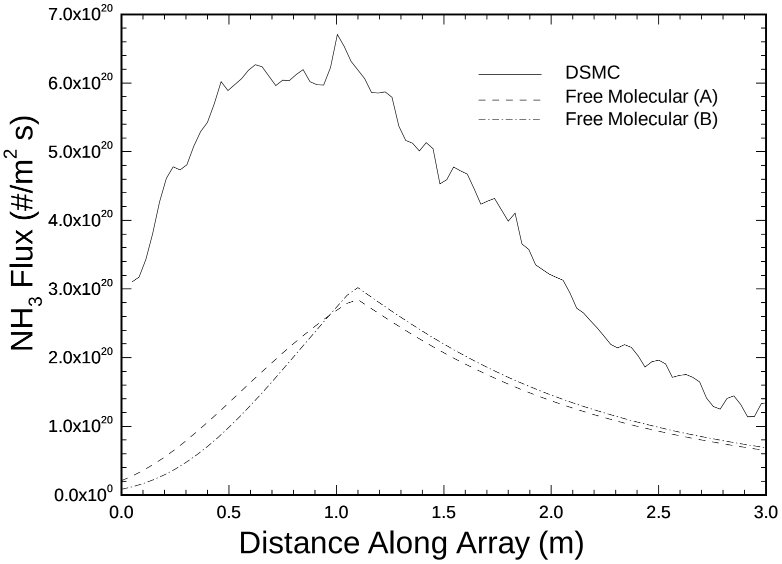

Figure 7.19. Comparison of simulation and free molecular NH_3 flux on array surface for 0°array case.

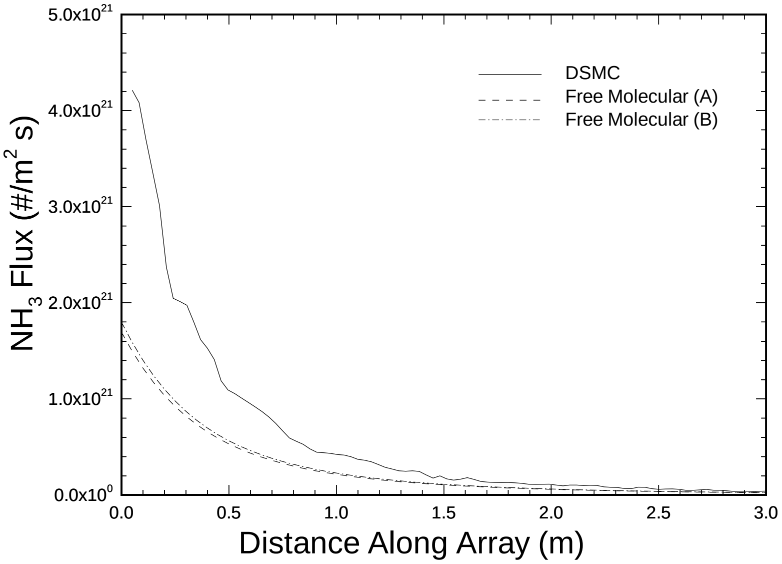

Figure 7.20. Comparison of simulation and free molecular NH_3 flux on array surface for 30°array case.

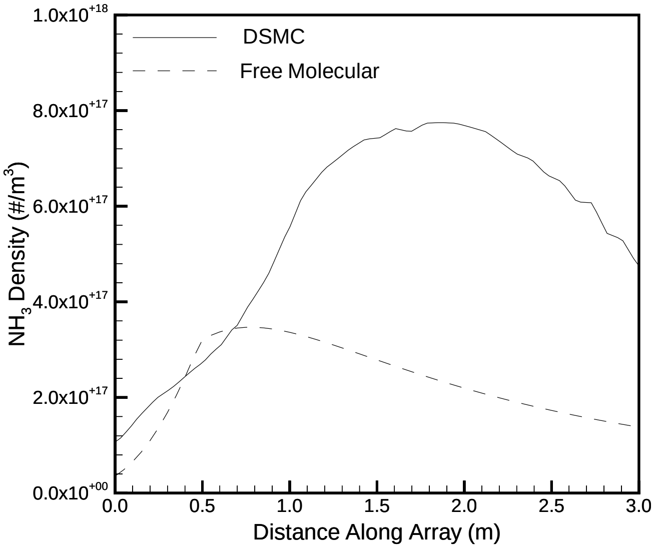

Figure 7.21. Comparison of NH_3 densities from stand alone plume simulation and free molecular model. Data at location of array surface for 0°case.

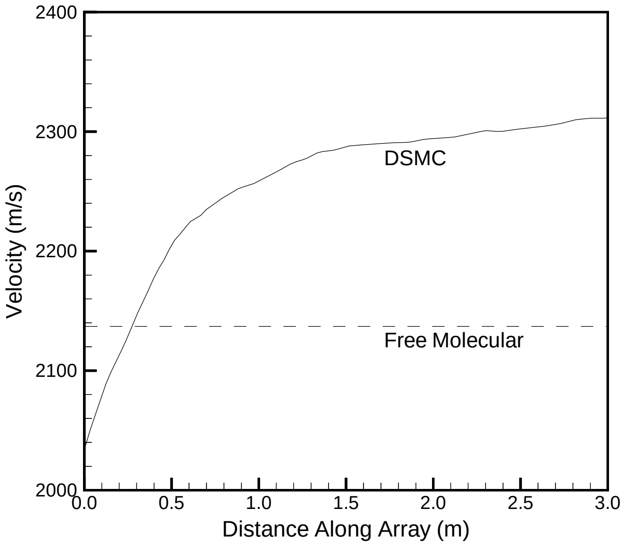

Figure 7.22. Comparison of NH_3 velocities from stand alone plume simulation and free molecular model. Data at location of array surface for 0°case.

shown in the linear profiles. The net forces are comparable which is consistent with similar pressures. Free molecular flow predicts a higher energy transfer and a significantly lower net flux of ammonia, both consistent with the previous comparisons.

In all of the previous analyses and simulations, surfaces were assumed to be fully accommodating. A more accurate representation of a real surface would assume partial accommodation. An accommodation coefficient, σ, of 0.8 is typical for a metallic surface. Heat transfer and shear stress in free molecular flow scale linearly with accommodation coefficient. Free molecular pressure (eq. 6.4 for full accommodation) is given by:

$$p = \frac{1}{2}\rho U^2 \left[2(2-\sigma)(\sin\alpha)^2 + \sigma\sqrt{\pi\frac{\gamma-1}{\gamma}}\sqrt{\frac{T_W}{T_0}}\sin\alpha \right] \tag{7.4}$$

Table 7.5 compares integrated impingement effects for full and partial accommodation. As expected, net heat transfer to the array scales with accommodation coefficient. The net force in the x direction, parallel to the plume axis, is a result of surface shear and consequently also scales with σ. The other two force components and torque are essentially unaffected by a change in accommodation coefficient. These properties are primarily caused by surface pressure which scales non-linearly with σ.

Examination of the comparisons in Table 7.5 and the expression for impingement pressure (eq. 7.4) indicate that the surface model will have some effect on the net impingement effects. Heat transfer will be over predicted by a fully diffuse model. Momentum transfer will be over predicted if primarily a result of shear forces. If more direct impingement is involved, such as might occur in a docking maneuver, the effect is less clear and will depend on the surface and plume temperatures and the geometry through angle of attack. These observations should hold for both an analytical model and DSMC simulations.

7.5 Conclusions

The calculation of a satellite configuration indicates the ability to simulate real engineering configurations. Calculations are expensive in three dimensions but can be performed efficiently in parallel. Careful use of variable scaling can reduce the overall cost significantly.

Problems which can be broken down into two distinct parts such as the near and far field of a plume can be more efficiently computed using two separate simulations. If the first portion can be computed assuming axial symmetry, the overall cost can be reduced significantly. This type of hybrid approach is particularly appropriate for engineering problems where simulations of a number of far field problems may be begun using a single near field result.

Free molecular theory provides reasonable qualitative agreement for impingement properties. However, boundary layer and multi-species effects are not properly handled by the simple plume model.

Table 7.3. Integrated impingement effects for $\Phi = 0°$.

	DSMC	Free Molecular
Net Force, X component	0.100 N	0.118 N
Net Force, Y component	-0.0107 N	-0.00588 N
Net Force, Z component	-0.0814 N	-0.0647 N
Net Torque (centerline)	0.0405 Nm	0.0369 Nm
Net Heat Transfer	101 W	112 W
Net incidence of NH_3	$1.44 \times 10^{21} s^{-1}$	$4.70 \times 10^{20} s^{-1}$

Table 7.4. Integrated impingement effects for $\Phi = +30°$.

	DSMC	Free Molecular
Net Force, X component	0.0922 N	0.148 N
Net Force, Y component	0.0228 N	0.0226 N
Net Force, Z component	-0.0645 N	-0.0563 N
Net Torque (centerline)	0.0519 Nm	0.0509 Nm
Net Heat Transfer	88.3 W	137 W
Net incidence of NH_3	$1.43 \times 10^{21} s^{-1}$	$6.15 \times 10^{20} s^{-1}$

Table 7.5. Effect of accommodation coefficient on free molecular integrated impingement effects.

	$\Phi = 0°$	$\Phi = 0°$	$\Phi = +30°$	$\Phi = +30°$
	$\sigma = 1.0$	$\sigma = 0.8$	$\sigma = 1.0$	$\sigma = 0.8$
Net Force, X component	0.118 N	0.0945 N	0.148 N	0.118 N
Net Force, Y component	-0.00588 N	-0.00471 N	0.0226 N	0.0227 N
Net Force, Z component	-0.0647 N	-0.0656 N	-0.0563 N	-0,0531 N
Net Torque (centerline)	0.0369 N	0.0377 Nm	0.0509 Nm	0.0490 Nm
Net Heat Transfer	112 W	89.5 W	137 W	109 W

Chapter 8

Conclusions

The DSMC method has been shown to provide accurate modeling of low density plume impingement flows. Comparisons with experimental data for impingement properties yield good agreement for both axisymmetric and three-dimensional problems. The qualitative and quantitative agreement in surface properties indicates that the plumes are modeled correctly.

Although the problems used for verification purposes involved only a single species gas the method should extend to multi species plumes. The model satellite configuration which was considered in chapter 7 considered a hydrazine thruster with three molecular species. The results from these simulations showed reasonable behavior and magnitude.

The simulations of hydrazine thruster impingement indicate the importance of plume modeling for spacecraft design. When the thruster is fired in the simulated configuration, the simulations indicate that approximately 20% of the generated thrust is exerted on the spacecraft solar array. A noticeable fraction of the propellant energy is therefore wasted. A significant flux of ammonia is found to be incident on the array which could be a contamination concern.

Free molecular analysis provides a reasonable estimate of impingement properties for highly rarefied plume flows. Analytical calculations of momentum and energy transfer will generally result in over estimates of these properties. As a result, this type of analysis will generally result in a conservative safety factor for impingement effects.

The free molecular model is most accurate for simple plumes involving a single gas species. A difficulty arises in determining molecular parameters for gases with variable composition. This study has only considered fluid mechanical effects and basic surface physics. More complicated flow physics such as chemistry or electro-magnetic fields occurring in the plumes of electric propulsion devices will cause further problems for the free molecular model.

The MONACO implementation of the DSMC method provides a solid platform for simulating complex plume flows efficiently. A three-dimensional version has been developed which includes all of the features and flow physics of the two-dimensional version. Unstructured and structured grid capability provides flexibility for modeling complex geometries and flow fields. Variable scaling of time steps and particle weights provide the ability to tailor the local simulation parameters to flow conditions.

Both versions of MONACO provide high numerical efficiency using an object oriented data structure and optimized routines for workstation processors. Parallel operation using simple load balancing schemes results in greatly reduced computation times. Parallel efficiencies over 90% have been attained. Large simulations involving up to three million particles and 300,000 cells have been performed in parallel on an IBM SP2 using up to 16 processors.

Even with a numerically efficient code it is important to maximize the efficiency of a simulation. Since the computational cost of DSMC is to first order proportional to the number of particles employed, an optimal simulation uses as few particles as are needed to produce an accurate result. Scaling of the number of particles per cell with density results in wasted computational effort over resolving low density regions where there is limited flow physics. This is particularly important in plume flows where densities can vary by many orders of magnitude.

A combination of variable scaling and grid manipulation can be used to reduce the variation in particle count and improve efficiency. Variable time scales are generally applicable and do not affect the accuracy of a steady state calculation. Variable particle weights can cause significant statistical fluctuations and should be used with caution. There should be minimal bulk flow across changes in weight scale and the weight scales should increase with decreasing density in order to minimize particle cloning.

Designing an optimal computational grid is always important. Although ideally cell sizes should be proportional to the local mean free path in all dimensions this is not strictly necessary in directions perpendicular to the local flow gradients. Cell stretching in directions perpendicular to gradients produces an increase in particle resolution at the cost of some spatial resolution. Proper use of cell stretching can greatly improve the efficiency of a calculation. The use of hybrid grids which combine structured and unstructured elements can facilitate this for both two and three-dimensional flows.

Complex flow problems can often be simulated more efficiently by breaking the problem down into stages. If properties at an interface surface can be represented by equilibrium distributions as quantified by a breakdown parameter, then results from one stage can be used to initialize the second stage calculation. This is especially appropriate for plume impingement problems where a single calculation of the near field plume of a thruster can be used to simulate several spacecraft configurations.

The requirement of an equilibrium interface surface poses a limitation for this technique. This requirement could be relaxed if the first stage sampled actual distribution functions rather than mean, bulk properties. These distributions could then be used as a nonequilibrium inflow condition for the second stage. While this would be memory intensive, the benefits of splitting a problem into two stages might warrant it.

8.1 Future Work

The results presented in this thesis demonstrate the ability to simulate the fluid mechanics of plume flows in realistic geometric configurations at reasonable computational cost. More advanced physical effects should now be incorporated into plume simulations in order to more accurately represent real thrusters and model a wider range of plume effects.

Development of more advanced surface models is important for accurate prediction of contamination and deposition effects. A surface model which assumes diffuse reflection with full or partial accommodation is sufficient to calculate impingement forces and heat transfer. In order to directly calculate surface contamination, the probability of a particle sticking to a surface should be included. This probability would ideally be a function of a particle's energy. This type of model is being developed for DSMC for materials processing applications. Removing particles which stick to a surface could have a significant effect on the plume in the vicinity of the surface, with this effect increasing with decreasing Knudsen number.

An advanced surface model could include a representation of the surface within a DSMC simulation. Adsorbed particles would be associated with a section of the surface allowing direct simulation of surface chemistry and the growth of a deposition layer.

Modeling of electro-magnetic fields is another important area of development for calculation of plume impingement. The high specific impulse electric thrusters (ion thrusters and arcjets) ionize the propellant gases. In order to simulate the resultant plasma, acceleration due to electric and magnetic fields must be included.

Further work is also needed on the computational aspects of DSMC. As plume simulations become larger and involve more complex physics it is important to continue to improve the numerical efficiency of simulations.

The use of grid manipulation to improve particle resolution can have a significant impact on computational cost. It is important that this technique be used for three-dimensional problems as well as two-dimensional. Hybrid grids involving hexahedral or prismatic cells mixed with tetrahedral cells can provide three-dimensional grid flexibility. Alternatively, triangles and tetrahedra with a high aspect ratio could be used to provide cell stretching at surfaces,

One of the complications in generating a computational grid is the fact that flow field properties are not accurately known until a simulation has been performed. In this thesis, a coarse, preliminary simulation run on a simple, unadapted grid is used to produce flow field data needed during generation of the final grid. This is a time consuming first step and might not produce an suitable grid due to insufficient resolution in the preliminary simulation.

A more robust solution would be to perform grid adaption periodically during a simulation. Local, transient flow properties would be used as the basis for adaption. The adaption would proceed either through dividing or combining existing cells or complete regeneration of the grid. Complete regeneration could be costly and more difficult to parallelize but would offer better grid quality. A staged approach using both methods at different times would likely be the best solution.

Dynamic grid adaption would also incur the cost of reassigning the all of the affected particles to new cells in order to preserve the object oriented data structure. The cell centered data structure makes this potentially expensive if a complete regeneration of the grid is performed.

Dynamic adaption of the domain decomposition would improve the parallel efficiency of parallel simulations. Load balancing could be performed when the parallel efficiency measured by the code drops below a specified limit. While load balancing can be performed manually by stopping the simulation, automating this step would reduce user input and decrease the amount of real time needed for a calculation. Dynamic load balancing would require the transfer of cells and their constituent particles between processors.

Bibliography

[1] D. Baganoff and J. D. McDonald. A collision-selection rule for a particle simulation method suited to vector comp[uters. *Physics of Fluids*, pages 1248–1259, July 1990.

[2] E. J. Beiting, L. Garman, I. D. Boyd, and D. VanGilder. Cars velocity and temperature measurements in a hydrogen resistojet: Comparison with Monte Carlo calculations. AIAA Paper 95-2382, July 1995.

[3] G. A. Bird. Monte-Carlo simulation in an engineering context. In *Rarefied Gas Dynamics*, pages 239–255, 1980.

[4] G. A. Bird. *Molecular Gas Dynamics*. Clarendon Press, Oxford, 1994.

[5] C. Borgnakke and P. S. Larsen. Statistical collision model for Monte Carlo simulation of polyatomic gas mixture. *Hournal of Computational Physics*, 18:405–420, 1975.

[6] I. D. Boyd. Analysis of rotational nonequilibrium in standing shock waves of nitrogen. *AIAA Journal*, 28:1997–1998, November 1990.

[7] I. D. Boyd. Extensive validation of a Monte Carlo model for hydrogen arcjet flowfields. *Journal of Propulsion and Power*, 13:775–782, 1997.

[8] I. D. Boyd, G. V. Candler, and D. A. Levin. Dissociation modeling in low density flows of air. *Physics of Fluids*, 7:1757–1763, 1995.

[9] I. D. Boyd, Y. Jafry, and J. Vanden Beukel. Particle simulations of helium microthruster flows. *Journal of Spacecraft and Rockets*, 31:271–277, 1994.

[10] I. D. Boyd, P. F. Penko, D. L. Meissner, and K. J. DeWitt. Experimental and numerical investigations of low-density nozzle and plume flows of nitrogen. *AIAA Journal*, 30:2453–2461, October 1992.

[11] I. D. Boyd and J. P. W. Stark. Modeling of a small hydrazine thruster plume in the transition flow regime. *Journal of Propulsion and Power*, 6:121–126, 1990.

[12] Primex Aerospace Company. private communication, 1997.

[13] S. Dietrich and I. D. Boyd. Scalar and parallel optimized implementation of the direct simulation Monte Carlo method. *Journal of Computational Physics*, 126:328–342, 1996.

[14] S. Doring. Experimental plume impingement heat transfer on inclined flat plates. Technical Report IB 222–90 A 36, DLR, September 1990.

[15] G. I. Font and I. D. Boyd. Numerical study of the effects of reactor geometry on a chlorine helicon etch reactor. AIAA Paper 96-0591, January 1996.

[16] D. Giordano and R. Niccoli. Analysis of the msg satellite thruster and plume flows. In *Rarefied Gas Dynamics*, pages 519–524, ?, 1996. ?

[17] T. G. Howell. An experimental analysis of low density plume flow angles and impingement for code validation. Master's thesis, University of Toledo, Toledo, Ohio, 1995.

[18] M. S. Ivanov, G. N. Markelov, Y. I. Gerasimov, A. N.‘ Krylov, L. V. Mishina, and E. I. Sokolov. Free-flight experiment and numerical simulation for cold thruster plume. AIAA Paper 98-0898, January 1998.

[19] M. S. Ivanov, G. N. Markelov, A. V. Kashkovsky, and D. Giordano. Numerical analysis of thruster plume interaction problems. In *Second European Spacecraft Propulsion Conference*, ?, 1997. ?

[20] H. Legge. Plume impingement forces on inclined flat plates. In *Rarefied Gas Dynamics*, pages 955–962, Weihneim, 1991. VCH Press.

[21] H. Legge and R. D. Boettcher. Modelling control thruster plume flow and impingement. In *Rarefied Gas Dynamics*, pages 983–992, ?, ? ?

[22] F. E. Lumpkin, P. C. Stuart, and G. J. Le Beau. Enhanced analyses of plume impingement during shuttle-mir docking using a combined cfd and dsmc methodology. AIAA Paper 96-1877, June 1996.

[23] R. P. Nance, D. B. Hash, and H. A. Hassan. Role of boundary conditions in Monte Carlo simulation of mems devices. AIAA Paper 97-0375, January 1997.

[24] D. F. G. Rault. Methodology for thruster plume simulation and impingement effects characterization using dsmc. AIAA Paper 95-2032, June 1995.

[25] Samuel A. Schaaf and Paul L. Chambré. *Flow of Rarefied Gases.* Princeton University Press, Princeton, NJ, 1961.

[26] G. A. Simons. Effect of nozzle boundary layers on rocket exhaust plumes. *AIAA Journal*, 10:1534–1535, 1972.

[27] G. P. Sutton. *Rocket Propulsion Elements.* John Wiley and Sons, New York, sixth edition, 1992.

[28] J. F. Thompson. National grid project. Computing Systems in Engineering, 3, No. 1–4, 1992.

[29] D. B. VanGilder, G. I. Font, and I. D. Boyd. Hybrid Monte Carlo - Particle in Cell simulation of an ion thruster plume. 25th International Electric propulsion Conference, August 1997.