2007-03-29

A Three-dimensional Direct Simulation Monte Carlo Methodology on Unstructured Delaunay Grids with Applications to Micro and Nanoflows

Ryan Earl Chamberlin
Worcester Polytechnic Institute

A Three-dimensional Direct Simulation Monte Carlo Methodology on Unstructured Delaunay Grids with Applications to Microflows and Nanoflows

by

Ryan E. Chamberlin

A Dissertation

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirement for the degree of

Doctor of Philosophy

in

Mechanical Engineering

By

_____

March 2007

APPROVED:

_____

Dr. Nikolaos A. Gatsonis, Advisor
Professor, Mechanical Engineering Department

_____

Dr. John J. Blandino, Committee Member
Assistant Professor, Mechanical Engineering Department

_____

Dr. David J. Olinger, Committee Member
Associate Professor, Mechanical Engineering Department

_____

Dr. George E. Karniadakis, Committee Member
Professor, Division of Applied Mathematics, Brown University

_____

Dr. Mark W. Richman, Graduate Committee Representative
Associate Professor, Mechanical Engineering Department

ABSTRACT

The focus of this work is to present in detail the implementation of a three dimensional direct simulation Monte Carlo methodology on unstructured Delaunay meshes (U-DSMC). The validation and verification of the implementation are shown using a series of fundamental flow cases. The numerical error associated with the implementation is also studied using a fundamental flow configuration.

Gas expansion from microtubes is studied using the U-DSMC code for tube diameters ranging from 100μm down to 100nm. Simulations are carried out for a range of inlet Knudsen numbers and the effect of aspect ratio and inlet Reynolds number on the plume structure is investigated. The effect of scaling the geometry is also examined.

Gas expansion from a conical nozzle is studied using the U-DSMC code for throat diameters ranging from 250 μm down to 250 nm. Simulations are carried out for a range of inlet Knudsen numbers and the effect of inlet speed ratio and inlet Reynolds number on the plume structure is investigated. The effect of scaling the geometry is examined.

Results of a numerical study using the U-DSMC code are employed to guide the design of a micropitot probe intended for use in analyzing rarefied gaseous microjet flow. The flow conditions considered correspond to anticipated experimental test cases for a probe that is currently under development. The expansion of nitrogen from an orifice with a diameter of 100μm is modeled using U-DSMC. From these results, local 'free stream' conditions are obtained for use in U-DSMC simulations of the flow in the vicinity of the micropitot probe. Predictions of the pressure within the probe are made for a number of locations in the orifice plume. The predictions from the U-DSMC simulations are used for evaluating the geometrical design of the probe as well as aiding in pressure sensor selection.

The effect of scale on the statistical fluctuation of the U-DSMC data is studied using Poiseuille flow. The error in the predicted velocity profile is calculated with respect to both first and second-order slip formulations. Simulations are carried out for a range of channel heights and the error between the U-DSMC predictions and theory are calculated for each case. From this error, a functional dependence is shown between the scale-induced statistical fluctuations and the decreasing channel height.

ACKNOWLEDGEMENTS

Sincerest appreciation is extended to my advisor, Prof. Gatsonis. I have learned a great deal from you over the years, both of science and of life. You have shaped the professional that I have become and further defined my sense of scientific rigor and engineering creativity. Thank you for the years of support and inspiration.

I would like to thank the members of my committee for their time, patience, and expertise. Your insights strengthen my work as well as my understanding.

Great thanks are given to the past members of CGPL whose work has been a basis for the current implementation of U-DSMC.

Gratitude is extended to the Mechanical Engineering Department, specifically Barbara Edilberti, Barbara Furhman and Pam St. Louis. Thank you kindly for your intuition, experience, and pleasantry. You bring a sense of family to the Mechanical Engineering department and have created a home-away-from-home for countless graduate students.

Several of the WPI faculty provided invaluable expertise on many aspects of this research. However, I would like to specifically acknowledge the efforts of Sia Najafi and Randy Robertson for their expertise, continued support and endless patience with the Linux uninitiated.

I would like to thank all of the teachers, coaches and professors who have collectively molded the person that I am today. I would like to specifically thank Prof. Nunes of the Univ. of Rhode Island, it is your discipline and scrutiny of detail that is instilled in all the work that I do, and at times of need, it is your teachings that resound in my mind and enforce the purity of science and the rigor that must define all that it touches.

I would like to extend the sincerest thanks and appreciation to all of the friends that I have made here at WPI throughout the years. You are too numerous to name yet too dear to forget.

Special acknowledgment must be made to the friends who always had the misfortune of having to serve as an outlet for the occasional yet inevitable frustrations of the graduate student experience. Anton, Brian, Dave, and Jimmy, thank you for your friendship, patience and comradery. Thank you kindly for always answering the call of duty when a friend was in need.

Special thanks go out to all of my family. Your continued love, support, and misplaced admiration are empowering. To my parents, I learned the value of education both from your instruction and your influence. My success is a reflection of your hard work and sacrifice, thank you for all that I am and all that you've done. To my Grandmother and Mr. Mac, the financial and motivational support you have lent throughout the years have helped to make this possible. You taught me the value of perseverance, thank you. To my Avó and Avô, you have instilled within me a deep-rooted work ethic that I shall carry with me throughout life. You have achieved so much starting from so little and in doing so you have helped to make this endeavor possible. Thank you.

My undying gratitude goes out to my wife, Michelle. Your endless love, support and patience have given me the peace of mind needed to dedicate my efforts toward this goal. You, more so than anyone, have shown me the value of combining an unrelenting work ethic with a focused intellect. This work, and so much else, would not have been possible without you. I will love you, and owe you, always.

This work was partially supported by NSF's NIRT Program through Grant DMI-0210258 and AFOSR's Computational Mathematics Program through Grants F49620-03-1-0219 and FA9550-06-1-0236. I would also like to acknowledge the support obtained through the 2004 Fellowship from the Massachusetts Space Grant Consortium and teaching assistantships obtained through the Mechanical Engineering Dept. for 2002 and 2003.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Boldface denotes a vector. The magnitude of a vector is denoted using the same symbol as the vector, but without boldface. Duplicate use of a symbol, or usage not defined below, will be clarified within the text.

a	sound speed
b	distance of closest approach
d	molecular diameter
$\boldsymbol{c}$	molecular velocity
$\boldsymbol{c}_0$	drift velocity
$\boldsymbol{c}'$	thermal velocity
$\boldsymbol{c}_r$	relative velocity
E	portion of available energy
F_N	particle numerical weight
k_B	Boltzmann constant
Kn	Knudsen number
L	characteristic length
m	mass of molecule
Ma	Mach number
n	number density
$\boldsymbol{n}$	normal vector
N	number of molecules in volume
$\dot{N}$	number flux
p	pressure
$\boldsymbol{r}$	position vector
Re	Reynolds number
S	Speed Ratio
T	temperature
U_∞	free stream speed
V	sample volume
u	x-component of drift velocity
v	y-component of drift velocity
w	z-component of drift velocity
u'	x-component of thermal velocity
v'	y-component of thermal velocity
w'	z-component of thermal velocity
u_0	x-component of molecular velocity
v_0	y-component of molecular velocity
w_0	z-component of molecular velocity
β	reciprocal of most probable speed
Δt	elapsed time
ε	diffuse fraction
Ξ	available modes
Λ	rotational relaxation number
ρ	number density
χ	scattering angle
ζ	internal degrees of freedom
σ_T	total collision cross section
δ	mean molecular spacing

1. INTRODUCTION

Steady advances in aerospace technologies coupled with the rapid expansion of Micro-Electro-Mechanical Systems (MEMS) and the promise of Nano-Electro-Mechanical Systems (NEMS) have resulted in a substantial demand for modeling tools capable of capturing gaseous flows in micro and nano devices, typically featuring complex geometries. These flows are usually quantified through the use of the Knudsen number (Kn), defined as the ratio of the mean free path (λ) to the characteristic length (L). The Knudsen number can be related to the Mach number and Reynolds number by (Karniadakis and Beskok, 2002):

$$Kn = \frac{\lambda}{L} = \sqrt{\frac{\gamma\pi}{2}}\frac{M}{Re} \tag{1.1}$$

By evaluating the Knudsen number the respective flow regime can be classified according to the following commonly accepted guidelines (Schaaf and Chambre, 1961):

- $Kn < 0.01$ Continuum Flow
- $0.01 < Kn < 0.1$ Slip Flow
- $0.1 < Kn < 10$ Transitional Flow
- $10 < Kn$ Free-Molecular Flow

For many gaseous flows occurring in MEMS and NEMS devices the Knudsen number is in the range of 0.1 to 10 (Liou and Fang, 2000). As a result, the typical continuum based fluid modeling tools, built upon the Navier-Stokes equations, utilized during the design process cannot meet the needs of a growing number of developers whose applications lie in the rarefied regime. Furthermore, the fundamental understanding of gas flow characteristics at micro and nano scales is lacking in the slip and transitional Knudsen regimes. The range of validity of continuum based modeling is seen in Figure 1.

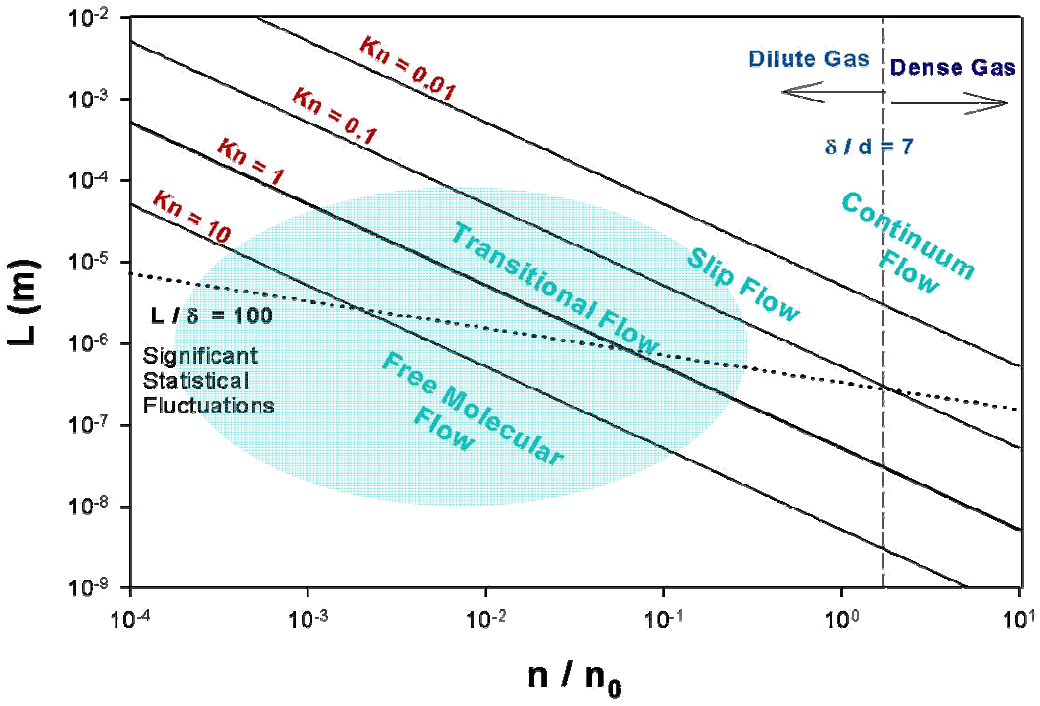


Figure 1. Graphical representation of the Knudsen regime, $n_0 = 2.68 \times 10^{25}\,m^{-3}$ for ideal gas at STP (adapted from Karniadakis and Beskok, 2002) as well as the dilute gas limit and region of significant statistical fluctuations in volume sampled properties, where *L* is a characteristic length, *d* is the molecular diameter and δ is the mean molecular spacing.

In order to address these issues this work is devoted to the implementation of the direct simulation Monte Carlo (DSMC) methodology of Bird (1994) on unstructured three dimensional meshes which feature extremely flexible geometric resolution and can thus be applied to a number of micro and nano scaled flow scenarios.

This work is motivated by three primary goals.

- The first goal of this dissertation is to develop a fully functional DSMC code implemented on unstructured Delaunay grids (U-DSMC) with extensive geometric flexibility and ease-of-use.

- The second goal is to verify each underlying algorithm of the U-DSMC code and to validate its overall applicability to modeling rarefied gaseous flows.
- The final goal of this research is to apply the U-DSMC code in a series of investigations which probe the physical phenomena associated with gaseous flows in micro and nanoscale devices and to establish the relationship between the characteristic scale of the flow and statistical fluctuations in U-DSMC results.

1.1 DSMC Basics

The DSMC method has become readily accepted as an effective method of modeling rarefied and nonequilibrium gaseous flows. The method was initially developed in the early 1960's by G.A. Bird (1994). DSMC is a direct simulation approach that relies on a number of probabilistic procedures that are valid only for dilute gases. The basis of the method lies in the tracking of representative simulated molecules. Each simulated molecule represents a large number of real molecules which lie in a similar range of phase-space. The representative molecules are tracked as they interact with other molecules and with domain boundaries. The fundamental assumption of the method is that for dilute gases the molecular motion and the intermolecular collisions can be uncoupled over a small time step (τ). Within the simulation, the simulated particles transverse distances which correspond to the time step and their respective velocity. Any resulting boundary interactions are treated during the motion step. It should be noted that the time step must be chosen such that it is small with respect to the mean collision time of the flow (Bird, 1994). The intermolecular collisions are treated at the end of each time step in a probabilistic fashion formed from basic kinetic theory.

Modifications and additions to the original algorithms aided in increasing the accuracy of the DSMC method in the early developmental stages (Borgnakke and Larsen, 1975, Cercignani and Lampis, 1974, Lordi and Mates, 1970). Over the last two decades DSMC has become widely accepted as the primary method for modeling rarefied gas flows. In recent years further algorithmic refinements have been introduced (Boyd, 1993, Haas and Boyd, 1993, Haas et al, 1994) that have extended the applicability of the method. Analytical efforts have also been made to lend further validity to the method and to quantify the statistical error inherent to DSMC simulations (Rjasanow and Wagner, 1998, Alexander et al, 1998, Garcia and Wagner, 2000, Hadjiconstantinou, 2000).

1.2 Outstanding Mathematical and Computational Issues of U-DSMC

Implementation of the DSMC method on unstructured grids yields vast improvements in regards to geometric flexibility when compared to structured DSMC codes, but the additional flexibility comes at the cost of added complexity and computational overhead. Several major algorithmic issues arise when implementing the DSMC method on unstructured grids. Difficulties resulting from the unstructured nature of the local cell configuration require algorithmic advances as pertaining to particle motion, particle-surface boundary interaction, resulting surface force calculations, tracking of fluxal properties, as well as the implementation of both hypersonic and subsonic injection routines. As a result of these issues the majority of DSMC codes utilized in the literature are based on structured meshes. Due to the significance of the added difficulties only a few unstructured DSMC codes have been developed to date, however, their utility has been made apparent in the following studies.

Wu et al (2002) present the implementation of DSMC on an unstructured 2-D grid. Although 2-D, the code does benefit from the added feature of mesh-spacing control through adaptive meshing. The work clearly displays the benefits of unstructured meshes as well as mesh-spacing control in the resultant flow fields of hypersonic flow over a cylinder.

In Wu and Lian (2003) the DSMC method is extended into three dimensions by use of unstructured tetrahedral cells. Code validation is reported by means of a benchmark comparison for supersonic corner flow as well as orifice expansion flow. The method is also applied to hypersonic flow over a sphere as well as an analysis of the flow field in a spiral drag pump.

In Wu and Tseng (2005) dynamic domain decomposition is added to the DSMC code and the resulting capabilities are applied to model a number of cases including two two-dimensional hypersonic flows, a three-dimensional hypersonic flow and a three-dimensional near-continuum hypersonic twin-jet gas flow problem.

In Boyd and Wang (2001) the range of applicability of their DSMC code MONACO was presented. MONACO is reported to be a general, cell-based, object-oriented, parallelized implementation of the DSMC method which can operate on both structured and unstructured grids. In this particular study, the code was applied to near-continuum regime simulations of hypersonic flow over hollow-cylinder and double-cone geometries using structured meshes. The computational cost of near-continuum regime simulations with such a flexible code structure was reported to be so significant that the results given in the paper were not steady state values. Other simulations utilizing the MONACO code have been carried out over a wide range of applications. In the work of Kannenberg and Boyd (1999), MONACO was used to carry out plume studies. In the work by Karipides et al (1999), MONACO was applied to a detailed simulation of the surface chemistry that is responsible for spacecraft glowing. Additional studies

have also been conducted into thin film deposition (Chen et al, 1998, Balakrishnan et al, 2000) as well as meteoroid trail analysis (Boyd, 2000).

Development of unstructured particle simulations at the Computational Gas and Plasma Dynamics Laboratory (CGPL) on the campus of Worcester Polytechnic Institute has been ongoing for some time. In the work of Kovalev (2000) the development of an unstructured Voronoi-Delaunay grid generator, which provides the data structure underlying the unstructured DSMC code, was presented. In Hammel et al. (2001) the findings of preliminary DSMC simulations of gaseous flows in nozzles and microchannels were shown. In Hammel (2002) the basic elements of the unstructured DSMC code as well as a particle-in-cell (PIC) code were presented. In Spirkin (2006) further development of the PIC methodology was shown, as well as code validation, numeric heating studies and plasma microdevice simulations.

To date, only a few DSMC codes have been reported in literature to have been implemented on unstructured grids. Out of those unstructured codes, very few have full three-dimensional capabilities. Additionally, extensive literature searches yield limited reports of a three-dimensional unstructured DSMC code that has subsonic flow modeling capabilities. As a result there is a strong need to combine the geometric flexibility of a three-dimensional unstructured DSMC implementation with a flexible and robust subsonic modeling technique. To address this need the current implementation of U-DSMC has been established by means of algorithmic refinement, extension and addition from the foundations laid by the work of Kovalev (2000) and Hammel (2002) and now includes the capability of providing subsonic modeling of geometrically complex devices. Furthermore, the U-DSMC code provides a platform from which the effects of statistical fluctuations in micro and nano scaled unstructured DSMC can be in investigated and characterized.

1.3 Major Limits and Fundamental Assumptions of the DSMC Method

As is true for all modeling schemes, the DSMC method has a number of built in assumptions, underlying limits and bounds on the computational accuracy of its results. One of the primary limits of the DSMC method is based upon the assumption that molecular motion occurs outside the range of influence of neighboring molecules and that molecular motion may be uncoupled from inter-molecular collisions. Furthermore, the method assumes that when inter-molecular collisions do occur the overwhelming majority of them will be binary. Based on these assumptions the DSMC method is limited to modeling of dilute gases. The dilute gas assumption is taken to be valid for flows where the mean molecular spacing δ is much greater than the effective molecular diameter (Bird, 1994):

$$\delta >> d_{mol} \tag{1.2}$$

The mean molecular spacing can be related to the number density of the flow through:

$$\delta = n^{-1/3} \tag{1.3}$$

The number density can in turn be related to the pressure and temperature of a dilute gas by means of the Ideal Gas Law:

$$P = nk_B T \tag{1.4}$$

where k_B is the Boltzmann constant ($k_B = 1.3805 \times 10^{-23}$ J/K), P is the pressure and T is the temperature. Figure 2 illustrates the limiting values of pressure for the dilute assumption as a function of effective molecular diameter over a range of common gases.

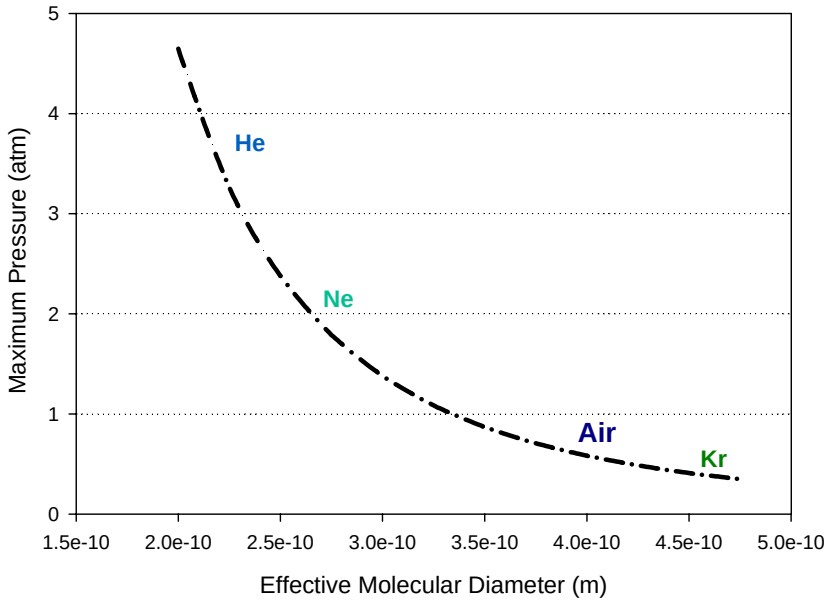


Figure 2. Maximum pressure for which the dilute gas assumption is valid plotted as a function of effective molecular diameter, with T = 300K and δ / d = 7.

The DSMC method is stochastic in nature. Furthermore, many of the underlying models contained in the method are phenomenological. As such, the applicability of many of the core routines which comprise the method is limited by the assumption of molecular chaos and the requirement that a large number of real molecular interactions occur for every single corresponding simulated interaction. The assumption of a large ratio of real events corresponding to a single phenomenologically simulated event can be achieved through the application of a modestly large particle weight (the number of real particles represented by a simulated particle). Unfortunately, the applicability of molecular chaos is not as clearly defined and may in fact be in question for extremely rarefied flow in micro and nanoscale domains. Furthermore, as will be shown in Section 4.4, the application of a sufficiently large particle weight is also questionable at nanoscales. These scale induced artifacts require significant

investigation in order to quantify the limit of applicability of the DSMC method in nanoscale gaseous devices.

Another key consideration in DSMC computations relates to statistical fluctuations. Due to the nature of the DSMC method, the results for the macroscopic flow parameters obtained in a given simulation are subject to statistical fluctuations. The onset of significant statistical fluctuations occurs when the volume being used for sampling of macroscopic parameters does not contain a large enough number of simulation particles. It has been reported in Karniadakis and Beskok (2002) that a sampling volume containing 10,000 molecules results in 1% statistical fluctuations in the sampled quantities. Furthermore, following the argument given by Bird (1994), a formal relation can be established between the statistical fluctuations of the macroscopic parameters to the sample volume, V, and the number of molecules in this sample volume, N.

The number of molecules in a volume element is subject to statistical fluctuation about an average value of nV, where n is the time averaged number density in the region about the volume of interest. The probability, $P(N)$, of having a particular value of N in the volume at any instant is given by the Poisson distribution as follows:

$$P(N) = \frac{(nV)^N \exp(-nV)}{N!}. \tag{1.5}$$

For large values of nV, this distribution becomes indistinguishable from a normal or Gaussian distribution of the form

$$P(N) = \frac{1}{\sqrt{2\pi nV}} exp\left\{\frac{-(N - nV)^2}{2nV}\right\}. \tag{1.6}$$

Integration of the normal distribution over the limits of $nV + A\sqrt{nV}$ to $nV - A\sqrt{nV}$, reveals that the probability of an individual sample falling within a region $A\sqrt{nV}$ about the average nV, following:

$$\phi(A\sqrt{nV}) = \int_{nV-A\sqrt{nV}}^{nV+A\sqrt{nV}} P(N)dN = erf(A/\sqrt{2}) \tag{1.7}$$

The standard deviation of the fluctuations is then $1/\sqrt{nV}$. Correspondingly, the combination of the standard deviation, given by $1/\sqrt{nV}$, in conjunction with a requirement of a standard deviation value that is preferably much less than unity results in the following requirement, which is taken as an approximate limit to assure a sufficiently small amount of statistical fluctuation will occur in samples made over volume V with a local number density of n, to be:

$$nV \gg 1 \tag{1.8}$$

Further work has been done to quantify the statistical fluctuation in sampled properties as pertaining to the DSMC method in a number of more recent studies. In Hadjiconstantinou et al (2003) predictions for the statistical error due to finite sampling in the presence of thermal fluctuations in molecular simulations was presented in detail. The authors established a relation between statistical fluctuation error and key flow properties such as Mach number, Knudsen number and the number of simulation particles in each cell. In Chen and Boyd (1996) the statistical error associated with the DSMC technique was studied in depth using nonequilibrium hypersonic flows as well as nozzle flows. Using a root mean square error the level of statistical fluctuation was quantified in each flow case. Similar studies have also been carried out to

determine the error associated with the computational cell size (Alexander et al, 1998) and the time step size (Garcia and Wagner, 2000).

Although much work has been done to quantify and bound the statistical fluctuations in DSMC in regards to sample volume and the number of simulation particles in this volume, there has not been any work to date that investigated the effects of decreasing scale on the statistical fluctuations. Therefore there is a need to characterize the increase in statistical fluctuations that occur due to the decrease in the number of simulation particles in the computational domain that results from the significant reduction of the number of real particles in rarefied gaseous flow in micron and submicron scale devices.

The limits of underlying assumptions and major approximations are shown in Figure 1. The vertical line corresponds to the limit of the dilute gas assumption for air at standard temperature and pressure. The diagonal lines indicate the limiting ranges for the various Knudsen regimes. Finally, the nearly horizontal dashed line represents the maximum length from which a sampling volume can be constructed so that statistical fluctuations in sampled properties would be negligible. It should be noted that the majority of the flow regimes encountered in this work lie in the shaded region of Figure 1. This flow regime is beyond the limits of applicability of Navier-Stokes based methods and well into the limiting range of past DSMC studies.

1.4 Microscale Propulsion Systems and Related Flows

The efforts underlying this work are primarily focused toward meeting the gaseous phase modeling needs generated from the miniaturization of devices in numerous fields of technology and science. One such field of direct interest is that of spacecraft propulsion. The underlying

motivation driving the miniaturization of propulsion devices is that mass is critical in spacecraft design. Inherent to all spacecraft systems is a delicate and difficult balance between the mass of the mission components and the mass of the propulsion system. As a result, modern spacecraft and microspacecraft development has been intimately linked with many recent MEMS breakthroughs. In addition, the drive towards ever smaller micro and eventually nanospacecraft necessitates the miniaturization of onboard propulsion technology. To date many MEMS based systems have been developed for onboard propulsion (Micci and Ketsdever, 2000, Lewis et al, 2000, Rossi et al, 2002, Kohler et al, 2002).

Performance, efficiency and spacecraft integration of micro and nanopropulsion requires investigation of internal and external jet flow. In addition, experimental investigation of microjets requires new sensors that can operate in these regimes. Due to its fundamental nature, free jet expansion has been investigated numerically and experimentally in previous studies (Campbell et al, 1992, Sharipov, 2002). Not until recently however has it been necessary to investigate the nature of free jet expansion in detail as pertaining to the expansion from microtubes of variable aspect ratio. Therefore there is a need to extend previous investigations that were limited to orifice geometries where the aspect ratio is near unity and the orifice diameter is a millimeter or larger. Additionally, there is a need to provide a basis study which offers the transitional background needed for investigations of gaseous expansion from nanotubes. To meet these needs several parametric studies have been carried out in order to characterize the effects of the primary parameters as well as the effect of scale (Chamberlin and Gatsonis, 2006 (b), Chamberlin and Gatsonis, 2007). Details of these studies are given in Section 4.1

A simple form of micropropulsion can be that of cold gas expansion from a nozzle. Due to its common utilization as a form of satellite propulsion, nozzle expansion has been investigated numerically and experimentally in numerous studies (Boyd et al, 1992, Boyd et al, 1994, Mo et al, 2005). Not until recently however has it been necessary to investigate the nature of gaseous expansion from nozzles in detail as pertaining to the expansion from micronozzles. Therefore studies are needed that will compliment and extend previous investigations that were limited to nozzles with a throat diameter of a few millimeters or larger. Additionally, basis information is needed that can serve as the transitional background needed for investigations of gaseous expansion from nanonozzles. To supply the required basis information, as well as to extend previous investigations, several parametric studies have been carried out (Chamberlin and Gatsonis, 2006 (c)). Details of these studies are given in Section 4.2.

Given the need for reduced scale thruster systems, design and testing of novel microscaled propulsion devices have seen a recent surge. As such, there is a growing need for plume diagnostics with sub-millimeter spatial resolution. The majority of past measurements in rarefied flows have often been obtained with instruments connected through orifices or tubes (e.g., neutral and ion mass spectrometers, pressure gauges, pitot probes). Following the progression of manufacturing capabilities, steel tubing with outer diameters of less than $500\,\mu m$ are now readily available from a number of industrial suppliers for a low cost. This development, coupled with the advancement of sensor technology, allows for an extension of the pitot probe design to be applied in a manner which can attain sub-millimeter spatial resolution. The incident flux and molecular composition at the entrance of such a device are related to the external flow field while the measurements taken inside the apparatus are affected by the compounding effects of the internal flow. The cases of flow in the near and free molecular regimes through tubes and

orifices have been studied both analytically and computationally for many years due to their importance in numerous technical applications (Hughes and de Leeuw, 1965, Fan and Robertson, 1969, Kannenberg and Boyd, 1996, Gatsonis et al, 1997). However, the work to date did not consider the coupled effects of jet expansion and pitot probe measurements as pertaining to microscaled orifices and probe assemblies. Therefore work was needed to extend the previous investigations by coupling past findings with numerical studies at the microscale. To meet this need the coupled study presented in Chamberlin and Gatsonis (2006 (a)) was carried out. Details of this work are given in Section 4.3.

1.5 Objectives and Approach

The primary goal of this work is to further develop, revise and enhance the unstructured DSMC code. The second goal is to verify each underlying algorithm of the U-DSMC code and to validate its overall applicability to modeling rarefied gaseous flows. The final goal of this research is to apply the U-DSMC code in a series of investigations which probe the physical phenomena associated with gaseous flows in micro and nanoscale devices and to establish the relationship between the characteristic scale of the flow and statistical fluctuations in U-DSMC results. The objectives and approaches are listed below.

1. Develop and implement algorithms in order to develop a fully functional unstructured DSMC code (U-DSMC):
    a. Revise previous implementations of particle loading and particle motion.
    b. Revise and rewrite portions of the previous implementations of hypersonic injection, specular reflection, and diffuse reflection.

c. Implement corrected algorithms for calculating elastic inter-particle collisions using either the Hard Sphere (HS) or Variable Hard Sphere (VHS) models following the methods of Bird (1994), with provisions for the future addition of the Variable Soft Sphere (VSS) model.

d. Implement corrected algorithms for calculating the rotational energy exchange in inelastic inter-particle collisions using the Larsen-Borgnakke Method (Borgnakke and Larsen, 1975).

e. Develop and implement a simple model for non-diffuse reflection featuring partial accommodation of momentum and energy for the modeling of solid boundary surfaces.

f. Develop and implement moving solid boundaries, with wall velocity specified by scalar values of speed in each Cartesian direction.

g. Modify existing calculations of macroscopic parameters in order to include the calculation of the scalar pressure for each gas species as well as for the bulk gas following the methods of Bird (1994).

h. Develop and implement calculations of species specific and bulk gas number flux and mass flux through arbitrary surfaces in the flow domain with arbitrary shape and orientation.

i. Develop and implement the capture of species specific and bulk gas distribution function data through surfaces with arbitrary shape and orientation.

j. Develop and implement calculations of gas-surface interaction properties such as pressure, shear and heat flux on arbitrary surfaces for individual species as well as the bulk gas.

k. Develop and implement calculations of both global and local surface coefficients such as the coefficient of pressure, coefficient of drag and the heat flux coefficient on arbitrary surfaces for both individual species and the bulk gas.

l. Develop and implement calculations of the total force vector on a solid body immersed in gaseous flow.

m. Develop and implement subsonic inflow boundary conditions featuring specified upstream pressure and temperature with floating face-normal drift velocity for arbitrarily oriented inlets based on the methods of Wang and Li (2004).

n. Develop and implement subsonic outflow boundary conditions featuring specified downstream pressure with floating temperature and face-normal drift velocity for arbitrarily oriented outlets based on the methods of Nance et al (1997).

o. Identify, develop and implement a means of utilizing a surface triangulation from a commercial software package as an input to the 3-D unstructured Voronoi-Delaunay grid generation program.

p. Identify, develop and implement a means of exporting and reformatting a 3-D unstructured Delaunay grid from a commercial software package into a format acceptable for use with U-DSMC.

2. Verify and validate U-DSMC by applying it to:

   a. Heat transfer between stationary infinite parallel plates, thus verifying the current implementation of: particle loading, elastic collisions, specular reflection, diffuse reflection, and surface sampling of heat flux.

b. Also use heat transfer between stationary infinite parallel plates in order to approximate the order-of-error in the 3-D unstructured DSMC code as pertaining to the time step, cell size and number of simulation particles per collision cell.

c. Hypersonic flow over a blunt flat plate, thus verifying the current implementation of: hypersonic injection, inelastic collisions, non-diffuse reflection, and surface sampling of pressure.

d. Free jet expansion of argon, thus verifying the current implementation of: molecular flux injection with upstream-inlet free boundaries, downstream-vacuum free boundaries, mass flow rate, number flux calculations, as well as the code's ability to capture the dynamics of internal to external expansion flow.

e. Subsonic Poiseuille flow, thus verifying: subsonic inflow and subsonic outflow

f. Subsonic Couette flow, thus verifying: moving diffuse wall boundaries and subsonic inflow without a pressure gradient.

3. Apply U-DSMC to explore physical phenomena at the micro and nano scale:

   a. Characterize the effects of aspect ratio, Knudsen number, Reynolds number, and speed ratio on gaseous expansion from micro and nano tubes and micro and nano nozzles.

   b. Guide the design of a micropitot probe intended for use in analyzing rarefied gaseous microjet flow.

   c. Investigate and bound the increasing statistical fluctuation that occurs with decreasing scale in DSMC.

The presentation of this work is organized in the following manner. In Chapter 2, the theory underlying the DSMC methodology, as pertaining to its implementation on unstructured Delaunay grids, is presented in detail for each aforementioned code modification or addition. In Chapter 3, the validation cases used to verify the proper execution of each underlying algorithm are shown. Chapter 3 also contains a study which quantifies the approximate order-of-error for the current implementation. In chapter 4 the U-DSMC code is applied to several case studies. The first two studies presented are investigations of the effects of aspect ratio, Knudsen number, Reynolds number, speed ratio and scale on the expansion of gas from micro and nano tubes and micro and nano nozzles. The third case presented is a study of a developmental micropitot probe immersed in the plume of a microjet. The last case presented is a study which quantifies the increase in statistical fluctuations in U-DSMC results with decreasing scale. Conclusions and recommendations for future work are contained in Chapter 5.

2. U-DSMC METHODOLOGY AND IMPLEMENTATION

The underlying algorithms of the DSMC method have been developed over time by a number of investigators. Although the fundamentals have been laid out by Bird (1994) the details of numerous algorithms have undergone modification in order to incorporate new algorithmic developments. In this work, the particulars of each algorithm have in many cases undergone significant alterations in order to be implemented on unstructured Delaunay grids. This chapter describes the computational mathematical methods and implementation of each fundamental DSMC algorithm in the current version of U-DSMC.

2.1 U-DSMC Program Flowchart

The overall structure of the U-DSMC implementation and its supporting systems is shown in Figure 3. The U-DSMC flow solver requires three input files. The three required file types are a general input file, a boundary conditions file and a grid file. The general input file contains information on gas composition, collision model specification, time step selection, and output file timing. The boundary conditions file contains required information specifying which boundary conditions to apply at each boundary surface. The grid file contains the solver required data for the unstructured grid, such as node location, face and cell connectivity and face attribute listings. The general and boundary condition input files are generated as text files by the U-DSMC user. The grid file is typically generated using either the unstructured Voronoi-Delaunay grid generation program of Kovalev (2000) referred to as U-GridGen or a module which interfaces with COMSOL. Details of the grid generations methods available to U-DSMC users are given in Section 2.3.

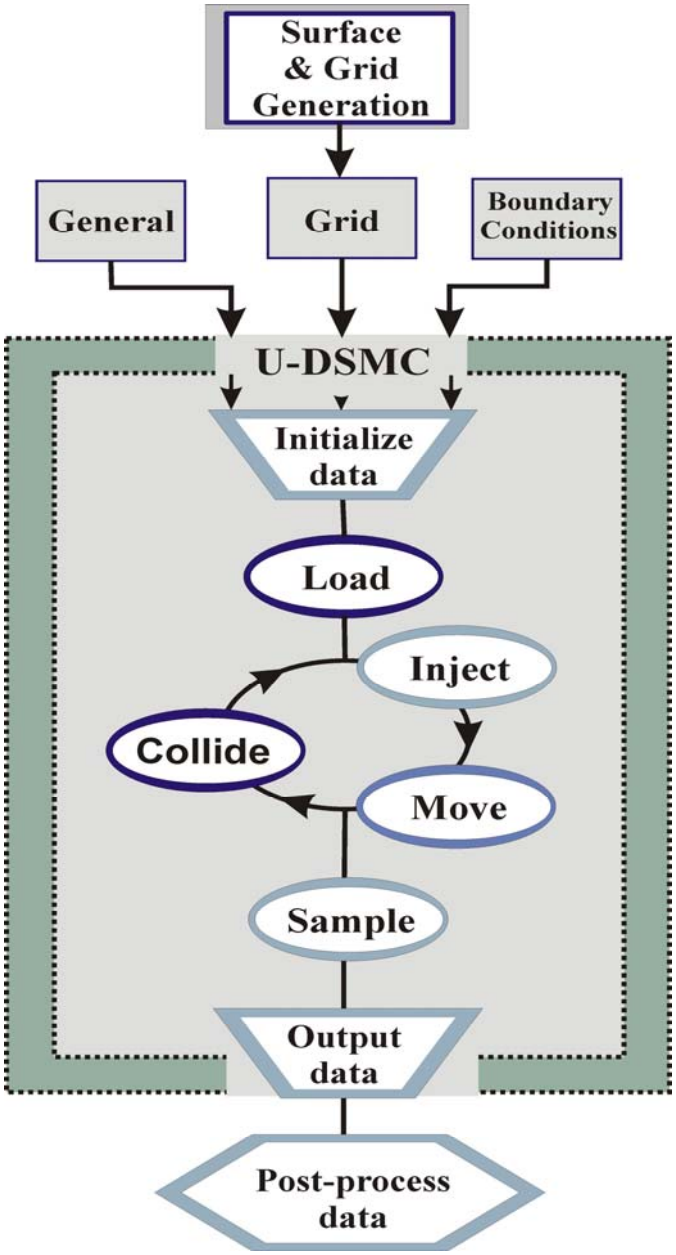


Figure 3. Underlying processes of U-DSMC.

2.2 Macroscopic Properties of Gas Mixtures

The DSMC method of Bird is stochastic by nature. However the numeric evolution of particle states by the underlying stochastic relations result in distribution functions equivalent to

those which would be obtained by solving the nonlinear Boltzmann equation. For a gas mixture the Boltzmann equation has the following form (Bird, 1994):

$$\frac{\partial\left(n_p f_p\right)}{\partial t}+\mathbf{c}\bullet\frac{\partial\left(n_p f_p\right)}{\partial \mathbf{r}}+\mathbf{F}_m\bullet\frac{\partial\left(n_p f_p\right)}{\partial \mathbf{c}}=\sum_{p=1}^{s}\int_{-\infty}^{\infty}\int_{0}^{4\pi} n_p n_q\left(f_p^{*} f_q^{*}-f_p f_q\right)c_{r,pq}\sigma_{T,pq}\,d\Omega\,d\mathbf{c}_q \quad (2.1)$$

where $n_p = n_p\left(\mathbf{r},t\right)$ is the number density at a given position and time for species p which ranges from 1 to the total number of species in the mixture (s), $\mathbf{c}$ is the mean velocity and $\mathbf{F}_m$ is the resulting acceleration due to an external force (external force per unit mass). The right hand side of the equation is commonly referred to as the *collision term.* The velocity distribution function for species p is given as f_p and represents the distribution of particle velocities within a small volume of velocity space, $d\boldsymbol{c} = du\,dv\,dw$, fitting (Bird, 1994):

$$dN / N = dn / n = f d\boldsymbol{c} \quad (2.2)$$

where $nfd\boldsymbol{c}$ is the number of particles per unit volume with a velocity between $\boldsymbol{c}$ and $\boldsymbol{c}+d\boldsymbol{c}$. The velocity distribution is normalized function and thus the integration over all velocity space follows:

$$\int_{-\infty}^{\infty} f d\boldsymbol{c} = N / N = 1 \quad (2.3)$$

Due to the high-dimensionality of the Boltzmann equation great difficulty is encountered when trying to obtain analytical solutions even for simple geometries. Complex geometric problems render analytical solution virtually impossible thus dictating the need for numerical methods.

Within the underlying structure of the DSMC method are a number of approximations derived from basic kinetic theory. As a result the progression of the algorithm mimics the progression of a Boltzmann-like solution. In fact the particle motion methodology follows the collisionless form of the Boltzmann equation. From this basis similarity it has been concluded by Wagner (1992) that the DSMC methodology of Bird is found to converge to a Boltzmann-like equation in the limit of infinite simulation particles. Although the DSMC method is not a direct numerical solution of the Boltzmann equation the results from DSMC simulations have been found to converge to those of the Boltzmann formulation when proper modeling considerations are made. Accordingly, the DSMC method can be used to model gas flows with results for macroscopic quantities, such as those described below, matching closely to those which would be obtained using the Boltzmann equation.

The primary objective of most U-DSMC studies is to obtain the macroscopic parameters of the flow. These parameters are given as moments of the distribution function. Within the work of Chapman and Cowling (1939) the moments of the distribution function are defined in relation to the average velocity of the gas mixture, often referred to as the mass-average mean velocity. The mass-average mean velocity is given by:

$$\mathbf{c}_0 = \frac{\sum_p n_p m_p \mathbf{c}_{0,p}}{\sum_p n_p m_p} \tag{2.4}$$

having components $\{u_0, v_0, w_0\}$, where the species mean velocity is given by:

$$\boldsymbol{c}_{0,p} = \overline{\boldsymbol{c}}_p = \int_{-\infty}^{\infty} \boldsymbol{c}_p f_p \, d\boldsymbol{c}_p \tag{2.5}$$

Using the mass-average mean as the reference velocity, the thermal velocity of a particle of species p is given as:

$$\boldsymbol{c}_p^{'} = \boldsymbol{c}_p - \boldsymbol{c}_0 \tag{2.6}$$

The physically significant moments of the species distribution function, when using the mass-average drift velocity as the reference velocity, are then calculated following:

Species diffusion velocity:

$$\overline{\boldsymbol{C}_p} = \overline{\boldsymbol{c}_p^{'}} = \overline{\boldsymbol{c}_p} - \boldsymbol{c}_0 \tag{2.7}$$

Mixture number density:

$$n = \sum_{p=1}^{s} n_p \tag{2.8}$$

Species translational temperature:

$$\frac{3}{2} k_B T_{tr,p} = \frac{1}{2} m_p \overline{c_p^{'2}} \tag{2.9}$$

Mixture translational temperature:

$$\frac{3}{2} k_B T_{tr} = \frac{1}{2} \overline{m\, c^{'2}} \tag{2.10}$$

Species scalar pressure:

$$p_p = \frac{1}{3} n_p m_p \overline{c_p^{'2}} \tag{2.11}$$

Mixture scalar pressure:

$$p = \frac{1}{3} n \overline{m c^{'2}} \tag{2.12}$$

Species pressure tensor:

$$\boldsymbol{p}_p = n_p m_p \overline{\boldsymbol{c}_p^{'} \boldsymbol{c}_p^{'}} \tag{2.13}$$

Mixture pressure tensor:

$$\boldsymbol{p} = n \overline{m \boldsymbol{c}^{'} \boldsymbol{c}^{'}} \tag{2.14}$$

Species stress tensor:

$$\boldsymbol{\tau}_p \equiv \tau_{ij,p} = -\left(nm_p \overline{c'_{i,p} c'_{j,p}} - \delta_{ij} p_p\right) \tag{2.15}$$

Mixture stress tensor:

$$\boldsymbol{\tau} \equiv \tau_{ij} = -\left(n\overline{mc'_i c'_j} - \delta_{ij} p\right) \tag{2.16}$$

Species heat flux vector:

$$\boldsymbol{q}_p = \frac{1}{2} n_p m_p \overline{c'^2_p \boldsymbol{c}'_p} \tag{2.17}$$

Mixture heat flux vector:

$$\boldsymbol{q} = \frac{1}{2} n\overline{mc'^2 \boldsymbol{c}'} \tag{2.18}$$

where a quantity $\overline{Q}$ is given by the mean value principle (Bird, 1994) following:

$$\overline{Q} = \frac{1}{N}\int_N Q dN = \frac{1}{N}\int_{-\infty}^{\infty} Q\, N\, f(\boldsymbol{c})\, d\boldsymbol{c} = \int_{-\infty}^{\infty} Q\, f(\boldsymbol{c})\, d\boldsymbol{c} \tag{2.19}$$

In non-equilibrium gases it is often convenient to define separate ‘species specific’ values in order to determine the degree of non-equilibrium as well as determine the macroscopic parameters applicable to each species individually. Using the species drift velocity $\boldsymbol{c}_{0,p}$ as the reference velocity the single species thermal velocity is given as:

$$\boldsymbol{c}''_p = \boldsymbol{c}_p - \boldsymbol{c}_{0,p} = \boldsymbol{c}_p - \overline{\boldsymbol{c}_p} \tag{2.20}$$

Accordingly, the physically significant moments of the species distribution function, when using the species drift velocity as the reference velocity, are then calculated following:

Species-specific translational temperature:

$$\frac{3}{2} k_B T^{ss}_{tr,p} = \frac{1}{2} m_p \overline{c''^2_p} \tag{2.21}$$

Species-specific scalar pressure:

$$p_p^{ss} = \frac{1}{3} n_p m_p \overline{c_p''^2} \tag{2.22}$$

Species-specific pressure tensor:

$$\boldsymbol{p}_p^{ss} = n_p m_p \overline{\boldsymbol{c}_p'' \boldsymbol{c}_p''} \tag{2.23}$$

Species-specific stress tensor:

$$\boldsymbol{\tau}_p^{ss} \equiv \tau_{ij,p}^{ss} = -\left(n m_p \overline{c_{i,p}'' c_{j,p}''} - \delta_{ij} p_p^{ss} \right) \tag{2.24}$$

Species-specific heat flux vector:

$$\boldsymbol{q}_p^{ss} = \frac{1}{2} n_p m_p \overline{c_p''^2 \boldsymbol{c}_p''} \tag{2.25}$$

2.3 Surface Generation, Grid Generation and Grid Data Structure

Within the current implementation of the U-DSMC method local data structuring is required in order to carry out the base processes such as particle motion, particle collisions, and the sampling of macroscopic parameters. In order to define the data structure in standard DSMC codes a background grid is utilized. An unstructured Delaunay grid formed of tetrahedral elements is used in this code for the background structure in order to define the localization of key data components. The use of unstructured tetrahedral meshes enables effective control of nodal density throughout the computational domain in addition to the geometric flexibility needed to capture arbitrary flow geometries. The flow chart shown in Figure 4 illustrates the three basic procedures that may be used in order to create an unstructured grid file which is formatted for use with U-DSMC.

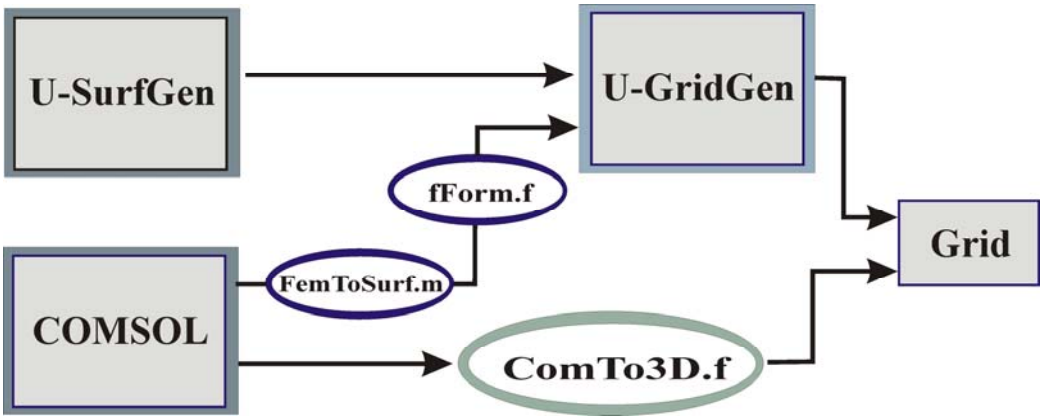


Figure 4. Methods of generating an unstructured grid for use with U-DSMC.

2.3.1 Surface Triangulation Methods

There are two methods of generating a suitable surface triangulation for use with the unstructured grid generation code U-GridGen. The first method utilizes a surface generation program written by Hammel (2002) referred to as U-SurfGen, while the second method makes use of the unstructured surface triangulation that bounds the tetrahedral mesh generated by the commercial software package COMSOL.

The first method utilized for generating a surface triangulation was developed by Kovalev (2000) and modified by Hammel (2002) into the current implementation of U-SurfGen. U-SurfGen was developed to handle axially symmetric objects as well as geometries that are definable by bi-linear elements. Two-dimensional topologies are constructed from the definition of control points which are connected by lines, arcs or parabolas. For axially symmetric geometries the topology generated from the control points are then tiled about the center axis analytically resulting in high-quality axisymmetric surfaces. Unfortunately, the program is restricted to full rotations and thus does not allow the user to capitalize on the symmetry of the axially symmetric flow domain. Bi-linear objects are created one planar face at a time. Examples of surface triangulations generated by U-SurfGen are shown in Figure 5. Although the

resulting surfaces from U-SurfGen are of sufficient quality, the process of defining the geometry is not end-user friendly. Additionally, the allowable node spacing is not sufficiently controllable as needed for complex flow fields.

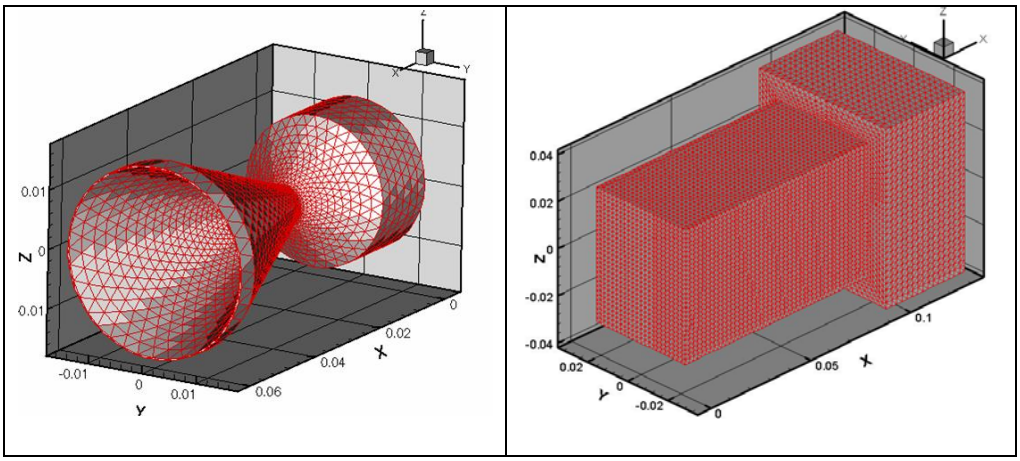


Figure 5. Examples of surface triangulations generated using the U-SurfGen surface generator.

The second method for generating a surface triangulation is accomplished using commercial software from COMSOL. Although COMSOL is a multiphysics package and is not a dedicated mesh generator, the geometric interface with CAD-type geometry entry is easy to use and the built in meshing parameters allow for precise control of the surface node spacing. Furthermore, the adaptation of a commercial software package allows for maximum geometric flexibility as compared to locally developed surface generation methods. The interface between COMSOL and U-GridGen is achieved through the use of two post processing scripts, given in Appendix B. These script files have been developed to import the resulting surface triangulation from COMSOL and format it for use in the U-GridGen mesh generator. The combination of the commercial surface generation and the U-GridGen mesh generator allows for extensive geometric modeling capabilities utilizing high-quality grids with acceptable generation times for

grids with less than 100,000 cells. Examples of surface triangulations generated by the COMSOL method are shown in Figure 6.

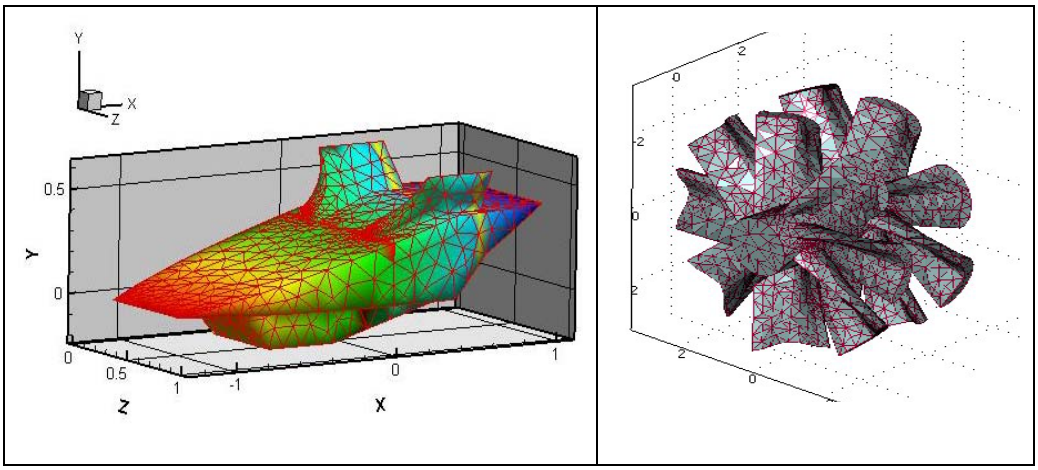


Figure 6. Examples of surface triangulations generated using COMSOL as a surface generator.

2.3.2 Unstructured Delaunay Grid Generation Methods and Grid Data Structure

There are two methods of generating an unstructured grid suitable for use with the U-DSMC code. The first method utilizes a grid generation program developed by Kovalev (2000) and modified by Hammel (2002), while the second method makes use of the unstructured mesh that underlies the commercial finite element software package COMSOL.

The first method of generating an unstructured Delaunay grid is by use of a mesh generation program developed by Kovalev (2000) called U-GridGen. Within the confines of this program the Delaunay grids are generated from a surface triangulation of the domain geometry using a formulation based on Watson's incremental node insertion method (Watson, 1981). The method of Watson is based on properties of Delaunay triangulations. The fundamental property of the Delaunay triangulation, as pertaining to grid quality, is that the nodes of all mesh elements will lie on a respective circumsphere, as illustrated in Figure 7. Further, the maximum radius of

the circumsphere may be chosen in order to improve the overall mesh quality. The underlying sizing function, which is defined by the surface triangulation, controls the node enrichment of the interior region of the computational domain. The method of Borouchaki and George (1997) has been extended to three dimensions and is used for interior node enrichment. Following this algorithm the characteristic distance between nodes is specified for each grid node. During the insertion algorithm every existing edge of the mesh is divided into a number of new prospective nodes such that the resulting edge segments vary gradually in length across the domain satisfying the surface triangulation spacing values.

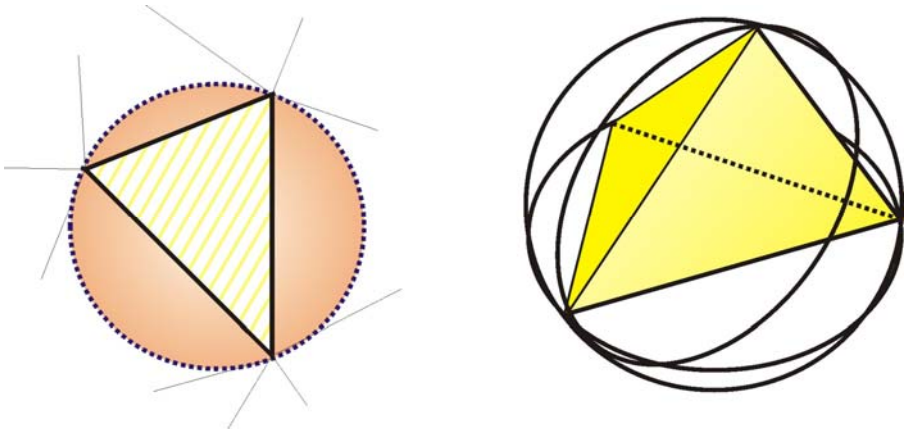


Figure 7. Delaunay elements lying within their respective circumsphere.

Prospective node insertion is filtered in order to satisfy the spacing criteria and ensure mesh quality. Nodes falling too close to existing nodes are not inserted, nor are nodes that worsen the local mesh quality. The measure of local mesh element quality is the dihedral angle. A user specified minimum is maintained as the cutoff value during insertion. The node is rejected if its insertion is calculated to create a tetrahedral element with a dihedral angle of less than the minimum. The nodes that are not rejected are inserted by means of Watson's algorithm.

Nodes are inserted until all spacing parameters are satisfied. The end result is a grid of high quality, however the success rate of the implementation of these algorithms is not acceptable for grids with more than 100,000 cells.

The second method for generating an unstructured three dimensional grid for use with the U-DSMC program is by using a tetrahedral Delaunay mesh generated by the commercial software package COMSOL. Although COMSOL is a multiphysics package and is not a dedicated mesh generator, the geometric interface allows for ease-of-use and the built in meshing parameters allow for precise control of the node spacing throughout the entirety of the flow domain. Furthermore, the adaptation of a commercial software package allows for significant decreases in grid generation time as well as a significant increase in the success rate incurred when generating grids with more than 100,000 cells, as compared to the U-GridGen program.

The interface between the COMSOL grid and the U-DSMC flow solver is developed by means of a short post processing script, given in Appendix B, which has been developed to reformat the COMSOL grid structure into a form acceptable for use in the U-DSMC solver. The combination of the commercial grid generator and the post processing script allows for extensive geometric modeling capabilities with increased cell-spacing control, reduced grid generation time, and an increased grid generation success rate. Examples of grids generated by the COMSOL method and U-Gridgen are shown in Figure 8.

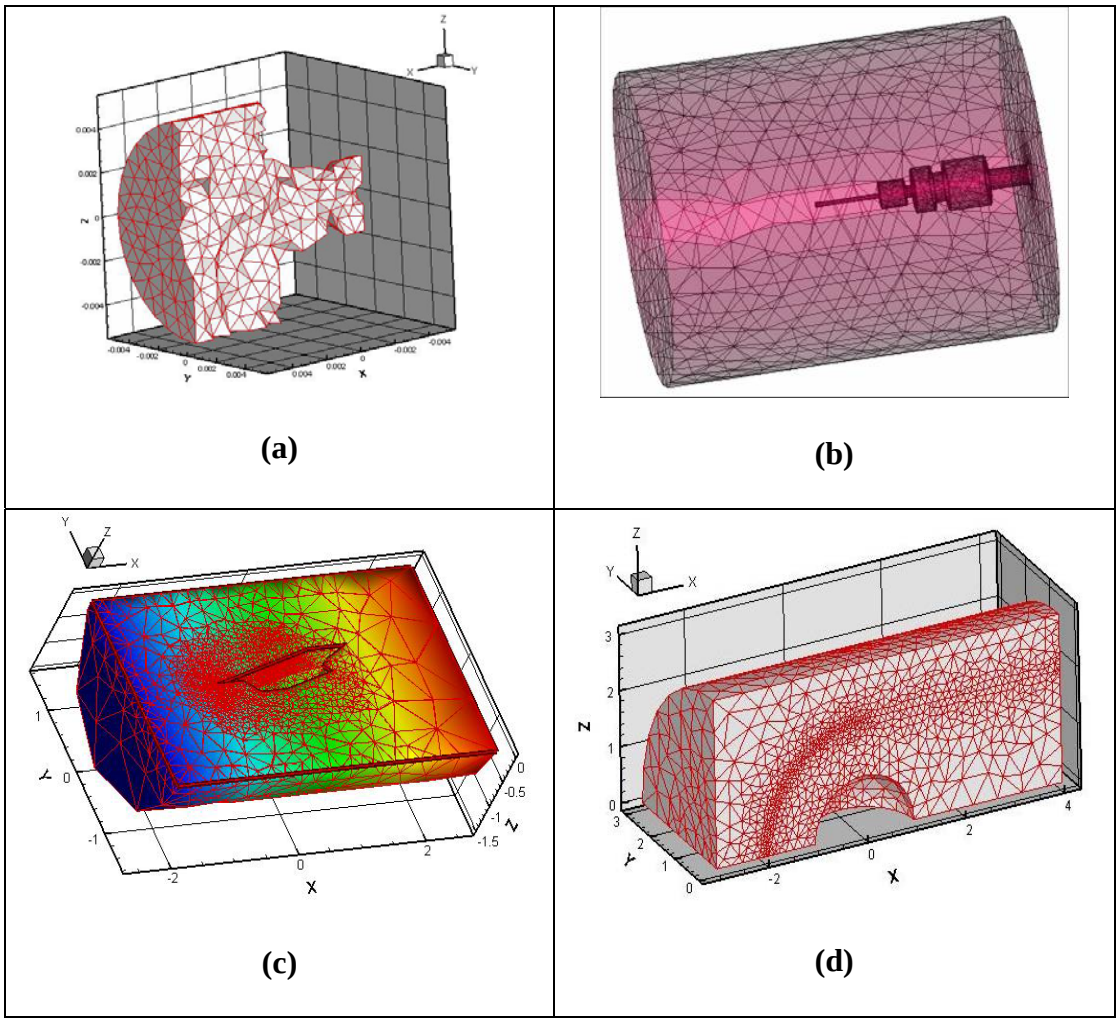


Figure 8. Examples of the geometric flexibility of the two grid generation methods, (a) and (b) generated using the U-GridGen mesh generation program, (c) and (d) generated using COMSOL.

For ease of computational manipulation the grid parameters are stored in a data structure that maintains node position, node connectivity, face sharing and cell nearest-neighbor information. The current data structure ensures minimal searching during particle motion and further lends itself to ease of parallelization. The tetrahedral cells are used in the code to define the volume of space where particle collisions are carried out. As such, a data structure that contains the particles local to each cell is maintained with indexing available to the cell's nodal, facial, and cell neighbor information. Since the grid cells are used for collision sampling the size

of the cell is maintained at a fraction of the local mean free path. Macroscopic flow sampling occurs over each cell and then a volume-weighted average is calculated at each nodal location. Indexing is maintained to ensure minimal computation during the sampling routine.

2.4 Particle Loading

At the onset of certain DSMC computations it is beneficial to insert an initial field of simulation particles. The population of the computational domain at the start of the simulation is typically referred to as loading. Loading is most often used as a means of reducing the computation time required to reach steady state. In U-DSMC, loading is typically carried out by placing the particles in randomly chosen positions fitting a specified overall velocity distribution function for each species. The standard velocity distribution function applied is the equilibrium or Maxwellian distribution:

$$f_{0,p} = \frac{\beta_p^{\,3}}{\pi^{3/2}} \exp\left(-\beta_p^{\,2} \mathbf{c}_p^{\,"\,2}\right) \tag{2.26}$$

where $\beta_p = \left\{ m_p / \left(2k_B T_p\right) \right\}^{1/2}$ and $\mathbf{c}_p^{"}$ is the species-specific thermal velocity. From the user defined global values of species number density (n_p), temperature (T_p), and drift velocity ($\mathbf{c}_{0,p}$), noting that $\mathbf{c}_p^{''} = \mathbf{c}_p - \mathbf{c}_{0,p}$, the distribution function above can be sampled to determine the molecular velocity components of each loaded particle. Details on sampling from a distribution function are given in Appendix A.

The unstructured nature of the computational domain in U-DSMC creates the need for localized loading. As such the global parameters for each species are used to determine the number of particles to load in each cell. This number is calculated from the specified number

density, the chosen computation weight, and the volume of the cell. Once this number is obtained, the random determination of the particle's position is carried out using a local vector coordinate system based on cell edges and corresponding limiting edge lengths. An illustration of the relevant geometry used in determining the particle's position is given in Figure 9.

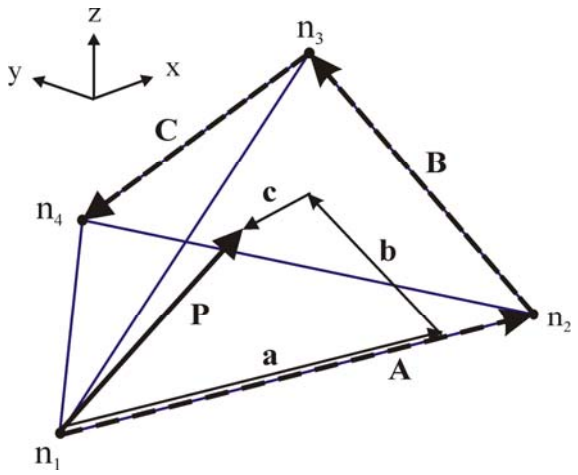


Figure 9. Particle position calculation geometry used in the loading algorithm.

A position vector, $\mathbf{P}$, is generated from three sub-vectors of random length which run along three cell edges following:

$$\mathbf{P} = \mathbf{a} + \mathbf{b} + \mathbf{c} \tag{2.27}$$

where the sub-vectors are determined from:

$$\begin{aligned} \mathbf{a} &= R_1^{1/3} \cdot \mathbf{A} \\ \mathbf{b} &= R_2^{1/2} \cdot R_1^{1/3} \cdot \mathbf{B} \\ \mathbf{c} &= R_3 \cdot R_2^{1/2} \cdot R_1^{1/3} \cdot \mathbf{C} \end{aligned} \tag{2.28}$$

where R_1, R_2 and R_3 are random fractions between 0 and 1. From vector **P** the particle position can be determined in Cartesian coordinates by

$$r^{p}{}_{i} = r^{n_1}{}_{i} + P_i \tag{2.29}$$

where $r^{p}{}_{i}$ is the particle's position and $r^{n_1}{}_{i}$ is node one's position. It should be noted that even simple calculations like those above can become ill suited for highly skewed cells. Therefore cell quality is of the utmost importance. Molecular velocity components are then calculated according to standard distribution sampling methods given in Appendix A.

2.5 Particle Injection

Flow boundaries are handled in U-DSMC by the application of molecular fluxes at specified surfaces. The introduction of particles into the computational domain is referred to as injection. Following the derivation by Bird (1994) the inward number flux $\dot{N}_p$ of species p can be defined by integration of the distribution function:

$$\dot{N}_p = n_p \frac{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{0}^{+\infty} u_p\, f_p \mathrm{d}u_p \mathrm{d}v_p \mathrm{d}w_p}{\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_p \mathrm{d}u_p \mathrm{d}v_p \mathrm{d}w_p} \tag{2.30}$$

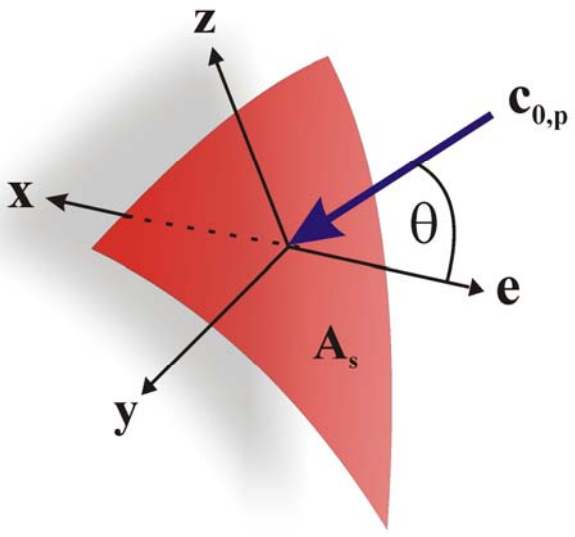


Figure 10. Local coordinate system for the analysis of molecular flux

For a general case, the flux of particles across a surface may occur such that the species mean flow velocity $\mathbf{c}_{0,p}$ is inclined at an angle θ to the unit normal vector $\mathbf{e}$ of a given surface, as shown in Figure 10. In an arbitrary coordinate system, such that the mean flow velocity lies in the x-y plane and the surface element lies in the y-z plane with the x-axis aligned in the negative $\mathbf{e}$ direction, the particle velocity can be expressed in terms of the species mean flow velocity and the species-specific thermal molecular velocity, denoted by $''$, as follows:

$$\begin{aligned} u_p &= u_p'' + c_{0,p}\cos(\theta) \\ v_p &= v_p'' + c_{0,p}\sin(\theta) \\ w_p &= w_p'' \end{aligned} \tag{2.31}$$

From these expressions the inward number flux can be written as

$$\dot{N}_p = \frac{n_p \beta_p^{\,3}}{\pi^{3/2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-c_{0,p}\cos(\theta)}^{\infty} \left(u_p'' + c_{0,p}\cos(\theta)\right)\exp\left(-\beta_p^{\,2}\left(u_p''^2 + v_p''^2 + w_p''^2\right)\right) \mathrm{d}u_p\,\mathrm{d}v_p\,\mathrm{d}w_p \tag{2.32}$$

If the integration is carried out the resulting inward number flux is found to be

$$\dot{N}_p = \frac{n_p}{2\beta_p\sqrt{\pi}}\left(\exp\left(-S_p{}^2\cos^2(\theta)\right) + \sqrt{\pi}\, S_p \cos(\theta)\left\{1 + \mathrm{erf}\left(S_p\cos(\theta)\right)\right\}\right) \tag{2.33}$$

where $S_p = c_{0,p}\beta_p$ is the species-specific molecular speed ratio. The value of $\dot{N}_p$ can be interpreted as the number of gas molecules of species p crossing a unit surface element per unit time with species mean flow velocity $\mathbf{c}_{\mathbf{0},p}$. Thus the number of simulation particles to be added to the domain in a given time step, ΔN_p, is given by

$$\Delta N_p = \frac{\dot{N}_p}{F_N}\tau\, A_s \tag{2.34}$$

where F_N is the particle weight, τ is the time step and A_s is the area of the surface element.

The number of injected particles is calculated for each face comprising the injection surface using the local values of species temperature, number density, and drift velocity. Since the surface elements of an unstructured domain are also unstructured, the calculation of the injection position is carried out using localized coordinates generated from the face edges, as shown in Figure 11.

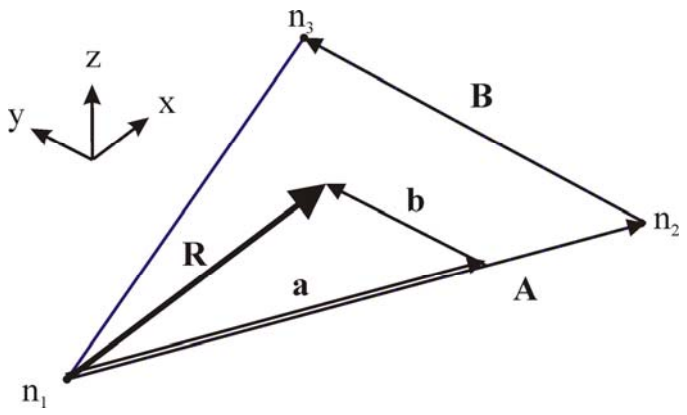


Figure 11. Particle position calculation geometry used in the injection algorithm

The determination of the particle's injection position is a random process bounded by the face edge lengths, very similar to that of loading. Using two face edges, a random component vector, $\mathbf{R}$, is generated from sub-vectors determined from that of the respective defining edge:

$$\begin{aligned} \mathbf{R} &= \mathbf{a} + \mathbf{b} \\ \mathbf{a} &= R_1^{1/2} \cdot \mathbf{A} \\ \mathbf{b} &= R_2 \cdot R_1^{1/2} \cdot \mathbf{B} \end{aligned} \tag{2.35}$$

where R_1 and R_2 are a random fractions between 0 and 1. From these vectors the intersection point for injection can be determined and the particle's position can be specified in Cartesian coordinates by $r_i^p = r_i^{n_1} + \mathrm{R}_i$, where $r^p{}_i$ is the particle's position and $r^{n_1}{}_i$ is node one's position. Once the particle's position has been determined the molecular velocity components are sampled from a drifting Maxwellian distribution by the standard methods outlined in Bird (1994) which are given in Appendix A.

2.5.1 Hypersonic Free Stream

One application of particle injection is that of modeling a hypersonic free stream. This occurs regularly in applications of U-DSMC to high-speed, high-altitude aircraft studies. The implementation of a hypersonic free stream follows the basis injection algorithm outlined above. In hypersonic injection the inlet flow parameters are assumed to be independent of the downstream flow conditions. Following this assumption, the parameters of the species specific free stream temperature, number density and drift velocity are chosen by the user and fixed throughout the simulation duration. The chosen values are applied across the entirety of the free stream surface, which is defined by the user using a face attribute tag in the gird generation process. The chosen parameters are then used for local determination of the species specific injection values, including the number of particles to inject, as well as the local temperature and drift velocity parameters to use when sampling the corresponding local distribution function. The initial implementation of this algorithm into the U-DSMC code was carried out by Hammel (2002). Algorithmic corrections and geometric generalizations were carried out in the current work to extend the validity and applicability of the previous implementation.

2.5.2 Subsonic Inflow

In subsonic internal or external flow scenarios the inlet or free stream parameters are affected by the downstream flow field. As such, the local values of pressure, temperature, number density and drift velocity can be functions of the downstream flow. Therefore the application of injection boundary conditions for subsonic flow cannot follow the fixed-parameter implementation used in the hypersonic free stream boundary condition. Instead, the boundary condition parameters must vary throughout the simulation time in a manner that captures the effect of the downstream flow field development on the upstream boundary. In order to meet

this need the implicit subsonic inlet boundary condition of Wang and Li (2004), which is based on the method of characteristics, has been incorporated into the U-DSMC code in the current work.

The implicit subsonic inlet methodology uses a first-order extrapolation based on the theory of characteristics to determine the local inlet drift velocity in the flow-parallel direction. Following the method of characteristics, the inlet pressure $\left(p_{in}\right)$ and temperature $\left(T_{in}\right)$ are independent of the downstream flow and are thus fixed throughout the simulation. If the boundary surface is chosen such that the flow-parallel direction is aligned with the x-axis then a general first order implicit extrapolation for the local velocity, such as that used in Fang and Liou (2002), may be written as:

$$\left(u_{in}\right)_j^k = u_j^k \tag{2.36}$$

where the subscript in references the inlet value for the boundary condition, the subscript j denotes the local face number and the superscript k denotes values computed at the k-th time step. However this simple implicit algorithm has been shown to require significant computational time to reach convergence and furthermore has been found to incur difficulties when the wall temperatures of internal flows are significantly different from the flow temperature. To correct these limitations the method of Wang and Li (2004) was derived from the method of characteristics and has been shown to improve convergence as well as handle significant inlet temperature gradients. Following the method of Wang and Li a general first-order extrapolation of the local stream-wise velocity for flow aligned with the x-axis may be written as:

$$\left(u_{in}\right)_j^k = u_j^k + \frac{p_{in} - p_j^k}{\rho_j^k a_j^k} \tag{2.37}$$

where $\left(u_{in}\right)_j^k$ is the calculated value for the local drift velocity for boundary face j, p_{in} is the user-specified inlet pressure (constant throughout the simulation), p_j, ρ_j and u_j are the cell values of pressure, density and x-direction drift velocity for the sample cell corresponding to boundary face j and a_j is the local speed of sound for the sample cell corresponding to boundary face j, which can be written as

$$a_j^k = \sqrt{\frac{2k_B T_j^k}{m}} \tag{2.38}$$

It should be noted that the extension of this method for multi-species gases is straight forward and would be accomplished by simply adding a secondary subscript to the relations given above. The x-axis aligned sampling of the local drift velocity using the method of characteristics based relations given above is shown schematically in Figure 12.

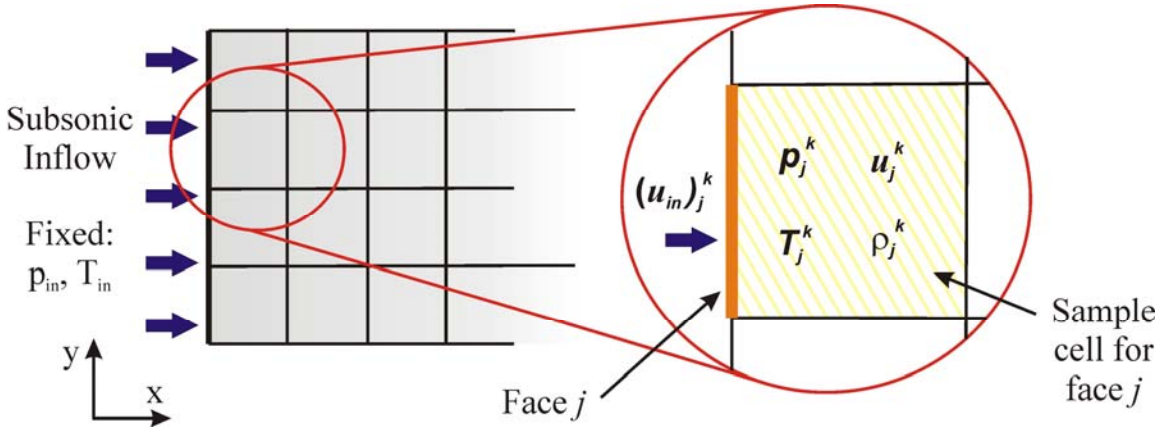


Figure 12. Illustration of x-axis aligned sampling of the local drift velocity using the method of characteristics on Cartesian grids.

Extension of the method of Wang and Li to U-DSMC has been carried out in this work. Several algorithmic modifications are required in order to handle the arbitrary orientation of the boundary faces with the stream direction that can occur within the confines of the unstructured grids. In order to generalize the method of Wang and Li, the relations given above have been mapped to local face-fitted coordinates. The implementation of the method of Wang and Li in U-DSMC also features additional boundary specification flexibility as well as an averaging technique used to limit fluctuations in the calculated cell values.

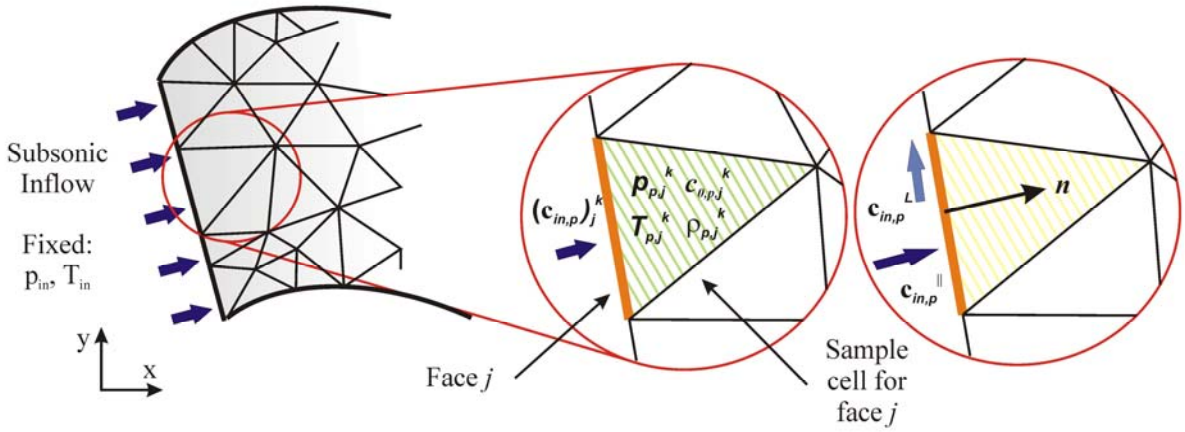


Figure 13. Illustration of surface normal aligned sampling of the local drift velocity based on the method of characteristics on unstructured grids.

For pressure driven flows within the unstructured subsonic inflow implementation a number of assumptions must be made. The first two assumptions define the orientation between the local face normal and the pressure gradient and are most applicable to internal flow. The relevant geometry is shown in Figure 13. The assumptions require that the pressure in the inward face normal direction is decreasing while the pressure along the surface plane is constant.

This is true for most pressure driven flow provided the injection surface is aligned properly. The pressure gradient-face normal alignment assumptions are given as:

$$\frac{\partial p}{\partial n_{\parallel}} < 0 \tag{2.39}$$

$$\frac{\partial p}{\partial n_{\perp}} = 0 \tag{2.40}$$

In order to satisfy these assumptions the user must define the inflow boundary surface such that it is perpendicular to the local pressure gradient. Due to the extensive geometric flexibility and ease-of-use of the modified grid generation methods, this restriction still yields expansive geometric flexibility and is not viewed as a significant limitation to the U-DSMC subsonic modeling capabilities. The second assumption follows from the first and is in regards to alignment of species-specific drift velocity with respect to the local surface normal vector. Following the gradient alignment assumption, a resulting restriction on the species-specific drift velocity orientation is:

$$\boldsymbol{n} \cdot \boldsymbol{c}^{\parallel}_{0,p,in} > 0 \tag{2.41}$$

This assumption is a direct physical result of the pressure gradient alignment assumption. The final assumption underlying the implementation of unstructured subsonic inflow is with regards to the species-specific drift velocity component perpendicular to the local surface normal. It is assumed that this component of the drift is zero and as such the perpendicular components of the injected particle's velocity may be sampled directly from a non-drifting Maxwellian distribution. The assumption can be written as:

$$c^{\perp}_{0,p,in} = 0 \tag{2.42}$$

It should be noted that for subsonic inflow boundaries applied in regions with zero pressure gradient the assumptions above are not needed.

The current implementation of the unstructured subsonic inflow boundary condition consists of the following steps applied to each species at each face composing the inlet surface:

Step 1. Determine the local face normal vector $\boldsymbol{n}$ from:

$$\boldsymbol{n} = \frac{\boldsymbol{r}_{12} \times \boldsymbol{r}_{13}}{|| \boldsymbol{r}_{12} \times \boldsymbol{r}_{13} ||} \tag{2.43}$$

where $\boldsymbol{r}_{12}$ is the vector from node 1 to node 2 and $\boldsymbol{r}_{13}$ is the vector from node 1 to node 3, as shown in Figure 14. Once calculated, the face normal is stored for future use.

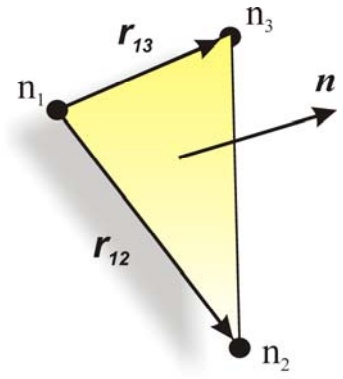


Figure 14. Geometry utilized in the calculation of local face normal vectors.

Step 2. Determine the sample cell for the current face, calculated once per face and stored. First calculate the sampling location from the user defined species-specific values for $p_{in,p}$, $T_{in,p}$ and $\lambda_{coef,p}$. Where $\lambda_{coef,p}$ is a user specified input used to control the location of the sampling cell within the domain with respect to the current face location. The value of $\lambda_{coef,p}$ can be adjusted in order to reduce the convergence time of the subsonic inflow boundaries or to adjust the

placement of the sample cell when localized grid spacing at the inlet surface is significantly smaller than required to obtain satisfactory downstream data. The calculation of the sample cell location x_j^c for face j follows:

$$x_j^c = x_j^f + \lambda_{coef,p}\,\lambda_p\,\boldsymbol{n} \tag{2.44}$$

where

$$\lambda_p = \frac{k_B T_{in,p}}{\sqrt{2}\pi p_{in,p} d_{ref,p}^2} \tag{2.45}$$

The relevant geometry is shown in Figure 15 for a general single-species case where the sample cell may not be the face cell. Once the sample cell location is calculated the sample cell number is determined by looping over the local cells and calculating if the sample location resides within each cell.

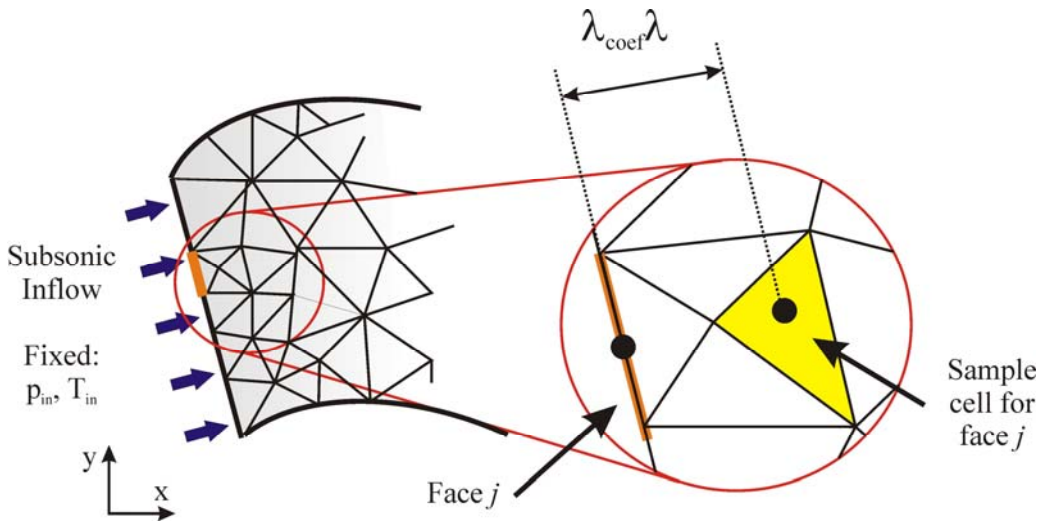


Figure 15. General geometry used to determine the sample cell for face *j*.

Step 3. During each iteration, the required species-specific parameters within the sample cell for face j are calculated following:

$$n_{p,j}^{k} = \frac{\sum_{i=1}^{N_{p,j}} F_N}{V_j} \tag{2.46}$$

$$\left(\boldsymbol{c}_{0,p}\right)_j^k = \frac{1}{N_{p,j}} \sum_{i=1}^{N_{p,j}} \boldsymbol{c}_p \tag{2.47}$$

$$\left(\overline{c_p''^2}\right)_j^k = \frac{1}{N_{p,j}} \sum_{i=1}^{N_{p,j}} \left[c_p - \left(c_{0,p}\right)_j^k\right]^2 \tag{2.48}$$

$$p_{p,j}^{k} = \frac{1}{3} m_p \, n_{p,j}^{k} \left(\overline{c_p''^2}\right)_j^k \tag{2.49}$$

$$T_{p,j}^{k} = \frac{p_{p,j}^{k}}{n_{p,j}^{k} k_B} \tag{2.50}$$

$$a_{p,j}^{k} = \sqrt{\frac{2 k_B T_{p,j}^{k}}{m_p}} \tag{2.51}$$

where V_j is the volume of the current sample cell, F_N is the number of real molecules represented by each simulated particle and m_p is the mass for the species under consideration. It should be noted that the calculations in this step are carried out one species at a time.

Step 4. Following the calculations of the sample cell parameters, the inlet species-specific drift velocity is calculated using corrections from the method of characteristics for pressure driven flows. In order to reduce oscillations in the inlet species-specific drift velocity that may be caused by statistical fluctuations in the sample cell data, a weighted average is applied when calculating the drift velocity for the face following:

$$\left(c_{0,p}^{||}\right)_j^k = \left(\boldsymbol{c}_{0,p}\right)_j^k \bullet \boldsymbol{n} \tag{2.52}$$

$$\left(c_{in,p}^{\|}\right)_j^{k^*} = \left(c_{0,p}^{\|}\right)_j^{k} + \frac{p_{in,p} - p_{p,j}^{k}}{m_p \, n_{p,j}^{k} \, a_{p,j}^{k}} \tag{2.53}$$

$$\left(c_{in,p}^{\|}\right)_j^{k} = \frac{1}{4}\left(c_{0,p}^{\|}\right)_j^{k^*} + \frac{3}{4}\left(c_{0,p}^{\|}\right)_j^{AVE} \tag{2.54}$$

Step 5. Once all the species-specific drift velocity for the face is determined the number of particles to be injected is calculated using Eq. (2.34). The corresponding velocity components of each injected particle can then be sampled from the velocity distribution functions following the methods described in Appendix A using $n_{in,p} = p_{in,p} \,/\, k_B T_{in,p}$, $T_{in,p}$ and $\left(c_{in,p}^{\|}\right)_j^{k}$.

2.5.3 Subsonic Outflow

In subsonic internal or external flow scenarios the upstream conditions are affected by the downstream flow field. As such, detailed capture of both the upstream inlet boundary conditions and downstream exit boundary conditions must be obtained to successfully model subsonic flow. Therefore the application of injection boundary conditions for subsonic flow cannot follow the 'vacuum' implementation that is typically utilized in most hypersonic DSMC studies conducted to date. Instead, the boundary condition parameters must vary throughout the simulation time in a manner that captures the effect of the upstream flow field development on the downstream boundary. In order to meet this need the downstream pressure condition correction equations proposed by Nance et al (1997) and extended by Liou and Fang (2000) which employ Whitfield's characteristic formulation (Whitfield and Janus, 1984) have been incorporated into the U-DSMC code in the current work.

Following the methodology outlined in Liou and Fang (2000), the correction equations of Nance et al (1997) for the downstream boundary of a flow aligned in the x-direction of a Cartesian grid system are given as:

$$(\rho_e)_j^k = \rho_j^k + \frac{p_e - p_j^k}{\left(a_j^k\right)^2} \tag{2.55}$$

which can also be written in terms of an exit number density as,

$$(n_e)_j^k = n_j^k + \frac{p_e - p_j^k}{m\left(a_j^k\right)^2} \tag{2.56}$$

with the exit velocities given by,

$$(u_e)_j^k = u_j^k + \frac{p_j^k - p_e}{\rho_j^k a_j^k} \tag{2.57}$$

$$\left(v_e\right)_j^k = v_j^k \tag{2.58}$$

and the exit temperature following,

$$\left(T_e\right)_j^k = \frac{p_e}{\left(n_e\right)_j^k k_B} \tag{2.59}$$

where the subscript e corresponds to exit boundary condition values, the subscript j denotes values obtained from the sample cell corresponding to boundary face j and the superscript k denotes values computed at the k-th time step. The sample cell values of density and pressure are obtained from:

$$\rho_j^k = n_j^k m \tag{2.60}$$

$$p_j^k = n_j^k k_B T_j^k \tag{2.61}$$

where n_j^k and T_j^k are obtained through sampling the particles in the sample cell. Figure 16 illustrates the geometric relations of the correction equations as pertaining to structured grids.

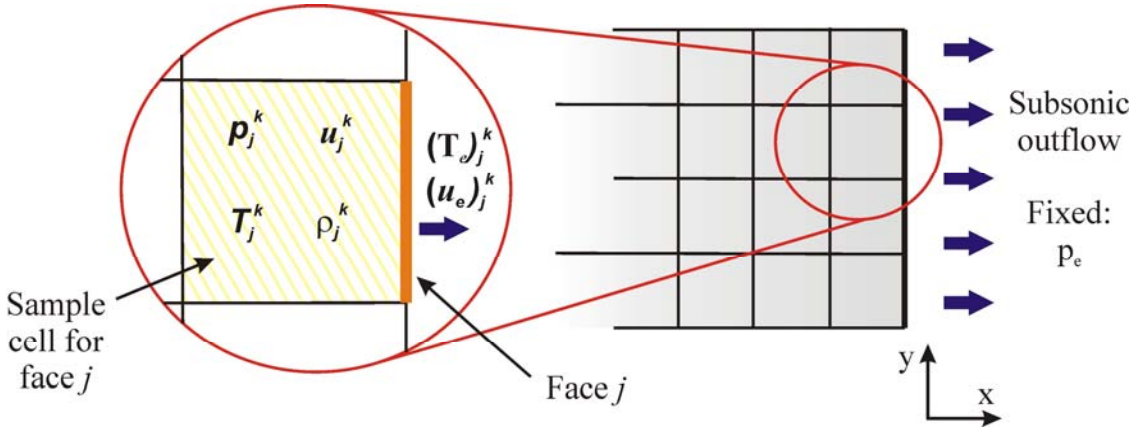


Figure 16. Illustration of x-axis aligned sampling of the local exit drift velocity using the method of characteristics on Cartesian grids.

Extension of the method of Nance et al (1997) to unstructured DSMC has been carried out in this work. Algorithmic modifications for handling the arbitrary orientation of the boundary faces with respect to the stream direction within the confines of the unstructured grid have been made. In order to generalize the implementation of Liou and Fang (2000), the relations given above have been mapped to local face-fitted coordinates. Similarly to the unstructured subsonic inflow boundary conditions, the implementation of the method of Nance et al (1997) in U-DSMC also features additional boundary specification flexibility as well as an averaging technique used to limit fluctuations in the calculated cell values. A schematic illustrating the parameters involved in the unstructured subsonic outflow boundary are shown in Figure 17.

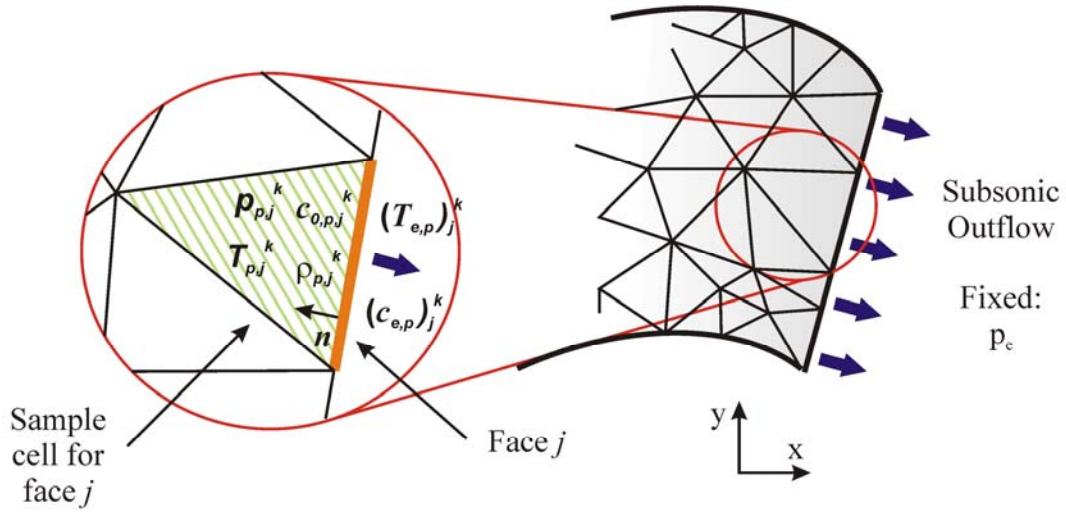


Figure 17. Illustration of surface normal aligned sampling of the local exit drift velocity based on the method of characteristics as applied to unstructured grids.

For pressure driven flows the unstructured subsonic outflow implementation follows the same assumptions underlying the subsonic inflow boundary condition. Namely, the pressure gradient alignment and the corresponding species-specific drift velocity components must follow:

$$\frac{\partial p}{\partial n_{\parallel}} < 0 \tag{2.62}$$

$$\frac{\partial p}{\partial n_{\perp}} = 0 \tag{2.63}$$

$$\boldsymbol{n} \cdot \boldsymbol{c}^{\parallel}_{0,p,in} < 0 \tag{2.64}$$

$$c^{\perp}_{0,p,in} = 0 \tag{2.65}$$

The current implementation of the unstructured subsonic outflow boundary condition consists of the following steps applied to each species over each face composing the inlet surface:

Step 1. Determine the local face normal vector $\boldsymbol{n}$ from:

$$\boldsymbol{n} = \frac{\boldsymbol{r}_{12} \times \boldsymbol{r}_{13}}{|| \boldsymbol{r}_{12} \times \boldsymbol{r}_{13} ||} \tag{2.66}$$

where $\boldsymbol{r}_{12}$ is the vector from node 1 to node 2 and $\boldsymbol{r}_{13}$ is the vector from node 1 to node 3. Once calculated, the face normal is stored for future use.

Step 2. Determine the sample cell for the current face, calculated once per face and stored. First calculate the sampling location from the user defined values for species-specific exit pressure $p_{e,p}$, an initial guess of the exit temperature $T_{e,p}$ and the multiplicative factor $\lambda_{coef,p}$ following:

$$x_j^c = x_j^f + \lambda_{coef,p}\, \lambda_p\, \boldsymbol{n} \tag{2.67}$$

where

$$\lambda_p = \frac{k_B T_{e,p}}{\sqrt{2}\pi p_{e,p} d_{ref,p}^2} \tag{2.68}$$

Once the sample cell location is calculated the sample cell number is determined by looping over the local cells and calculating if the sample location resides within each cell. Once identified, the cells number is stored for future access.

Step 3. During each iteration, the required species-specific parameters within the sample cell for face j are calculated following:

$$n_{p,j}^k = \frac{\sum_{i=1}^{N_{p,j}} F_N}{V_j} \tag{2.69}$$

$$\left(\boldsymbol{c}_{0,p}\right)_j^k = \frac{1}{N_{p,j}} \sum_{i=1}^{N_{p,j}} \boldsymbol{c}_p \tag{2.70}$$

$$\left(\overline{c_p''^2}\right)_j^k = \frac{1}{N_{p,j}} \sum_{i=1}^{N_{p,j}} \left[c_p - \left(c_{0,p}\right)_j^k\right]^2 \tag{2.71}$$

$$p_{p,j}^k = \frac{1}{3} m_p \, n_{p,j}^k \left(\overline{c_p''^2}\right)_j^k \tag{2.72}$$

$$T_{p,j}^k = \frac{p_{p,j}^k}{n_{p,j}^k k_B} \tag{2.73}$$

$$a_{p,j}^k = \sqrt{\frac{2k_B T_{p,j}^k}{m_p}} \tag{2.74}$$

where V_j is the volume of the current sample cell, F_N is the number of real molecules represented by each simulated particle and m_p is the mass of the species p currently under consideration. Once again, it should be noted that the calculations in this step are carried out one species at a time.

Step 4. Following the calculations of the sample cell parameters, the outlet species-mean velocity $\left(c_{e,p}^{||}\right)_j^k$, number density $n_{e,p}^k$ and temperature $T_{e,p}^k$ are calculated, using corrections from the theory of characteristics for pressure driven flow. In order to reduce oscillations in the outlet drift velocity that may be caused by statistical fluctuations in the sample cell data weighted averages are applied following:

$$n_{e,p}^k = n_{p,j}^k + \frac{p_{e,p} - p_{p,j}^k}{m_p (a_{p,j}^k)^2} \tag{2.75}$$

$$T_{e,p}^k = \frac{p_{e,p}}{n_{e,p}^k \, k_B} \tag{2.76}$$

$$\left(c_{0,p}^{||}\right)_j^k = \left(\boldsymbol{c}_{0,p}\right)_j^k \bullet \boldsymbol{n} \tag{2.77}$$

$$\left(c_{0,p}^{||}\right)_j^{k^*} = \left(c_{0,p}^{||}\right)_j^k + \frac{p_{p,j}^k - p_{e,p}}{m_p\, n_{p,j}^k\, a_{p,j}^k} \tag{2.78}$$

$$\left(c_{0,p}^{||}\right)_j^k = \frac{1}{4}\left(c_{0,p}^{||}\right)_j^{k^*} + \frac{3}{4}\left(c_{0,p}^{||}\right)_j^{AVE} \tag{2.79}$$

Step 5. Once the exit drift velocity, number density and temperature for the face are determined the number of particles to be injected is calculated using Eq. (2.34). The corresponding velocity components of each injected particle can then be sampled from the velocity distribution functions following the methods described in Appendix A using $n_{e,p}^k$, $T_{e,p}^k$ and $\left(c_{e,p}^{||}\right)_j^k$.

2.6 Particle Motion

In a DSMC simulation the state of the system is given by the positions and velocities of the particles. These values can be specified as vectors, $\{\mathbf{r}_i, \mathbf{c}_i\}$, typically referred to collectively as phase space. Within a DSMC simulation the motion and collisions of particles are uncoupled, and as such the particle motion can be linked with the collisionless Boltzmann equation:

$$\frac{\partial(nf)}{\partial t} + \mathbf{c} \bullet \frac{\partial(nf)}{\partial \mathbf{r}} + \mathbf{F}_m \bullet \frac{\partial(nf)}{\partial \mathbf{c}} = 0 \tag{2.80}$$

from which the equations of motion are simply given by:

$$m\frac{d\mathbf{c}}{dt} = \mathbf{F} \tag{2.81}$$

$$\frac{d\mathbf{r}}{dt} = \mathbf{c} \tag{2.82}$$

Within a U-DSMC simulation, particle motion is carried out using a particle tracing technique. During motion, particles are assumed to move free of the influence of all other particles. A general description of particle motion within a cell can be given by $\boldsymbol{r} = \boldsymbol{r}_0 + \boldsymbol{c}\tau$, where $\mathbf{r}$ is the new position vector, $\mathbf{r}_0$ is the old position vector, $\mathbf{c}$ is the particle's velocity vector and τ is the time step and thus the elapsed time for motion.

When working with particle methods on unstructured tetrahedral meshes the governing algorithms of motion become a bit more complex. Although motion within the cell is relatively straight forward, capturing the transit of a particle from one cell to another requires special attention. Identifying the cell in which a particle currently resides as well as calculating the cell to which a particle will move into is an extremely computationally intensive procedure on unstructured grids. The current method used for capturing the motion of particles within the U-DSMC code is based upon the successive-neighbor methodology of Lohner and Ambrosiano (1990) and has been implemented in successive increments by Hammel (2002) and Spirkin (2006). Recently, within this work, minor modifications to supporting subroutines have been required in order to extend the geometric generality of the successive-neighbor algorithm.

The overall structure of the particle motion routine can be broken into a number of principle steps. Those steps are outlined below along with relevant details into the underlying implementation.

Step 1. The algorithm loops over all cells and each particle in each cell. For each particle, the new position of the particle, assuming free motion without surface interaction, is calculated following:

$$\boldsymbol{r}_f = \boldsymbol{r}_i + \boldsymbol{c}\tau \tag{2.83}$$

Step 2. The tracing algorithm considers each face of the current cell in turn in order to determine the probability that the particle of interest has left the current cell. In order to reduce the computational requirements of particle tracking a series of pre-tests are calculated to determine if the current face may have been crossed.

Test 1. The first pre-test is a simple assessment to determine if the new particle position is in the cell inward or cell outward direction with respect to the current face. Figure 18 illustrates the geometry involved as well as the possible scenarios. Using the face normal $\boldsymbol{n}$, along with its predetermined orientation, the projection of the particle's ray of motion $\boldsymbol{r}_{proj}$ can be generated following:

$$r_{proj} = \boldsymbol{r}_{f1} \bullet \boldsymbol{n} \tag{2.84}$$

If $\boldsymbol{r}_{proj} > 0$ then the particle has not crossed the plane of the face and therefore intersection is not possible. If $\boldsymbol{r}_{proj} < 0$ then intersection is possible. However, due to the nature of the tetrahedral cells it is also possible that the intersection occurred through a neighboring face. Therefore a second test must be performed.

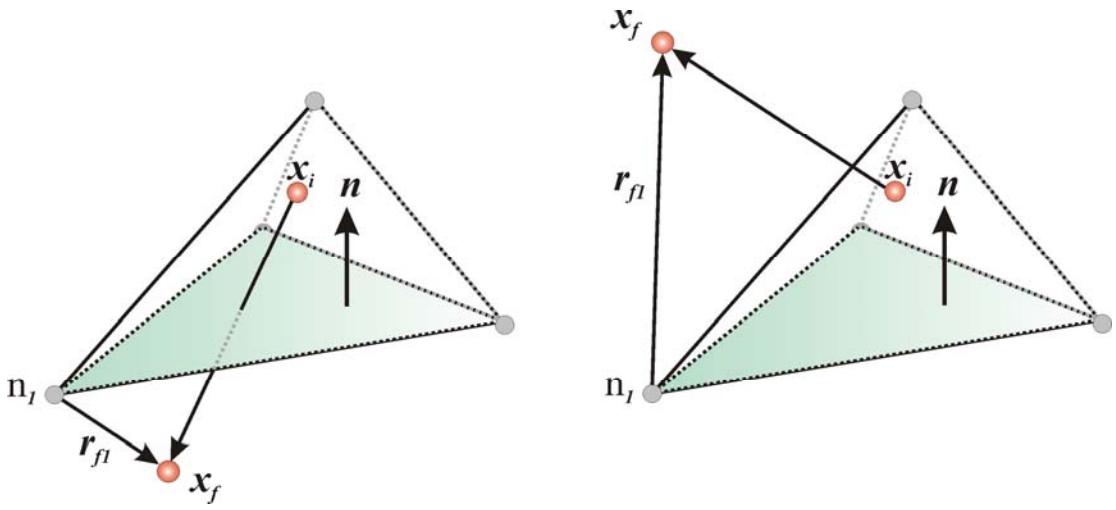


Figure 18. Geometry utilized to determine cell inward or cell outward directionality with respect to the current face.

Test 2. The second test is applied to cases that pass the first test. In order to determine the likely hood that the particle intersected a face a volume-weighted function is generated for each face following:

$$N_1\left(\boldsymbol{r}_p(t+dt)\right)=\frac{\Omega_{p234}}{\Omega_{1234}}=\frac{\frac{1}{6}\left[\boldsymbol{r}_{24}\bullet\left(\boldsymbol{r}_{2p}\times\boldsymbol{r}_{23}\right)\right]}{\frac{1}{6}\left[\boldsymbol{r}_{14}\bullet\left(\boldsymbol{r}_{12}\times\boldsymbol{r}_{23}\right)\right]} \tag{2.85}$$

$$N_2\left(\boldsymbol{r}_p(t+dt)\right)=\frac{\Omega_{p134}}{\Omega_{1234}} \tag{2.86}$$

$$N_3\left(\boldsymbol{r}_p(t+dt)\right)=\frac{\Omega_{p124}}{\Omega_{1234}} \tag{2.87}$$

$$N_4\left(\boldsymbol{r}_p(t+dt)\right)=\frac{\Omega_{p123}}{\Omega_{1234}} \tag{2.88}$$

A schematic illustrating the geometric physicality of the volume-weighted function is shown in Figure 19. The volume-weighted functions are evaluated with regard to the maximum value of 1. If $N_1 + N_2 + N_3 + N_4 < 1$ then the particle continues to reside within the current cell. However, if $N_1 + N_2 + N_3 + N_4 > 1$ then the particle has left the cell. From an analysis of face values of N the face which is intersected can be determined. If it has been determined that the current particle has left the cell than the details of the cell transit must be obtained in order to trace the particle's new position and cell owner.

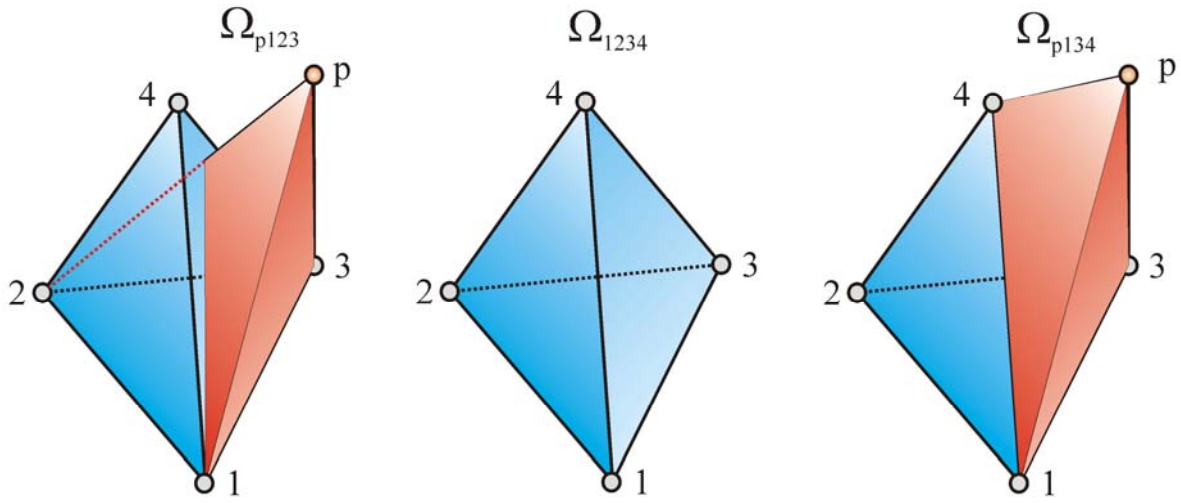


Figure 19. Illustration of the geometric physicality of the volume-weighted functions.

Step 3. If it has been determined that the particle has exited the current cell then the details of the respective face transit must be determined. To determine the details of the transit the intersection of a particle with one of the cell-edge planes is expressed as a system of linear equations. The parameters of these equations are the particle's position and velocity and also the geometric information regarding two edges of the cell face plane. From these equations the time

of intersection along with the point of intersection with a given face can be extracted. The sketch shown in Figure 20 illustrates the geometry involved.

As shown in Figure 20, the intersection of a particle with initial position $\mathbf{r}_0$ and velocity $\mathbf{c}$ with face ABC is given by:

$$\mathbf{r}_0 + \mathbf{c}\Delta t = \alpha_1 \mathbf{AB} + \alpha_2 \mathbf{AC} \tag{2.89}$$

where **AB** and **AC** are the vectors from point A to point B and from point A to point C respectively, Δt is the time elapsed in moving from the initial point to the point of intersection with the plane defined by points A, B, and C. The parameters α_1 and α_2 define the point of intersection in the skewed coordinate system of face ABC.

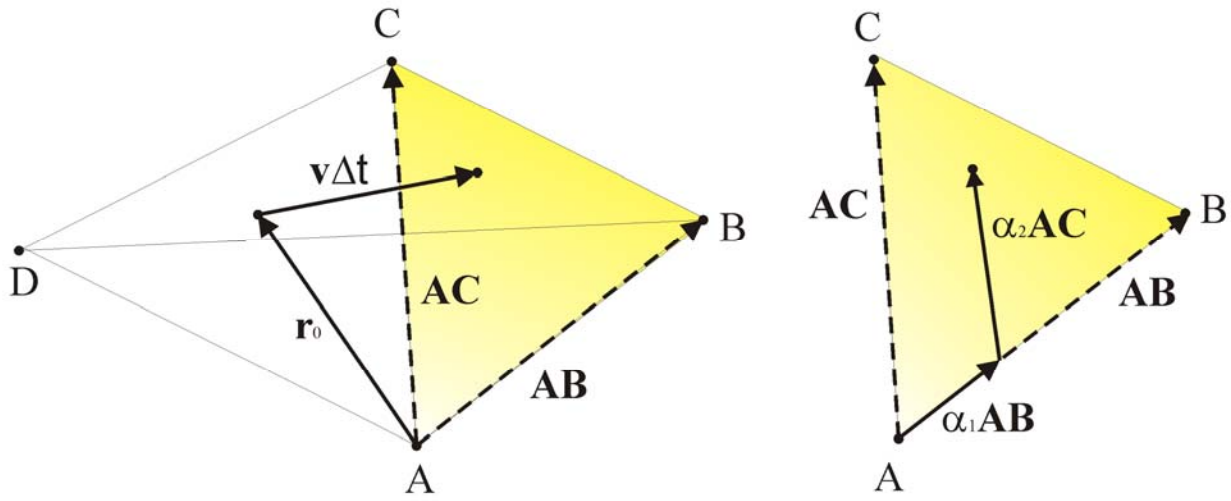


Figure 20. Particle-face intersection geometry used in particle motion algorithm.

Once a solution has been reached the values of Δt, α_1, and α_2 are analyzed. If Δt is negative then intersection with face ABC does not occur. If α_1 or α_2 are less than zero or greater than unity then the intersection with the plane defined by A, B, and C occurs outside of

face ABC. Since the linear system of equations may be ill-conditioned if the cells and corresponding faces are poorly shaped, great care must be taken when constructing a computational mesh. If the faces are poorly shaped, usually defined by a very small dihedral angle, the solution of the linear equations become unstable and particles can get 'lost', which is a term applied when a particle is without a cell owner. If the particle does intersect a face, then the cell ownership of the particle is updated accordingly as is the new particle location.

2.7 Collisions

The modeling of collisions in U-DSMC captures the collision effects seen in the right hand side of the Boltzmann equation:

$$\frac{\partial}{\partial t}\left(n_p f_p\right) = \sum_{q=1}^{s} \int_{-\infty}^{+\infty} \int_{0}^{4\pi} n_p n_q \left(f_p^* f_q^* - f_p f_q \right) c_{r,pq} \sigma_{\mathbf{T}pq} \mathrm{d}\Omega \mathrm{d}\mathbf{c}_q \qquad (2.90)$$

The collision procedure is applied in every cell of the computational domain independently. The procedure assumes a uniform distribution of particles throughout the cell volume. The complex collision processes that would occur between real gas molecules are substituted by stochastic interactions of model particles, where collision pairs are chosen irregardless of their positions inside the cell. Furthermore, the collision does not change the position vector of either particle.

When two real molecules collide in nature the resultant collision mechanics are complex functions of an interaction potential whose roots lie in the quantum mechanics realm (Bergemann and Boyd, 1994, Haas et al, 1994). Although the modeling of each individual collision would thus require complex algorithms to capture the underlying physics the stochastic nature of the DSMC method allows for a significantly simplified phenomenological approach that treats the bulk effect of the underlying quantum relations. Within this simplified

methodology two key collision divisions exist, namely elastic collisions and inelastic collisions. The applicability of each is dependent upon the molecular model chosen and the implementation of each is described in the following discussions.

2.7.1 Elastic Collisions: Mathematical Model

In elastic binary collisions both linear momentum and energy must be conserved, therefore:

$$\begin{aligned} m_1 c_1^{\,2} + m_2 c_2^{\,2} &= m_1 \left(c_1^{\,*}\right)^2 + m_2 \left(c_2^{\,*}\right)^2 \\ m_1 \mathbf{c}_1 + m_2 \mathbf{c}_2 &= m_1 \mathbf{c}_1^{\,*} + m_2 \mathbf{c}_2^{\,*} \end{aligned} \tag{2.91}$$

where m is the particle mass and $\mathbf{c}$ is the particle velocity, the subscripts denote particles 1 and 2, the superscript * denotes post-collision values. In the center of mass reference frame, which is moving with velocity $\mathbf{c}_{cm}$, the particle velocities are given by:

$$\begin{aligned} \mathbf{c_1} &= \mathbf{c}_{cm} + \frac{m_2}{m_1 + m_2} \mathbf{c}_r \\ \mathbf{c_2} &= \mathbf{c}_{cm} + \frac{m_1}{m_1 + m_2} \mathbf{c}_r \end{aligned} \tag{2.92}$$

with,

$$\mathbf{c}_{cm} = \frac{m_1 \mathbf{c_1} + m_2 \mathbf{c_2}}{m_1 + m_2} \tag{2.93}$$

and

$$\mathbf{c}_r = \mathbf{c_1} - \mathbf{c_2} \tag{2.94}$$

where $\mathbf{c}_r$ is the relative velocity between the two particles. The collision occurs in the plane defined by the two velocities and the collision dynamics can be characterized by the schematic shown in Figure 21.

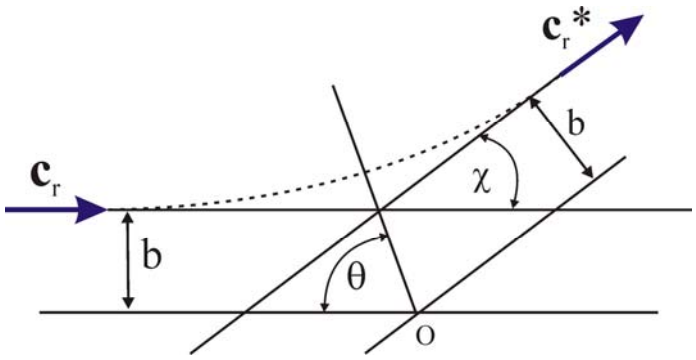


Figure 21. Illustration of the impact parameters used in binary collisions (adapted from Bird, 1994).

Applying a local coordinate system, $\{x', y', z'\}$ that aligns the x'-axis with the pre-collision relative velocity $\mathbf{c}_r$, the components of the post-collision relative velocity $\mathbf{c}^*_r$ become:

$$\begin{aligned} c^*_{r,x'} &= c_r \cos(\chi) \\ c^*_{r,y'} &= c_r \sin(\chi)\cos(\varepsilon) \\ c^*_{r,z'} &= c_r \sin(\chi)\sin(\varepsilon) \end{aligned} \tag{2.95}$$

where χ is the scattering angle and ε is the angle between the collision-plane and the absolute x-y plane.

Once the post-collision relative velocity is obtained in the local coordinate system a transformation can be carried out to obtain the post-collision relative velocity components in the absolute reference frame following the determination of χ and ε. Therefore, apart from the translational velocities of the two particles undergoing collision, just two parameters, called the *impact parameters*, are required to completely specify a binary elastic collision.

The first parameter required is the distance of closest approach, b, as seen in Figure 21. The second parameter required is the angle ε. These two parameters can be related to the scattering angle χ by means of a differential cross section $\sigma d\Omega$ through the following relations:

$$\sigma \, d\Omega = b \, db \, d\varepsilon \tag{2.96}$$

where,

$$d\Omega = \sin\chi \, d\chi \, d\varepsilon \tag{2.97}$$

yielding:

$$\sigma = \frac{b}{\sin\chi}\left|\frac{db}{d\chi}\right|. \tag{2.98}$$

From these relations the *total collision cross-section* σ_T is defined as:

$$\sigma_T = \int_0^{4\pi} \sigma \, d\Omega = 2\pi \int_0^{\pi} \sigma \, sin\chi \, d\chi \tag{2.99}$$

The formulation of σ_T is dependant upon the molecular model, however, once a model is chosen the integral above is specified. Upon specification the two collision parameters can be defined and the post-collision velocities can be calculated. Within the current implementation molecular cross sections may be modeled as either Hard Sphere (HS) or Variable Hard Sphere (VHS). Details for determining σ_T for the two collision models implemented in U-DSMC are given below.

In general the total collision cross section of an arbitrary collision is given as:

$$\sigma_T = \pi d^2 \tag{2.100}$$

where d is the distance between the centers of the molecules' effective potential sphere. A drawing illustrating the interaction distance is given in Figure 22.

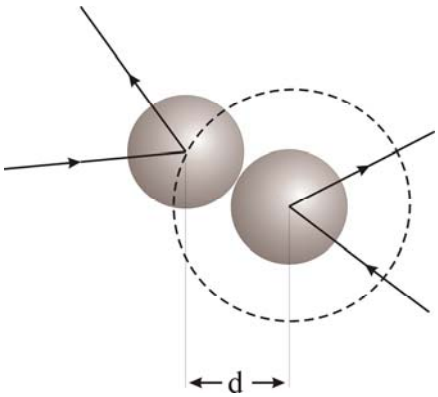


Figure 22. Illustration of the interaction distance d (adapted from Bird, 1994)

Within the confines of the hard sphere model the complex nature of the true molecular interaction potential is simplified such that the interactive force becomes effective at a distance equal to the average of the two molecules' effective diameters:

$$d = \frac{d_1 + d_2}{2} = d_{12} \tag{2.101}$$

from which the parameter b can be calculated as,

$$b = d_{12} \sin \theta_A = d_{12} \sin\left(\chi/2\right). \tag{2.102}$$

giving,

$$\left| db/d\chi \right| = \tfrac{1}{2} d_{12} \sin\left(\tfrac{1}{2}\chi\right) \tag{2.103}$$

yielding:

$$\sigma = \frac{d^2_{12}}{4} \tag{2.104}$$

From the relation for σ given above it can be seen that the collision cross section for a hard sphere molecule is independent of χ and thus the scattering that occurs is isotropic in the center

of mass reference frame. This in turn indicates that all scattering angles are equally probable. The total collision cross section for the hard sphere model is given by:

$$\sigma_T = \int_0^{4\pi} \sigma \, d\Omega = \pi \, d^2_{12} \tag{2.105}$$

which is used in calculating the number of collisions that occur in each cell for each time step as discussed below.

Although the hard sphere model benefits from the ease of calculation afforded by its isotropic scattering, the hard sphere model does not capture the translational energy dependence of the collision cross section. To compensate for this factor Bird (1981) developed the variable hard sphere model. Within the confines of the variable hard sphere model the collision cross section is taken to be a function of the relative velocity of the collision partners as well as the temperature exponent of the coefficient of viscosity. The effective diameter is given as:

$$d = d_{ref} \left(v_{rel,ref} / v_{rel} \right)^{\omega} \tag{2.106}$$

which yields,

$$\chi = 2 \cos^{-1} \left(b / d \right) \tag{2.107}$$

with,

$$\sigma_T = \int_0^{4\pi} \sigma \, d\Omega = \pi \, d^2_{12} = \frac{\pi}{4} \left[d_{1,ref} \left(v_{1,ref} / v_{rel} \right)^{\omega_1} + d_{2,ref} \left(v_{2,ref} / v_{rel} \right)^{\omega_2} \right] \tag{2.108}$$

2.7.2 Elastic Collisions: U-DSMC Implementation

The implementation of the elastic collision algorithm within U-DSMC follows the standard methods of Bird (1994). The Delaunay tetrahedral cells are taken as the collision volume and as such local cell spacing is restricted to a fraction of the local mean free path.

Furthermore, due to the current implementation of macroscopic sampling, the Delaunay cells can be viewed as collision-sub cells of a larger macroscopic sampling cell. Further details on this principle are given in Section 2.9. The collision algorithm is applied one cell at a time until all cells are treated. Figure 23 illustrates the relevant collision volume used in U-DSMC. The current implementation of collisions in U-DSMC has been developed in the current work by means of a rewrite of the implementation of Hammel (2002). Due to the condition of the past implementation upon inheritance by the current investigator, only three supporting subroutines from the previous implementation could be reused.

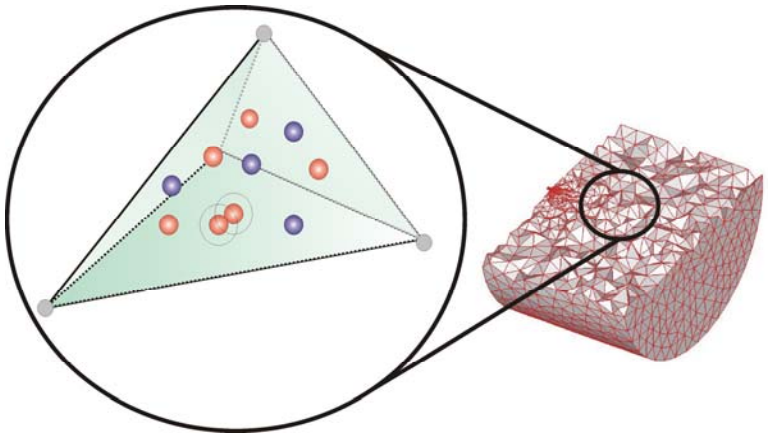


Figure 23. Delaunay cell used as collision volume in U-DSMC.

The implementation of collisions in the U-DSMC code follows the unstructured Delaunay no-time counter collision sampling scheme (UD-NTC) which is an extraction of the original no-time counter scheme of Bird (1994). The steps involved in calculating elastic collisions using the UD-NTC method are applied to every Delaunay cell in the computational domain and follow the algorithm given below:

Step 1. Calculate the number of particles in the Delaunay cell by cycling over the cell's particle stack:

$$N_c = \sum_{i=1}^{stack\_last} 1 \tag{2.109}$$

Step 2. Evaluate $\left(\sigma_T c_r\right)_{MAX}$ for the current cell. This calculation is done once at initialization and then the value of $\left(\sigma_T c_r\right)_{MAX}$ is updated on a cell by cell basis as larger values of $\left(\sigma_T c_r\right)_{MAX}$ are encountered in Step 4. The $\left(\sigma_T c_r\right)_{MAX}$ term captures the maximum value, when applied to all particles in the cell, of the total collision cross section and the relative velocity of a collision pair.

Step 3. Calculate the number of collision pairs for the current cell. Under the UD-NTC methodology the number of collision pairs per Delaunay cell per time step is given by:

$$N_{pairs} = \frac{1}{2V_{cell}} N_c \overline{N} F_N \tau \left\{\left(\sigma_T c_r\right)_{MAX}\right\} \tag{2.110}$$

where N_{pairs} is the number of pairs selected from the cell which will be considered for collision, V_{cell} is the volume of the Delaunay cell, N_c is the current number of computational particles in the Delaunay cell, $\overline{N}$ is the time-averaged number of computational particles in the Delaunay cell, τ is the time step (time elapsed per iteration), and F_N is the number of real particles represented by each simulation particle (particle weight).

Step 4. Generate N_{pairs} of collision partners from the cell's particle stack and evaluate the probability of collision between each pair. The pairs of particles that will total N_{pairs} are chosen at random from the list of particles in the cell. For each collision pair, the value of $\sigma_T c_r$ is determined using:

$$\sigma_T = \int_0^{4\pi} \sigma \, d\Omega = \pi \, d^2{}_{12} = \begin{cases} (\sigma_T)_{HS} = \pi \left(\dfrac{d_1 + d_2}{2} \right)^2 \\ (\sigma_T)_{VHS} = \dfrac{\pi}{4} \left[d_{1,ref} \left(v_{1,ref} / v_{rel} \right)^{\omega_1} + d_{2,ref} \left(v_{2,ref} / v_{rel} \right)^{\omega_2} \right] \end{cases} \quad (2.111)$$

$$c_r = || \, \boldsymbol{c}_1 - \boldsymbol{c}_2 \, || \quad (2.112)$$

If the value of $\sigma_T c_r$ for the pair is greater than $(\sigma_T c_r)_{MAX}$ for the current cell then the cell value of $(\sigma_T c_r)_{MAX}$ is updated. The probability of any chosen pair colliding is given as:

$$P_{coll} = \frac{\sigma_T c_r}{\{(\sigma_T c_r)_{MAX}\}} \quad (2.113)$$

Using P_{coll}, each collision pair is chosen to undergo a collision according to the acceptance-rejection algorithm, which is outlined in detail in Appendix A. If the pair is selected to undergo collision then the algorithm moves to the next step, otherwise, a new pair is chosen and the algorithm loops over Step 4 again.

Step 5. Calculate the post-collision velocities of the collision pair selected in Step 4. Calculation of the post-collision velocities is a multi-step process and differs slightly for each molecular model. However, an outline of the post-collision velocity calculation can be formed which applies to both models. The steps of the general algorithm of post-collision velocity calculation follow:

- Calculate the pre-collision relative velocity using:

$$\mathbf{c}_r = \mathbf{c}_1 - \mathbf{c}_2 \quad (2.114)$$

- Calculate the center of mass velocity by:

$$\mathbf{c}_{\mathbf{cm}} = \frac{m_1\mathbf{c}_1 + m_2\mathbf{c}_2}{m_1 + m_2} \tag{2.115}$$

- Generate the collision deflection angle (or scattering angle), χ.
- Generate the angle between the collision-plane and the absolute x-y plane, ε.
- Calculate the post-collision velocities in the localized coordinate system using:

$$\begin{aligned} c^*_{r,x'} &= c_r \cos(\chi) \\ c^*_{r,y'} &= c_r \sin(\chi)\cos(\varepsilon) \\ c^*_{r,z'} &= v_{rel} \sin(\chi)\sin(\varepsilon) \end{aligned} \tag{2.116}$$

- Then map the resulting post-collision velocities to global Cartesian coordinates using:

$$\begin{aligned} \mathbf{c}_1 &= \mathbf{c}_{\mathbf{cm}} + \frac{m_2}{m_1 + m_2}\mathbf{c}^*_r \\ \mathbf{c}_2 &= \mathbf{c}_{\mathbf{cm}} + \frac{m_1}{m_1 + m_2}\mathbf{c}^*_r \end{aligned} \tag{2.117}$$

2.7.3 Inelastic Collisions: Mathematical Model

In inelastic collisions internal energy can be transferred from one particle to another or from one mode to another. The exchange of rotational energy is captured in U-DSMC using the Larsen-Borgnakke model (Borgnakke and Larsen, 1975). The current implementation of rotational energy exchange is a reformulation of the previous implementation of Hammel (2002). As with elastic collisions, the condition of the algorithm inherited by the current investigator was well beyond minor debugging and as such the current implementation of rotational energy exchange is an algorithmic rewrite of the past formulation. However, the current implementation of rotational energy exchange is built upon the data structures of the previous implementation and follows the Larsen-Borgnakke method as specified in Hammel (2002).

Following the Larsen-Borgnakke methodology, as developed in Bird (1994), a fraction of the simulated collisions are treated as inelastic where the post collision internal and translational energies are set by sampling the Larsen-Borgnakke distribution. The fraction of inelastic collisions calculated is determined from tabulated internal energy relaxation rates which vary according to molecular composition. Within the methodology of Bird (1994), if a collision is chosen to be modeled as inelastic then the total energy of all available degrees of freedom for both molecules is reassigned between both the translational and internal modes by sampling from the equilibrium distributions of each mode with the appropriate total energy. The corresponding distribution function for application of the Larsen-Borgnakke method is

$$f\left(\frac{E_a}{E_a+E_b}\right)=f\left(\frac{E_b}{E_a+E_b}\right)=\frac{\Gamma(\Xi_a+\Xi_b)}{\Gamma(\Xi_a)\,\Gamma(\Xi_b)}\left(\frac{E_a}{E_a+E_b}\right)^{\Xi_a-1}\left(\frac{E_b}{E_a+E_b}\right)^{\Xi_b-1} \tag{2.118}$$

where Ξ_a denotes a group of modes with energy available for redistribution and Ξ_b denotes the remaining available modes and E_a is the portion of the total energy to be redistributed to group Ξ_a while E_b is the portion of the total energy to be reserved for redistributed to group Ξ_b. The application of this method is carried out by sampling the distribution above for each mode available at the total collision energy in a serial fashion and will be discuss in detail in the following section.

2.7.4 Inelastic Collisions: U-DSMC Implementation

Within the confines of a U-DSMC simulation the user has the option of specifying the modeling approach used to capture collision dynamics of the gas under study. The options currently available to a U-DSMC user are: fully elastic using the hard sphere model, inelastic using the hard sphere model, fully elastic using the variable hard sphere model or inelastic using

the variable hard sphere model. If either inelastic modeling technique is chosen then an internal energy exchange algorithm is activated within the elastic collision algorithm outline in the previous section. As noted earlier, in keeping with the Larsen-Borgnakke methodology only a fraction of the simulated collisions are treated as inelastic where the post collision internal and translational energies are modified. Therefore, the combined elastic-inelastic algorithm benefits from improved data structuring as well as decreased computation time.

The comprehensive algorithm used to model inelastic collisions is given below. Added emphasis is given to the portions that are inelastic specific. Further details on the elastic portions of the algorithm can be found in the previous section. The calculation of inelastic collisions in the current implementation of U-DSMC is governed by the following steps as applied to each Delaunay cell within the computational domain:

Step 1. Calculate cell values of N_c, $\left(\sigma_T c_r\right)_{MAX}$.

Step 2. Calculate the number of collision pairs for the cell following:

$$N_{pairs} = \frac{1}{2V_{cell}} N_c \overline{N} F_N \tau \left\{\left(\sigma_T c_r\right)_{MAX}\right\} \tag{2.119}$$

Step 3. Generate N_{pairs} of collision partners from the cell's particle stack and evaluate the probability of collision between each pair using:

$$P_{coll} = \frac{\sigma_T c_r}{\left\{\left(\sigma_T c_r\right)_{MAX}\right\}} \tag{2.120}$$

by applying the acceptance-rejection algorithm, which is outlined in detail in Appendix A.

Step 4. Determine if the first particle, q, of the accepted collision pair undergoes an inelastic collision. The probability of an inelastic collision is determined using the reciprocal of the particle's rotational relaxation number Λ_q:

$$P_{inel} = \frac{1}{\Lambda_q} \tag{2.121}$$

The acceptance-rejection algorithm is then applied to determine if the particle undergoes an inelastic collision. If the particle is chosen to undergo internal energy exchange, then several additional steps are required.

1. The sum of the average degrees of freedom Ξ is calculated following:

$$\Xi = \left(5/2 - \omega_{12}\right) + \zeta_{rot,1}/2 + \zeta_{rot,2}/2 \tag{2.122}$$

2. The total available energy for the Larsen-Borgnakke redistribution of internal energy is calculated:

$$E_c = E_{tr,12} + E_{rot,1} + E_{rot,2} \tag{2.123}$$

3. Each individual available internal mode is then serially selected to undergo energy exchange. At each internal mode selection, the mode under consideration Ξ_a is subtracted from the total modes leaving Ξ_b modes.
4. The ratio of the post-collision internal energy to the pre-collision available energy is calculated using:

$$\frac{E_a}{E_a + E_b} = 1 - R^{1/\Xi_b} \tag{2.124}$$

where R is a random fraction between 0 and 1, for a case of a single internal mode with two internal degrees of freedom or by sampling:

$$f\left(\frac{E_a}{E_a+E_b}\right)=\Xi_b\left(1-\frac{E_a}{E_a+E_b}\right)^{\Xi_b-1} \tag{2.125}$$

5. From the ratio generated in the previous step, the ratio of the probability of the generated value E_a to the maximum probability is evaluated following:

$$\frac{P_{E_a}}{P_{max}}=\left\{\frac{\Xi_a+\Xi_b-2}{\Xi_a-1}\left(\frac{E_a}{E_c}\right)\right\}^{\Xi_a-1}\left\{\frac{\Xi_a+\Xi_b-2}{\Xi_b-1}\left(\frac{E_a}{E_c}\right)\right\}^{\Xi_b-1} \tag{2.126}$$

6. The acceptance-rejection algorithm is then applied using the probability P_{E_a}/P_{max}. If the value of E_a is accepted then the selected energy is redistributed to the degrees of freedom of the current mode. If the value of E_a is rejected then the process returns to sub-step 4 and a new value of E_a is generated until an acceptable value is obtained. Upon redistribution of an acceptable E_a the available energy remaining for redistribution to the remaining modes is updated and the remaining modes are considered in turn.

Step 5. Following the redistribution of energy to the available internal modes the post-collision relative velocity is calculated using the redistributed translational energy E_{tr}, the pre-collision relative velocity $\boldsymbol{c}_r$ and the pre-collision relative speed c_r following:

$$\boldsymbol{c}_r^*=\frac{\boldsymbol{c}_r}{c_r}\left[2E_{tr}\left(\frac{m_1+m_2}{m_1m_2}\right)\right]^{1/2} \tag{2.127}$$

From which the post-collision molecular velocities can be determined from:

$$\begin{aligned}\mathbf{c_1}&=\mathbf{c_{cm}}+\frac{m_2}{m_1+m_2}\mathbf{c}_r^*\\ \mathbf{c_2}&=\mathbf{c_{cm}}+\frac{m_1}{m_1+m_2}\mathbf{c}_r^*\end{aligned} \tag{2.128}$$

2.8 Solid Surface Modeling

In a DSMC simulation the particles are free to interact with the solid surfaces that form the domain boundaries. Within the confines of most problems of engineering significance the gaseous flow interacts with solid surfaces composed of a variety of materials. Generally speaking, the gas typically has either a stagnation temperature or a static temperature that differs from the temperature of the solid surface. As a result the distribution function of the incident molecules will typically differ from that of the reflected molecules. Furthermore, the energy of a molecule relative to the surface will in general be different from the corresponding energy of the molecule after it has reflected from the surface, as such the gas-surface collisions are typically inelastic. The models for gas-surface interaction developed to date are primarily phenomenological and as such their applicability varies with the nature of the surface and the magnitude of the molecule's energy relative to the surface. The most widely used surface models are diffuse and specular reflection as well as generalizations derived from these models. Descriptions of these models as well as two variations of these models that have been implemented into U-DSMC in this work are given below.

2.8.1 Specular Reflection

Two models for the interaction of a stationary equilibrium gas with a solid surface were proposed by Maxwell (1879), the first of which was specular reflection. Specular reflection is a model for a perfectly elastic collision between a gas molecule and a solid surface. Under the specular reflection model, an impinging molecule's surface-normal velocity component is

reversed during the collision while the surface-tangential velocity components remain unchanged:

$$\begin{aligned} \mathbf{c}_{\perp}{}^{*} &= -\mathbf{c}_{\perp} \\ \mathbf{c}_{\parallel}{}^{*} &= \mathbf{c}_{\parallel} \end{aligned} \tag{2.129}$$

As a consequence the angle between the molecule's incident velocity and the surface, θ_i, is equal to the angle between the surface and the molecule's reflected velocity, θ_r, this is shown schematically in Figure 24.

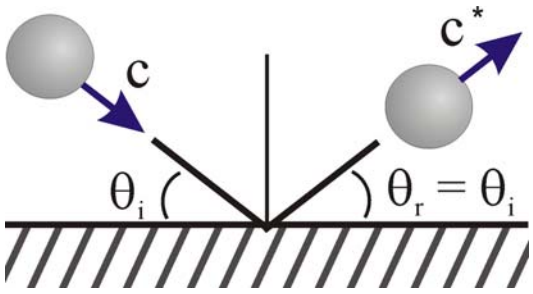


Figure 24. Illustration of specular reflection for a single particle.

Due to the nature of the specular reflection model, a specularly reflecting surface is functionally identical to a plane of symmetry, as such it is utilized to model a symmetry plane as needed throughout this work. From a bulk gas standpoint, a stream of particles that is incident to a specular surface will reflect as a coherent stream with the surface-normal component of the stream's directionality reversed. This effect is illustrated in Figure 25 below. The original implementation of the specular model in U-DSMC was carried out in Hammel (2002) and remains largely unchanged in the current implementation.

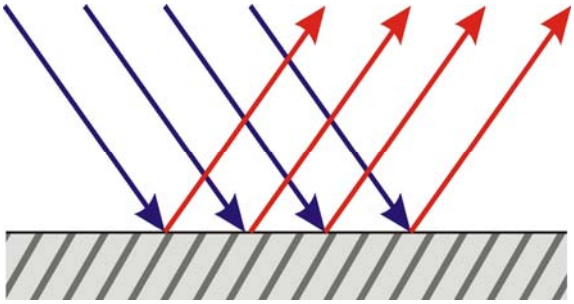


Figure 25. Illustration of the bulk effect of specular reflection for a stream of particles.

2.8.2 Diffuse Reflection

The second model proposed by Maxwell (1879) is that of diffuse reflection. In diffuse reflection a gas-surface interaction is modeled as an absorption-reemission process where the reemitted state of the particle is determined from sampling an equilibrium distribution with a temperature corresponding to that of the solid surface. From a bulk gas standpoint, a stream of particles that is incident to a diffuse surface will leave the surface in such a manner as to be equivalent to having the stream pass through the surface while molecules flux across the surface from a distribution equal to a stationary gas with a temperature equal to the wall temperature. The bulk effect of diffuse reflection is illustrated in Figure 26 below. Further details regarding implementation of a diffuse model are given in the following section. The initial implementation of the diffuse model in U-DSMC was carried out in Hammel (2002) and has undergone minor algorithmic correction in the current work.

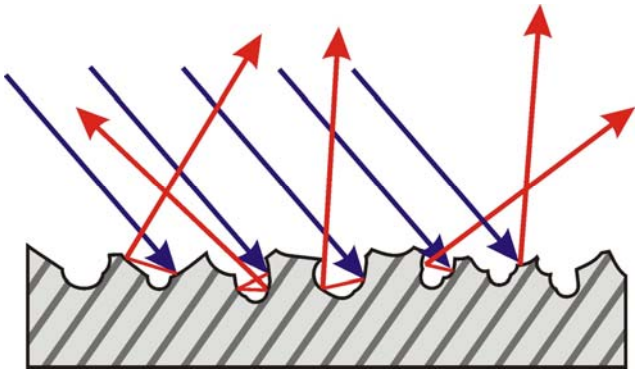


Figure 26. Illustration of the bulk effect of diffuse reflection for a stream of particles.

2.8.3 Non-diffuse Reflection

In many applications of engineering significance the gaseous interaction with the solid surface does not fit either the purely specular or fully diffuse models. Therefore, an extension model has been developed in the current work which combines the specular and diffuse models to form a model that achieves partial accommodation of energy and momentum. This model is typically referred to as the non-diffuse model (Bird, 1994). Within the non-diffuse model the user has control over the percentage of gas-surface interactions that are modeled as fully diffuse. The control parameter is the diffuse fraction ε, which determines the percentage of interactions that are treated as undergoing fully diffuse reflection. It should be noted that although the non-diffuse model does extend the phenomenological surface handling capabilities of U-DSMC the model is simplistic in nature and should be used to enhance surface approximations for problems of engineering significance only. The non-diffuse model lacks detailed capturing of the underlying potential interactions required to accurately scrutinize the meticulous balance which occurs during real gas-surface interactions. However, the non-diffuse model is a valid extension

of the phenomenological treatment of gas-surface interactions in U-DSMC and as such the current implementation will be outline below.

The process of modeling gas surface interactions in U-DSMC is composed of several steps. The first step is to determine if the particle's motion is such that a solid surface interaction occurs. Treatment of this consideration is handled in the particle tracing algorithm which was presented in Section 2.6. If the particle does reflect off a solid surface then the face of reflection is used to determine what type of reflection model is applied. Tabulation of the surface model applied at each face is handled using a surface label referred to as a face attribute. During the grid generation process the user specifies the face attribute to be applied to each flow boundary. This information is then linked to the boundary conditions applied. From this data the surface model along with its supporting parameters are extracted during gas-surface interactions.

Once the surface model for the current face-particle pair has been determined the interaction is mapped to face-fitted coordinates. Using the face normal $\boldsymbol{n}$ and the unit vector $\boldsymbol{a}_1$ of the face edge $\boldsymbol{r}_{12}$ a localized coordinate system can be generated as shown in Figure 27.

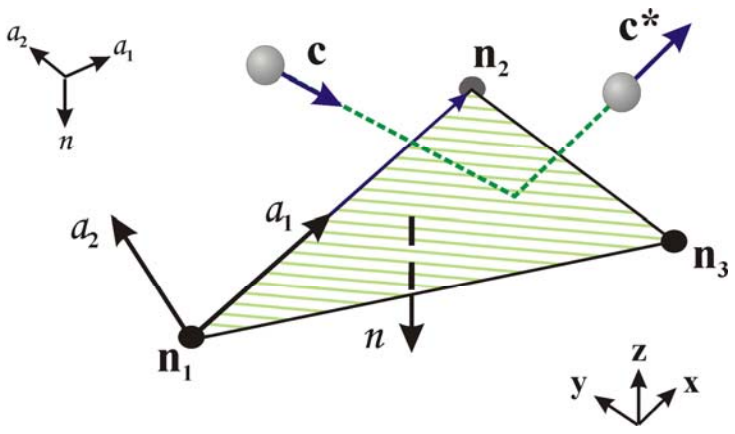


Figure 27. Face-fitted localized coordinate system used for surface modeling in U-DSMC.

Following the non-diffuse implementation, the next step of the reflection model is to determine if the particle will under go diffuse reflection or specular reflection. To determine the model used a random fraction R is generated and a selection algorithm is applied. If $R > \varepsilon$ then the interaction is modeled as specular. Following the specular reflection model the particle's velocity components are updated following:

$$\boldsymbol{c}_{||}^{*} = \boldsymbol{c}_{||} \tag{2.130}$$

$$\boldsymbol{c}_{\perp}^{*} = \boldsymbol{c}_{\perp} \tag{2.131}$$

If $R \leq \varepsilon$ then the interaction is modeled as fully diffuse. Following the diffuse reflection model the particle's velocity components are generated from sampling from the equilibrium distribution function corresponding to the wall temperature. The algorithm used for generating the local face-fitted velocity components is outlined below.

Step 1. Using the wall temperature T_w and the mass m for the current particle, the inverse of the most probable thermal speed β is calculated following:

$$\beta = \sqrt{\frac{m}{2k_B T_w}} \tag{2.132}$$

Step 2. The velocity components perpendicular to the face normal are generated by sampling the product of two non-drifting Maxwellian distributions yielding a distribution of the form:

$$f(\boldsymbol{c}^{*}_{a_1}) f(\boldsymbol{c}^{*}_{a_2}) = \left(\frac{\beta^2}{\pi}\right) exp\left\{-\beta^2 \left(\boldsymbol{c}^{*\,2}_{a_1} + \boldsymbol{c}^{*\,2}_{a_2}\right)\right\} \tag{2.133}$$

which can be sampled using the methods given in Appendix A. In order to generate $\boldsymbol{c}_{a_1}$ and $\boldsymbol{c}_{a_2}$ the following relations can be used:

$$\boldsymbol{c}^*_{a_1} = r\cos\theta \tag{2.134}$$

$$\boldsymbol{c}^*_{a_2} = r\sin\theta \tag{2.135}$$

For which values of r and θ can be generated utilizing separate calls to a random number generator, used to obtain a random fraction R:

$$\theta = 2\pi R \tag{2.136}$$

$$r = \frac{\sqrt{-ln(R)}}{\beta} \tag{2.137}$$

yielding:

$$\boldsymbol{c}^*_{a_1} = \frac{\sqrt{-ln(R_1)}}{\beta}cos(2\pi R_2) \tag{2.138}$$

$$\boldsymbol{c}^*_{a_2} = \frac{\sqrt{-ln(R_1)}}{\beta}sin(2\pi R_2) \tag{2.139}$$

Step 3. The velocity component parallel to the face normal is generated by sampling a biased non-drifting Maxwellian distribution of the form:

$$f(\boldsymbol{c}^*_n) = \left(\frac{\beta}{\pi^{1/2}}\right)exp\left\{-\beta^2\left(\boldsymbol{c}^{*\,2}_n\right)\right\} \tag{2.140}$$

utilizing the selection-rejection method outlined in Appendix A. Once a suitable face-normal component has been generated using the acceptance-rejection method the three components of the particle's new velocity are mapped back into Cartesian coordinates and the cycle is continued.

2.8.4 Moving Walls

In many problems of engineering significance the solid surfaces are moving in a manner that cannot be mapped through the transference to a fluid-moving reference frame. In order to allow for modeling cases where the wall motion must be considered directly, such as Couette flow, a model for capturing scalar Cartesian wall motion has been added to U-DSMC with in this work. The current implementation of wall motion is interfaced within the structure of the non-diffuse model. The effect of the motion of the solid surface during gas-surface interactions is handled by means of a local reference frame for the surface face being impinged. Within the local reference frame the surface interaction is modeled using the non-diffuse reflection algorithm outlined above. Upon completion of the stationary gas-surface interaction, with respect to the local reference frame, the reference frame's motion is added to the particle's resultant post-interaction velocities following:

$$\left(\boldsymbol{c}^{*}\right)_{abs} = \left(\boldsymbol{c}^{*}\right)_{loc} + \boldsymbol{c}_{w} \tag{2.141}$$

2.9 Sampling Macroscopic Parameters

In U-DSMC simulations, macroscopic flow properties such as density, pressure, temperature and velocity must be sampled. Within the current U-DSMC implementation, instantaneous cell based averages are calculated according to the following definitions (Bird, 1994):

$$\rho = \sum_{p=1}^{s}\left(m_p n_p\right) = n\bar{m} \tag{2.142}$$

$$p = -\sum_{p=1}^{s}\frac{1}{3}m_p n_p \overline{c'_p{}^2} = \frac{1}{3}n\overline{mc'^2} \tag{2.143}$$

$$\frac{3}{2}k_B T_{tr} = \frac{1}{2}\sum_{p=1}^{s}\left\{\left(n_p / n\right) m_p \overline{c'^2_p}\right\} = \frac{1}{2}\overline{mc'^2} \tag{2.144}$$

$$\frac{3}{2}k_B T_{tr,p} = \frac{1}{2}m_p \overline{c'^2_p} \tag{2.145}$$

$$\mathbf{c}_o = \frac{1}{\rho}\sum_{p=1}^{s}\left(m_p n_p \overline{\mathbf{c}}_p\right) \tag{2.146}$$

Nodal volume-weighted instantaneous averages are generated from the Delaunay cell based values following:

$$\Upsilon_{node} = \frac{\sum_{i=1}^{cells} \Upsilon_i V_i}{\sum_{i=1}^{cells} V_i} \tag{2.147}$$

where the summation occurs over all cells to which the node is attached, Υ is the property of interest, and V is the cell volume. A schematic illustrating the Delaunay structure and nodal volume-weighted averaging is shown in Figure 28.

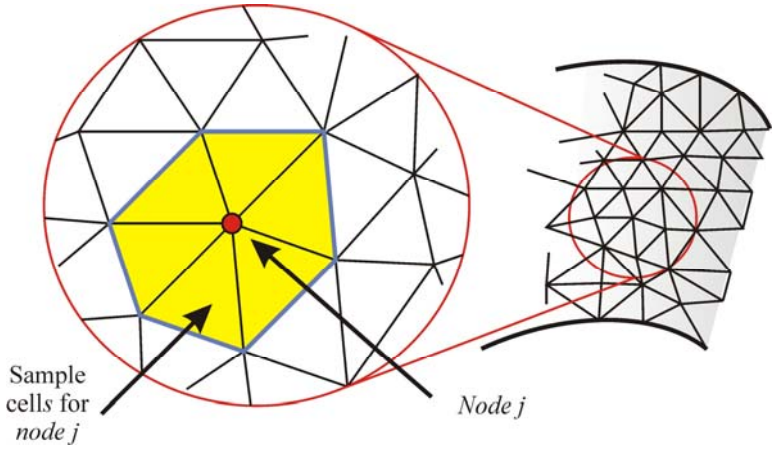


Figure 28. Illustration of the Delaunay structure used in nodal volume-weighted averaging.

The advantages of nodal averaging are a smoother parameter field, decreased statistical fluctuations and simplified data output formatting. The disadvantages of nodal averaging are the increased computations and the decrease in the minimum cell length requirement needed for flow gradient resolution. The benefits have been found to outweigh the detriments. The underlying implementation of macroscopic sampling in U-DSMC was developed in Hammel (2002). Addition of scalar pressure calculations as well as output file labeling required for ensemble averaging has been added to the base implementation within the current effort.

2.10 Flux Capturing Through Interior or Free Boundary Surfaces

When modeling gaseous flows it is often advantageous to capture the number flux, mass flux or distribution functions at arbitrary points within the flow domain. Flux capturing within the confines of unstructured grids has been developed and implemented into the U-DSMC code within the course of the current effort. The implementation of flux capturing is built upon the specification of a flux surface during the grid generation process. Within the confines of the modified grid generation techniques the user may place a sampling surface in any location within the flow field with arbitrary size, shape or orientation. The flux capturing algorithm requires only that the surface of interest be specified, through a face attribute label, as either a free boundary or an internal face. The flux capturing algorithm allows the user to obtain data outputs for any combination of number flux, mass flux or distribution function data for each species in the flow field. A general schematic of the geometry utilized in flux capturing is shown in Figure 29.

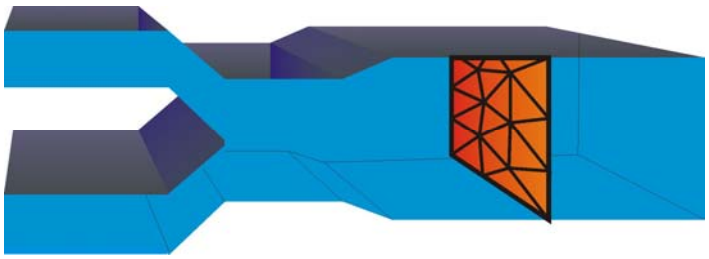


Figure 29. General schematic of a flux capturing surface imbedded in a flow domain.

The species specific number flux Γ_{N_q} through a specified surface is calculated by tabulating the number of molecules of species q to pass through the specified surface over an interval Δt through the total area of the specified surface following:

$$\Gamma_{N_q} = \frac{\sum_{i=1}^{\eta} F_N}{\Delta t \sum_{j=1}^{faces} A_{s,j}} \tag{2.148}$$

In order to keep the implementation geometrically flexible the area of the specified surface is calculated from a summation over all faces, with surface area $A_{s,j}$, that feature the specified face attribute.

The species specific mass flux Γ_{m_q} through a specified surface is calculated by tabulating the mass of species q molecules which pass through the specified surface over an interval Δt through the total area of the specified surface. The mass flux calculation takes the form:

$$\Gamma_{m_q} = m_q \Gamma_{N_q} = \frac{\sum_{i=1}^{\eta_q} m_q F_N}{\Delta t \sum_{j=1}^{faces} A_{s,j}} \tag{2.149}$$

The velocity distribution function of the particles of species q that pass through the specified surface can be formed from a tabulation of the three Cartesian velocity components of each particle of species q that transverses the specified surface. Each component of velocity is tabulated for each species following:

$$V_x^q(\eta(i),\Delta t) = u \tag{2.150}$$

$$V_y^q(\eta(i),\Delta t) = v \tag{2.151}$$

$$V_z^q(\eta(i),\Delta t) = w \tag{2.152}$$

where $\eta(i)$ is the data location of the current particle sample and Δt is the sampling duration over which the velocity components are tabulated. After the duration Δt has passed the sampled velocity components are written to an output file for post-processing into the respective velocity distribution functions. The duration Δt is a user specified value. Tabulated distribution function data is cleared from memory after each Δt duration and a new data set is collected.

2.11 Surface Transport Properties

In many problems of engineering significance the resulting forces and heat transfer of impinging particles on a surface must be quantified. As such, calculations of the pressure, shear and heat flux at the solid surface boundaries have been added to U-DSMC

The pressure and shear stress on the solid surface is determined from the momentum exchange of the impinging particles following:

$$p_{surf} = \frac{\sum_j m_j F_N \left[\left(\boldsymbol{c}_\perp^* \right)_j - \left(\boldsymbol{c}_\perp \right)_j \right]}{A_s \Delta t} \tag{2.153}$$

$$\tau_{surf} = \frac{\sum_j m_j F_N \left[\left(\boldsymbol{c}_{||}^* \right)_j - \left(\boldsymbol{c}_{||} \right)_j \right]}{A_s \Delta t} \tag{2.154}$$

where m_j is the mass of an impinging particle, F_N is the particle weight of the particle species, A_s is the area of the surface element and Δt is the duration of impingement sampling. The subscripts on the initial and final particle velocities signify directionalities normal, $(\perp)$, and tangential, $(||)$, to the surface element plane. The relevant geometry is shown in Figure 30.

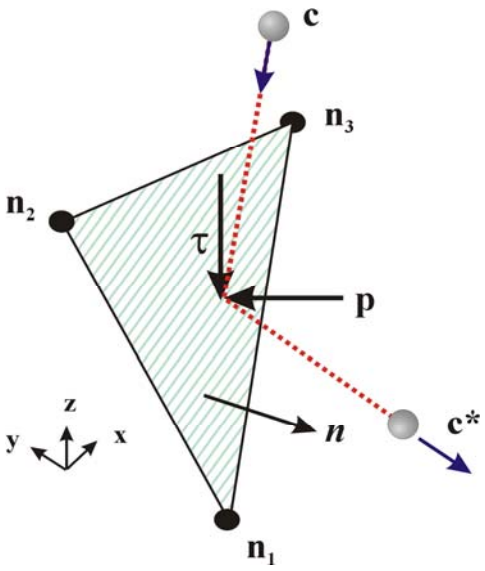


Figure 30. Illustration of the parameters used to calculate the pressure and shear stress on a solid surface face element.

In an unstructured computation, care must be taken in determining the normal and tangential components of the velocity vectors. The directionality of the surface normal may be constructed from the normalized cross product of two surface element edges. Once the surface

normal vector has been established the normal and tangential components of the particles initial and final velocity may be determined from simple vector relations. The heat transfer to the surface is comprised of the effects of the translational energy and internal energy exchange between the surface element and impinging particles. In a general form, the heat flux may be written as:

$$q_{surf} = \frac{\sum_j F_N \left[\left(E^* \right)_j - \left(E \right)_j \right]}{A_s \Delta t} \tag{2.155}$$

From this relation the heat flux to a surface element can be calculated for monatomic and polyatomic molecules.

From the values of face shear, pressure and heat flux local and global surface coefficients may be calculated. Calculations of both the local and global coefficients have been fully implemented in this work utilizing the basis data structure and storage arrays from the work of Hammel (2002). From the face-specific data for pressure p_j, in combination with the free stream data for pressure p_∞, density ρ_∞ and drift velocity U_∞ the pressure coefficient for the current face j can be calculated as (White, 1999):

$$\left(C_p \right)_j = \frac{p_j - p_\infty}{\frac{1}{2} \rho_\infty U_\infty^2} \tag{2.156}$$

Similarly, the local skin friction coefficient can be determined from:

$$\left(C_f \right)_j = \frac{\tau_j}{\frac{1}{2} \rho_\infty U_\infty^2} \tag{2.157}$$

Likewise the local heat transfer coefficient may be evaluated using:

$$\left(C_h\right)_j = \frac{q_j}{\frac{1}{2}\rho_\infty U_\infty^3} \tag{2.158}$$

From the local values of p_j, τ_j and q_j global values can be obtained for p_{tot}, τ_{tot} and q_{tot} following:

$$p_{tot} = \frac{\sum_{j=1}^{faces} p_j A_{s,j}}{\sum_{j=1}^{faces} A_{s,j}} \tag{2.159}$$

$$\tau_{tot} = \frac{\sum_{j=1}^{faces} \tau_j A_{s,j}}{\sum_{j=1}^{faces} A_{s,j}} \tag{2.160}$$

$$q_{tot} = \frac{\sum_{j=1}^{faces} q_j A_{s,j}}{\sum_{j=1}^{faces} A_{s,j}} \tag{2.161}$$

It should be noted that due to the directionality involved in the shear calculations the vast majority of surface geometries of engineering importance possess surfaces with multidirectional components of shear which can negate one another. To aid in the calculation of directional forces that are typically desired in engineering applications, the local and total directional forces are also calculated in the current U-DSMC implementation using both local coordinates and the more general Cartesian force vectors as well. The calculation of surface forces follows directly from first principles and can be written in vector form as:

$$\boldsymbol{F}_j = \frac{\partial \boldsymbol{M}_j}{\partial t} \frac{\sum_{p=1}^{collisions} m_p \left(\boldsymbol{c}_p - \boldsymbol{c}_p^*\right)}{\Delta t} \tag{2.162}$$

with,

$$\boldsymbol{F}_{tot} = \frac{\partial \boldsymbol{M}_{tot}}{\partial t} = \sum_{j=1}^{faces} \boldsymbol{F}_j = \frac{\sum_{j=1}^{faces} \sum_{p=1}^{collisions} m_p \left(\boldsymbol{c}_p - \boldsymbol{c}_p^* \right)}{\Delta t} \tag{2.163}$$

From the total values of $\boldsymbol{F}_{tot}$, p_{tot}, τ_{tot} and q_{tot} the global or total values of $\left(C_p\right)_{tot}$, $\left(C_f\right)_{tot}$, and $\left(C_h\right)_{tot}$ as well as the coefficient of drag $\left(C_D\right)_{tot}$ and the coefficient of lift $\left(C_L\right)_{tot}$ can be determined following:

$$\left(C_p\right)_{tot} = \frac{p_{tot} - p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{2.164}$$

$$\left(C_\tau\right)_{tot} = \frac{\tau_{tot}}{\frac{1}{2}\rho_\infty U_\infty^2} \tag{2.165}$$

$$\left(C_q\right)_{tot} = \frac{q_{tot}}{\frac{1}{2}\rho_\infty U_\infty^3} \tag{2.166}$$

$$\left(C_D\right)_{tot} = \frac{\left(\boldsymbol{F}_{tot}\right)_{Drag}}{\frac{1}{2}\rho_\infty U_\infty^2 A_{char}} \tag{2.167}$$

$$\left(C_L\right)_{tot} = \frac{\left(\boldsymbol{F}_{tot}\right)_{Lift}}{\frac{1}{2}\rho_\infty U_\infty^2 A_{char}} \tag{2.168}$$

where A_{char} is the characteristic area of the body. By means of the coupling the geometrically general implementation of the calculation of gas-solid surface force and energy exchange given above and the geometric flexibility and ease of use of the current U-DSMC grid generation methods significant advances in rarefied flow modeling over solid bodies has been achieved in this work. One example of the application of U-DSMC to characterize flow effects over a blunt body in a rarefied flow can be seen in Marchetti (2006).

3. VALIDATION AND ORDER-OF-ERROR APPROXIMATION

The validation and verification of the U-DSMC implementation is achieved through the modeling of a number of test cases with comparisons to theoretical formulations and experimental data. Each case presented in this chapter verifies and validates a specific subset of U-DSMC modeling capabilities. Additionally, a basis case has been applied in Section 3.2 in order to carry out an approximation of the order of error of the current U-DSMC implementation with respect to Delaunay cell size, time step, and the number of simulation particles in each Delaunay cell.

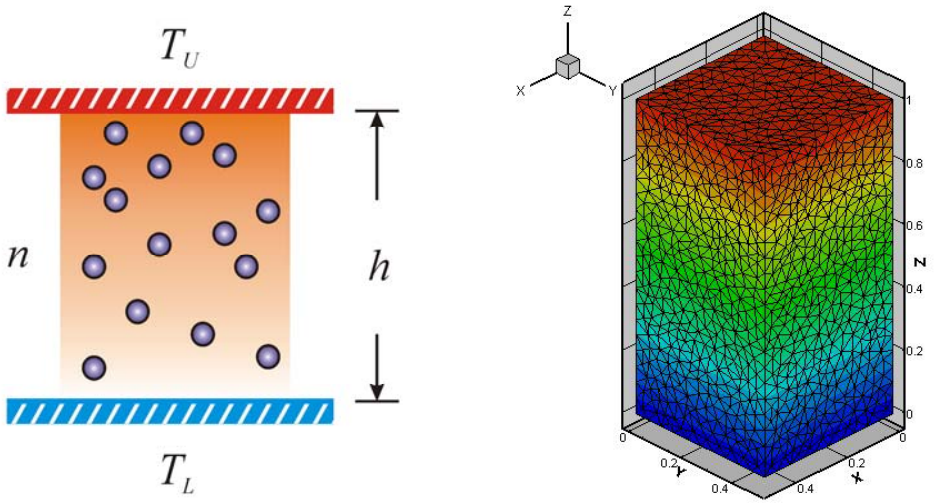
3.1 Transitional Heat Transfer Between Parallel Plates

The first test case involves one-dimensional heat transfer between two stationary infinite plane parallel plates. A range of Knudsen numbers is simulated and the heat transfer predicted by U-DSMC for each is compared to theoretical formulations. For large Knudsen numbers the heat transfer is carried out primarily through molecular transport across the plate separation. However, for small Knudsen numbers, in the slip to continuum range, the primary means of heat transfer is through energy exchange from molecular collisions. Therefore, this test case verifies the elastic portion of the VHS collision algorithm as well as the diffuse and specular boundary conditions, particle motion, and heat transfer sampling.

3.1.1 Geometry and Boundary Conditions

The schematic shown in Figure 31 (a) illustrates the test case geometry. The stationary parallel plates are separated by a distance h. The upper plate is assumed to reflect molecules diffusely with complete thermal accommodation to a temperature T_U while the lower plate is

also assumed to diffusely reflect molecules but to a temperature T_L. A gas composed of Maxwell molecules resides between the plates and has a overall number density of n. An example of the grid structure used in these investigations is shown Figure 31 (b).



(a) **(b)**

Figure 31. Schematic of parallel plate test scenario (a) and an example grid used in simulations (b).

3.1.2 Results

Three theoretical formulations from Bird (1994) are used to validate the U-DSMC code. The first formulation corresponds to the free-molecule limit where $h \ll \lambda$. The net upward heat flux for the free-molecule limit is given as:

$$q_{fm} = -2^{3/2} p \left(\frac{k_B}{\pi m} \right)^{1/2} \left(T_U^{1/2} - T_L^{1/2} \right) \tag{3.1}$$

where $p = n k_B T_U^{1/2} T_L^{1/2}$, m is the mass of the gas particles, and k_B is the Boltzmann constant.

The second formulation is that of continuum heat transfer between parallel plates, and is given by:

$$q_c = -\frac{C}{2h}\left(T_U^{\,2} - T_L^{\,2}\right) \tag{3.2}$$

with,

$$C = \frac{12 k_B^{\,3/2}}{8\left(\pi m T_{ref}\right)^{1/2} d_{ref}^{\,2}} \tag{3.3}$$

where T_{ref} is the reference temperature and d_{ref} is the reference diameter for the VHS model.

The third formulation corresponds to the transitional regime and is obtained by means of the four-moment method:

$$q_{4m} = \frac{\left\{\left(BT_U + AT_L + h/C\right)^2 - \left(B^2 - A^2\right)\left(T_U^{\,2} - T_L^{\,2}\right)\right\}^{1/2} - \left(BT_U + AT_L + h/C\right)}{B^2 - A^2} \tag{3.4}$$

where,

$$A = \frac{1}{2p}\left(\frac{\pi m T_U}{2k}\right)^{1/2} \tag{3.5}$$

and,

$$B = \frac{1}{2p}\left(\frac{\pi m T_L}{2k}\right)^{1/2} \tag{3.6}$$

The U-DSMC code was applied to parallel plate heat transfer with $T_U = 1000\,\mathrm{K}$, $T_L = 250\ \mathrm{K}$, and $h = 1\,\mathrm{m}$. The gas molecule properties are based on that of argon with $m = 66.3\times10^{-27}\ \mathrm{kg}$, $d_{ref} = 4.17\times10^{-10}\,\mathrm{m}$, and $T_{ref} = 273\ \mathrm{K}$. A range of Knudsen numbers was investigated by varying the number density between the plates. The results are shown in Figure 32.

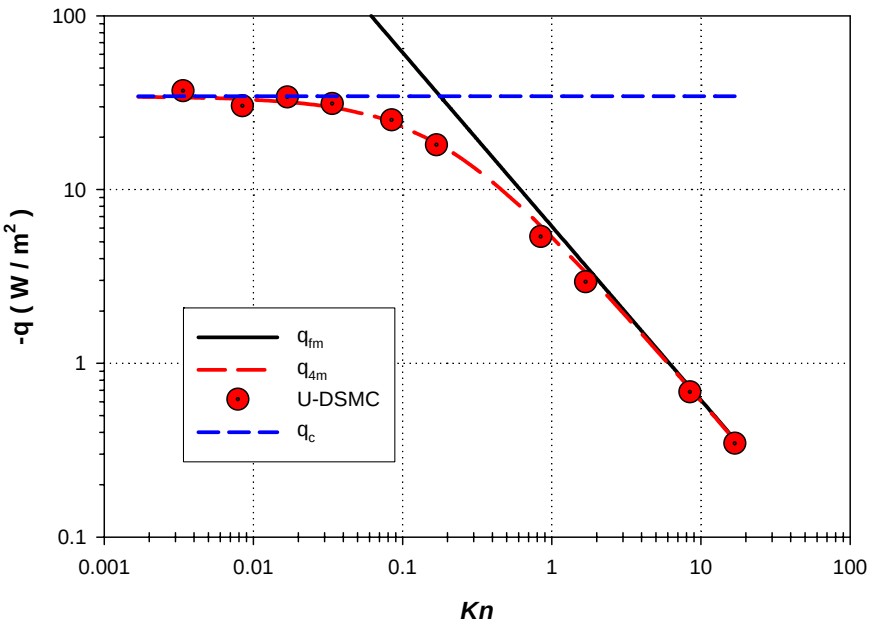


Figure 32. Heat transfer between parallel plates in transitional regime.

3.1.3 Conclusions

As shown in Figure 32, excellent agreement between the U-DSMC results and the four-moment solution occurs over the range of Knudsen numbers investigated. This agreement lends confidence to current implementation of particle motion, collisions, wall-interaction, and heat transfer calculations. Due to the wide range of Knudsen numbers modeled, the code was required to capture a variety of energy transport processes. As previously stated, in the free-molecule range the heat is transferred from the upper plate to the lower plate by individual molecular crossings while in the continuum regime the heat is transferred by means of collision-driven exchange. In the transitional regime a combination of heat transport phenomena occur. Therefore the agreement shown in Figure 32 illustrates the proper implementation of the underlying processes in the collision algorithm of U-DSMC. It should be noted that this test

problem exhibits a high degree of sensitivity to cell spacing and thus is an excellent test scenario for understanding the relationship between grid spacing and transport phenomena. Furthermore, the simplicity of this test case lends itself well to an order of error approximation and has thus been used as such in Section 3.2.

3.2 Order of Error Approximation

In order to determine the influence of the cell size, time step and particles per cell on U-DSMC results an order of error approximation has been carried out. The case of heat transfer between parallel plates is an excellent test case for approximating the order of error associated with each key parameter. For each parameter investigated, a base simulation was performed such that the error between the U-DSMC predictions and the 4-moment solution was negligible (less than 0.5 %). From this base parameter set, manipulations of the investigated parameter were made whilst the remaining parameters were fixed at their respective optimal values. The test case used for all investigations below corresponded to near continuum parallel plate heat transfer with $h = 1\,\mathrm{m}$, $Kn = 0.01$, and a wall temperature ratio of 4 with $T_U = 1000\,\mathrm{K}$.

3.2.1 Effect of Delaunay Cell Size

The first parameter investigated is the Delaunay cell spacing. The maximum cell spacing criteria for DSMC is commonly taken as

$$\frac{\lambda}{4} < l_c < \frac{\lambda}{3} \tag{3.7}$$

where λ is the local mean free path and l_c is the collision cell edge length. However, the effect of the cell spacing on the error of the simulation results is worth investigating for U-DSMC, due

to its implementation on unstructured grids. Therefore the error associated with the Delaunay cell spacing and U-DSMC results has been quantified.

The percent error between the U-DSMC predictions for the heat flux and those of the four-moment method are plotted against the normalized cell spacing in Figure 33. The error is calculated as the root-mean-square of the time-averaged U-DSMC results, denoted by $\langle\ \rangle$, and the theoretical value following:

$$error = \sqrt{\left(\left\langle q_{U-DSMC} \right\rangle - q_{4M}\right)^2} \tag{3.8}$$

$$\%Error = \frac{error}{q_{4M}} \tag{3.9}$$

The normalized cell spacing is simply the ratio of the cell spacing to the mean free path. It should be noted that in the work of Spirkin (2006) the standard deviation of cell edge length in grids generated using U-GridGen from the specified value was found to be less than 10% for uniform grids. Therefore a small uncertainty exists for the normalized cell spacing. In Figure 33 the percent error is plotted against the nominal normalized cell spacing. From the linearity of the data fit shown in Figure 33, it is clear that the error of the U-DSMC simulations for this case is first order with respect to the cell spacing.

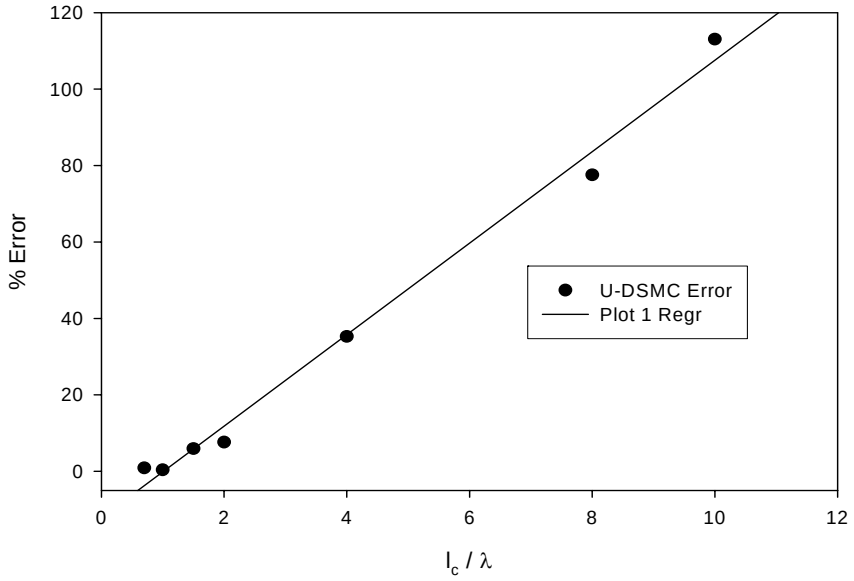


Figure 33. Percent error in U-DSMC results as a function of cell spacing.

3.2.2 Effect of Time Step

The second parameter under investigation is the time step. The commonly applied criterion for choosing the appropriate time step for DSMC is (Bird,1994):

$$\tau < \tau_c = \frac{1}{\nu} = \frac{1}{\sum_{p=1}^{s}\left(\frac{n_p}{n}\sum_{q=1}^{s}\left(n_p \overline{\sigma_{T,pq} c_{r,pg}}\right)\right)} \tag{3.10}$$

where τ_c is the mean collision time, ν is the mean collision rate, and τ is the simulation time step. However, the order of error associated with the time step must be ascertained for the current U-DSMC implementation.

The resulting error percentage between the U-DSMC predictions and those of the four-moment method are plotted against the normalized time step in Figure 34. Once again the error is calculated as the root-mean-square while the normalized time step is the ratio of the time step

to the mean collision time. From the linearity of the data fit in Figure 34 it is clear that the error of the U-DSMC simulations for this case is first order with respect to the time step.

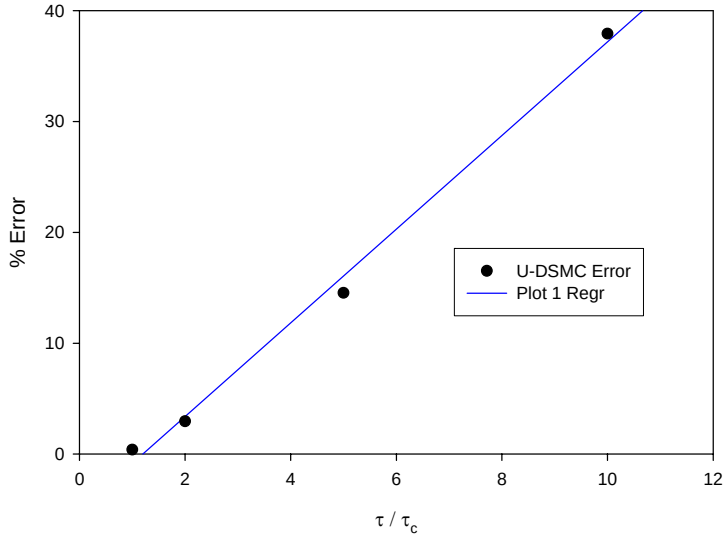


Figure 34. Error in U-DSMC results as a function of time step.

3.2.3 Effect of Simulation Particles in each Delaunay Cell

The final parameter under investigation is the number of simulation particles in each cell. A minimum number of twenty particles per collision cell is typically taken as satisfactory (Bird, 1994) while a minimum of 100 particles per sample cell is associated with negligible statistical fluctuations in macroscopic flow parameters. To clarify the effect of the number of particles in each Delaunay cell, used as the foundation of the current U-DSMC implementation, the order of error associated with the number of particles in each cell has been studied.

The resulting error between the U-DSMC predictions and those of the four-moment method are plotted against the normalized number of particles per cell in Figure 35. The error is calculated as the root-mean-square between the U-DSMC predictions and the theoretical value while the normalized number of particles per cell is the ratio of the number of particles per cell to

the commonly accepted value of 20 particles per cell. From the functionality of the data fit in Figure 35 it is shown that the error of the U-DSMC simulations for this case is of the order $1/N_{sim}$ with respect to the number of particles per cell.

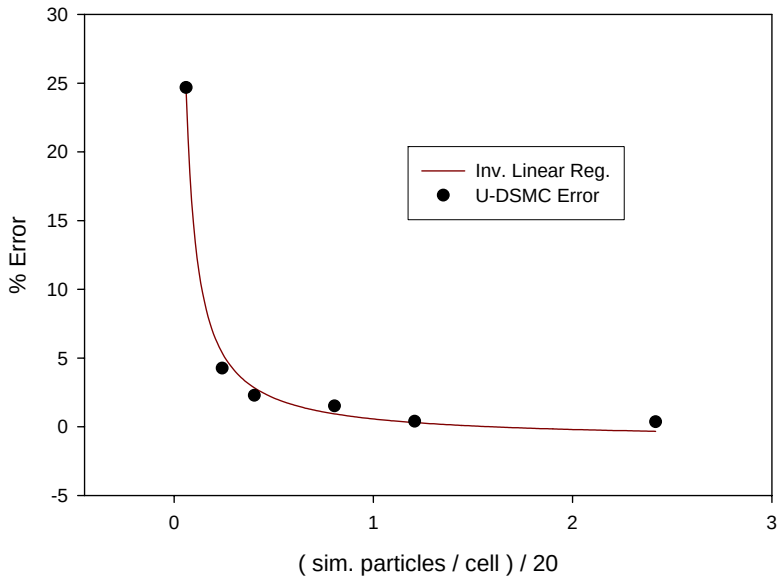


Figure 35. Error as a function of particles per cell for the current U-DSMC implementation.

3.2.4 Conclusions

From the investigations above the order of error associated with the cell edge length, time step and particles per cell have been determined. For the case of near-continuum heat transfer between parallel plates the current U-DSMC implementation exhibits first order error in both space and time as seen from the studies of cell spacing and time step size. However, the near-continuum parallel plate test case revealed an inverse first order error associated with the number of simulation particles in each Delaunay cell.

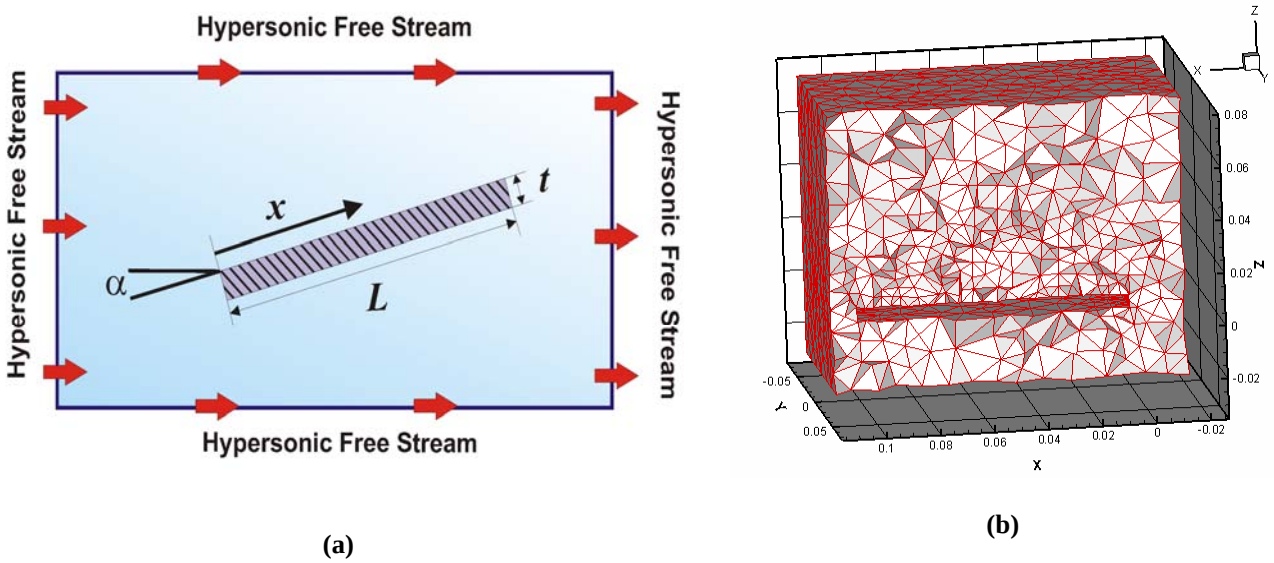
3.3 Hypersonic Flow Over A Flat Plate

The second test case validates the ability of the current implementation of U-DSMC to model external flows of real gases by means of comparison with experimental data for hypersonic flow over a flat plate. This case also provides an opportunity to test the effectiveness of the non-diffuse surface model. The test case is taken from Allegre et al (1993) and features a blunt flat plate immersed in a free-stream flow of nitrogen at two angles of attack. As such, this case requires that collisions are modeled as inelastic and that the rotational modes of the nitrogen molecules be properly modeled within U-DSMC. Furthermore, comparison with experimental data offers the opportunity to validate hypersonic free stream boundaries, local pressure and heat flux capturing.

3.3.1 Geometry and Boundary Conditions

The geometry and boundary conditions are shown in Figure 36. As seen in Figure 36 (a), the plate is oriented parallel to the flow direction when $\alpha = 0$ degrees and the flow is directed toward the top of the plate when $\alpha = 10$ degrees. Within the experiment of Allegre et al (1993) measurements of heat transfer and pressure were made at various locations along the center line of the upper surface of the plate. The length and width of the plate are both 0.1 m while the plate thickness is 0.005 m. The wall temperature was fixed at 290 K throughout the experiment and thus is fixed at this value throughout the U-DSMC simulations. For the purpose of validation, U-DSMC simulations have been carried out using the experimental flow conditions and the resulting heat transfer and pressure distributions are compared to experimental data. Hypersonic free stream boundaries are applied along the computational domain boundaries. The plate itself

is modeled as both fully diffuse and non-diffuse with $\varepsilon = 0.8$. An example grid utilized in this investigation is shown is shown in Figure 36 (b).



(a) **(b)**

Figure 36. Geometry and boundary conditions for flat plate test case (a) and an example grid from the flat plate simulations (b).

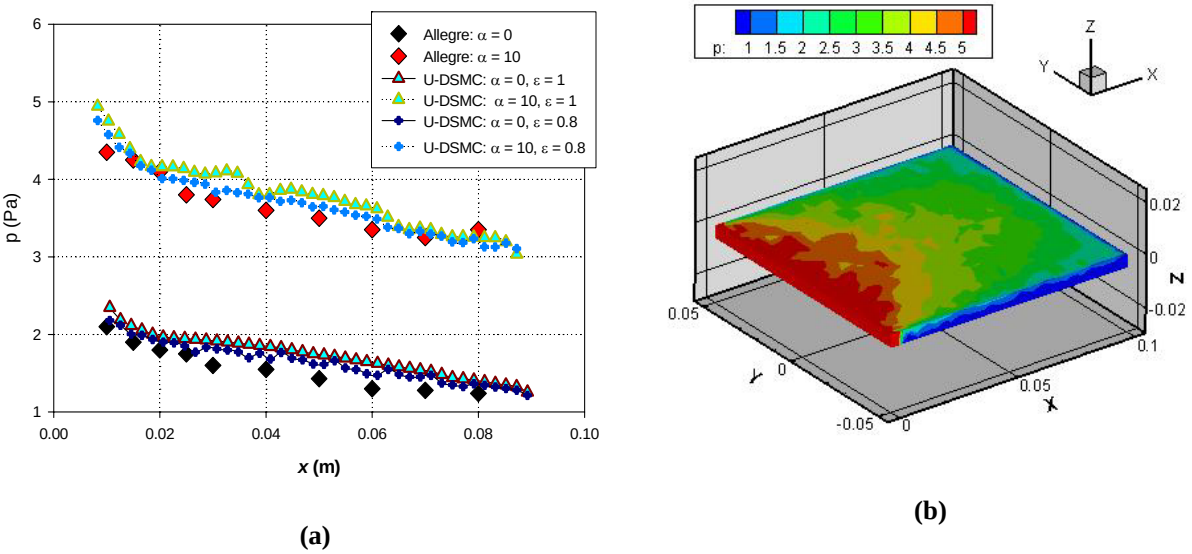
3.3.2 Results

The free stream conditions, as well as the surface temperature of the plate, that have been applied in this investigation are tabulated in Table 1. For each angle of attack, U-DSMC simulations were carried out using both fully diffuse and non-diffuse surface models. The nitrogen gas is modeled using the VHS model with active rotational internal degrees of freedom. Each simulation was allowed to run until steady state had been reached. Once this occurred, time-averaged sampling was carried out over several hundred iterations.

Table 1. Parameters used in flat plate simulations.

Re_∞	Ma_∞	P_∞	n_∞	T_∞	V_∞	λ_∞	T_{wall}
2800	20.2	0.6831 Pa	$3.716\times10^{20}\ \mathrm{m}^{-3}$	13.32 K	$1503\ \mathrm{m/s}$	$2.35\times10^{-3}\ \mathrm{m}$	290 K

The first data set used for validation comparison is that of local pressure along the surface of the plate. The resulting comparison plot of the pressure along the centerline of the plate is shown in Figure 37 (a). Close inspection of the data reveals a small bias in the U-DSMC results for the fully diffuse boundary conditions. Although the non-diffuse case displays a reduced bias the bias is still rather clear. A bias between experimental results and structured DSMC predicitions was also found in the work of Allegre et al (1993) and is credited to the definition of pressure used in the experimental measurements.

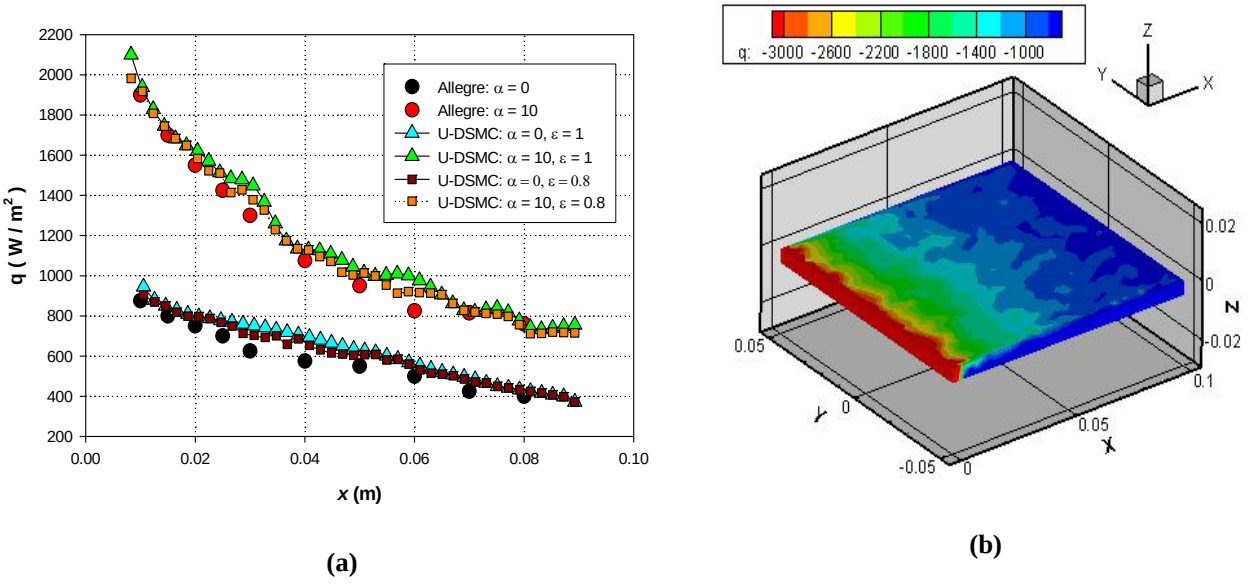


(a) **(b)**

Figure 37. Pressure on the plate surface as a function of distance along the centerline (a) and pressure contours along the surface of the plate (b).

The flow visualization capabilities as pertaining to surface transport properties can be seen in Figure 37 (b). The pressure distribution along the surface of the plate for a case with α = 10 degrees illustrates the high pressure region towards the leading edge of the plate. The steady decrease of the pressure along the plate in the flow direction is also clearly visible.

Figure 38 (a) shows the comparison of the U-DSMC predictions for centerline heat flux with the experimental measurements. Excellent agreement is seen between the experimental heat transfer measurements and the U-DSMC predictions for both the fully diffuse case as well as the non-diffuse case. The local heat transfer distribution along the plate surface is shown in Figure 38 (b).



(a) **(b)**

Figure 38. Heat flux as a function of the distance along the centerline (a) and heat flux contours along the surface of the plate (b).

Further validation of the code is demonstrated in the flow field around the plate for the case with α = 10 degrees and fully diffuse modeling of the plate surface. Figure 39 (a) shows

number density contours normalized with respect to the free stream value. The contours clearly illustrate the development of a boundary layer over the top surface of the plate. The development of the boundary layer is a collision based effect and thus is an additional visual indication that the current U-DSMC implementation is properly capturing collision based momentum exchange.

In Figure 39 (b) the x-component velocity contours, as well as velocity stream traces, are shown for the case with α = 10 degrees and non-diffuse modeling of the plate surface. Once again, the contours clearly indicate the formation of a boundary layer over the top surface of the plate. The entrance angle of the free stream is clearly indicated in the stream traces.

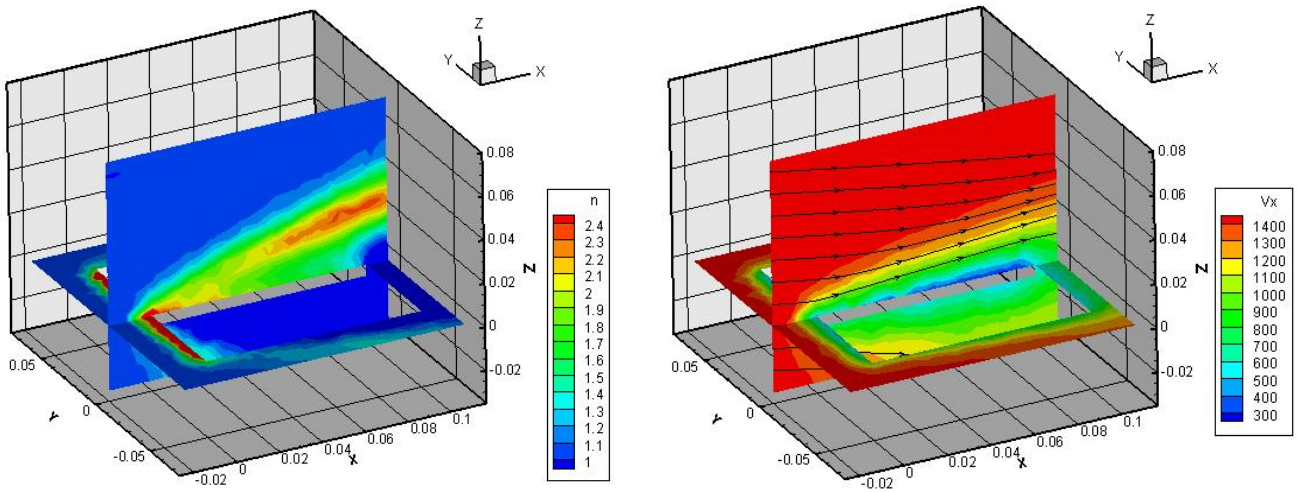


Figure 39. Flow field contours of normalized number density (a) and x-component velocity (b).

3.3.3 Conclusions

The implementation of the inelastic collision algorithm within U-DSMC is verified from the good agreement seen between the numerical results and the experimental data. Furthermore, it can be concluded that gas-surface interactions are modeled effectively. Also, this test case

demonstrates that the particle injection algorithm used to model the free stream is implemented properly. Finally, from investigation of the local values of the flow field, verification of the macroscopic sampling routines can be concluded.

3.4 Free Jet Expansion

The next test case involves U-DSMC modeling of gaseous expansion and comparison between numerical results and theoretical formulations. In this test case the local Knudsen number varies from 0.1 to well over 10. Therefore free jet expansion tests the current implementation's capability of capturing Knudsen regimes spanning from slip to free molecular within in a single computational domain. This case also verifies molecular flux injection with upstream-inlet free boundaries, downstream-vacuum free boundaries, mass flow rate and number flux calculations.

3.4.1 Geometry and Boundary Conditions

The simulation domain, shown in Figure 40, features an internal orifice region and a near field plume expansion region. Particle injection along the inlet of the orifice is modeled as a molecular flux of an equilibrium gas. As such, the pressure, temperature and drift velocity of the equilibrium distribution at the boundary are chosen at initialization and enforced throughout the simulation. To aid in the development of a corresponding equilibrium distribution in the region of the inlet surface at steady state, any particles that move upstream along the inlet surface are removed from the flow. The walls of the orifice are modeled as fully diffuse and the temperature of the wall is chosen to equal that of the inlet distribution. A hard vacuum was chosen to model the boundaries of the plume region. The working gas in the simulation was argon.

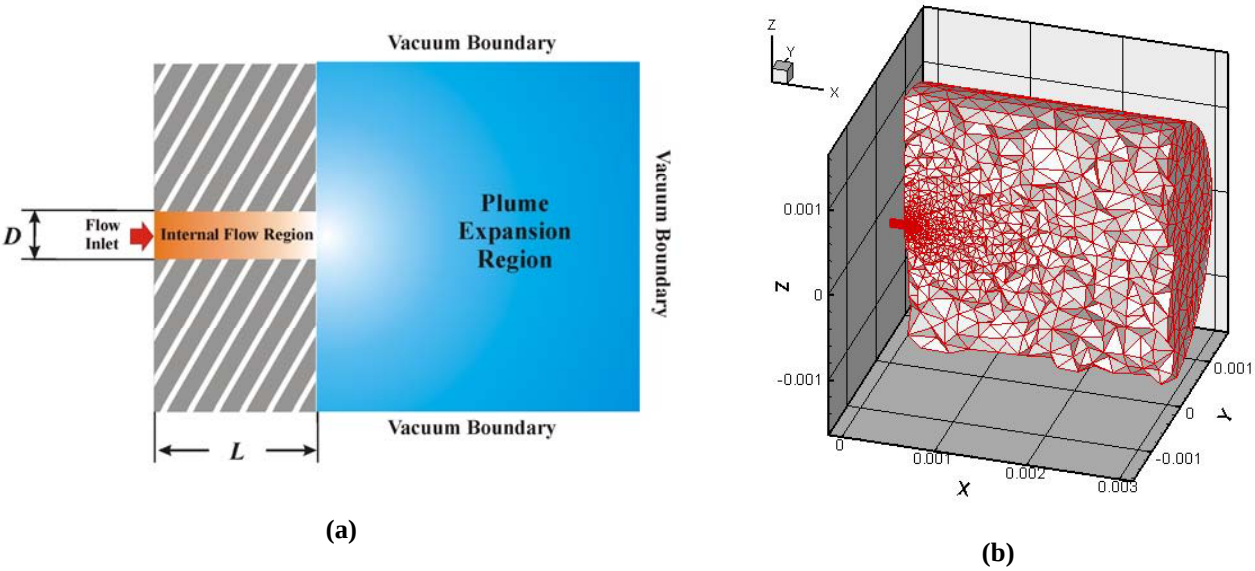


Figure 40. Geometry and boundary conditions for free jet expansion test case (a) and an example grid from the free jet simulations (b).

3.4.2 Results

The input parameters used in the simulation correspond closely to the limiting values for the underlying assumptions of the theoretical formulations and are shown in Table 2. The inlet Knudsen number is based on the VHS formulation given by (Bird, 1994):

$$Kn = \frac{\lambda}{D} = \frac{1}{\sqrt{2}\pi {d_{ref}}^2 n \left(T_{ref}/T\right)^{\omega-1/2}} \frac{1}{D} \tag{3.11}$$

where λ is the mean free path, D is the orifice diameter, d_{ref} is the reference diameter of the VHS molecule, T_{ref} is the reference temperature for d_{ref}, and ω is the viscosity index for the VHS molecule. The orifice diameter is 100 μm and the aspect ratio is 1.5. The inlet injection is sampled from a distribution with $T_I = 300\,\mathrm{K}$, $n_I = 1.29\times10^{23}\,\mathrm{m}^{-3}$, and an axial drift of $V_I = 13.5\,\mathrm{m/s}$. Using the VHS based formulation for the Reynolds number (Bird, 1994):

$$Re = \frac{2(5-2\omega)(7-2\omega)}{15\pi^{1/2}} \frac{S}{Kn} \tag{3.12}$$

where S is the speed ratio, the resulting inlet Reynolds number for this case is $\mathrm{Re}_I = 0.5$. The orifice wall temperature is fixed at $T_W = 300\,\mathrm{K}$.

Table 2. Selected parameters for the expansion test case.

$n_I\,(\mathrm{m}^{-3})$	$V_I\,(\mathrm{m/s})$	$D\,(\mu\mathrm{m})$	L/D	Kn	Re
1.29×10^{23}	13.5	100	1.5	0.1	0.5

Three formulations are used for verification of the simulation results. The first relation used was developed by Ashkenas and Sherman (1966) from a data fit of their method-of-characteristics calculations for the inertia-dominated region of free jet expansion:

[Ashkenas & Sherman] $$\frac{\rho(R,\theta)}{\rho(R,0)} = cos^2\left(\frac{\pi\theta}{2\phi}\right) \tag{3.13}$$

where R is the radial distance from the orifice exit plane and θ is the angle from the jet axis. The parameter ϕ was found by Ashkenas and Sherman to be a function of the ratio of specific heats (γ). The corresponding value of ϕ for argon is given as 1.365. Ashkenas and Sherman found this relation accurate to within 3% of the numerical data. Further validation of this formula was given in a study by Dettleff and Plahn (1998) where comparison between the Ashkenas and Sherman relation and experiment was found to differ between 10% and 20% for $0^o \le \theta \le 55^o$.

The second relation used to verify the simulation results is an approximate formula developed by Boynton (1967) from a data fit of results derived from computations of rocket exhaust-plume flow fields at high altitude. Boynton's expression was also used in the work by

Dettleff and Plahn (1998) and was found to differ from experiment by roughly 10% to 30% for $0^o \leq \theta \leq 55^o$. The form of Boynton's formula used by Dettleff and Plahn (1998) is:

[Boynton] $$\frac{\rho(R,\theta)}{\rho(R,0)} = cos^{\frac{2}{\gamma-1}}\left(\frac{\pi\theta}{2\theta_{lim}}\right) \tag{3.14}$$

where R is the radial distance from the orifice exit plane, θ is the angle from the jet axis and θ_{lim} is the Prandtl-Meyer angle.

The third relation is a data fit similar to Boynton's that is used by Albini (1965) and Hubbard (1966):

[Albini & Hubbard] $$\frac{\rho(R,\theta)}{\rho(R,0)} = cos^{\frac{1}{\gamma-1}}\left(\frac{\pi\theta}{2\theta_{lim}}\right) \tag{3.15}$$

Figure 41 shows normalized density profiles in the expansion plume. A value of $R = 1\,\text{mm}$, which corresponds to an axial distance from the exit plane (x) equal to ten tube diameters, was used to sample the angular profile shown in the figure. The number density values at each angle were normalized by the centerline number density n_0. Excellent agreement between the U-DSMC results and the three formulations is found. Over angular ranges of $-45^o \leq \theta \leq 45^o$ the difference between the U-DSMC results and Eq. (3.13) varies from 1% to 8%, while the difference between U-DSMC results and Eq. (3.14) ranges from 3% to 10%. The error for Eq. (3.15) is a bit higher for each point over the same range but is still less than 10%, varying from roughly 6% to 10%.

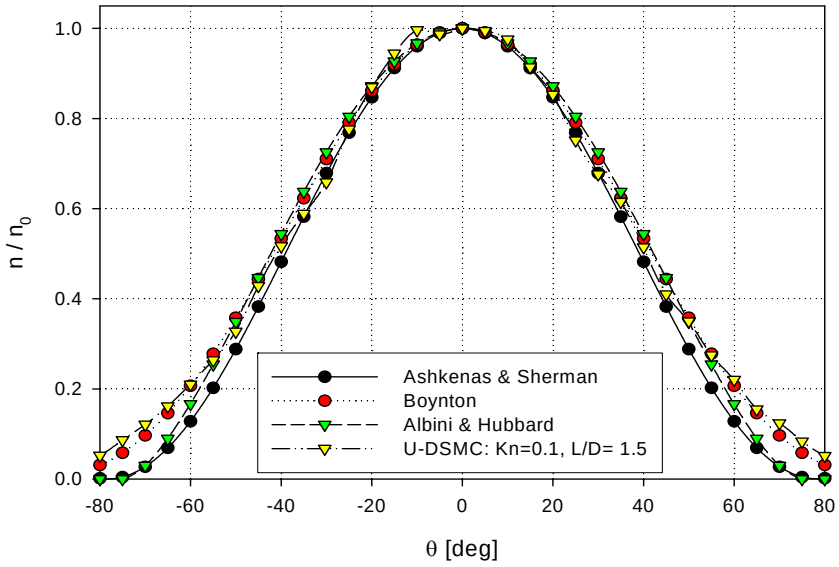


Figure 41. Comparison of U-DSMC results with theoretical formulations at a radial distance of *R* = 1mm.

The overall structure of the jet is shown in Figure 42 (a). The number density contours illustrate the development of a fully expanded plume. In Figure 42 (b) the x-component velocity contours are shown for the free jet expansion. Once again, the contours clearly indicate the full expansion of the free jet plume.

3.4.3 Conclusions

From the agreement shown above it can be concluded that the current implementation of the U-DSMC method properly captures gaseous expansion. This case verifies the molecular flux injection with upstream-inlet free boundaries and the implementation of downstream-vacuum free boundaries. Furthermore, the code demonstrates its capability to handle a wide Knudsen range within a single computational domain.

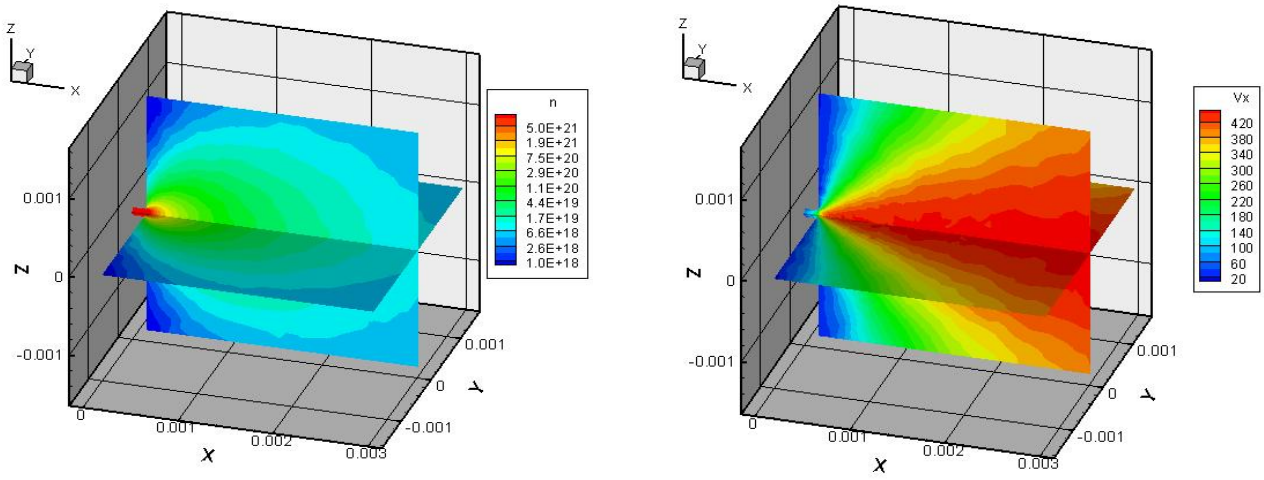


Figure 42. Flow field contours of number density (a) and x-component velocity (b).

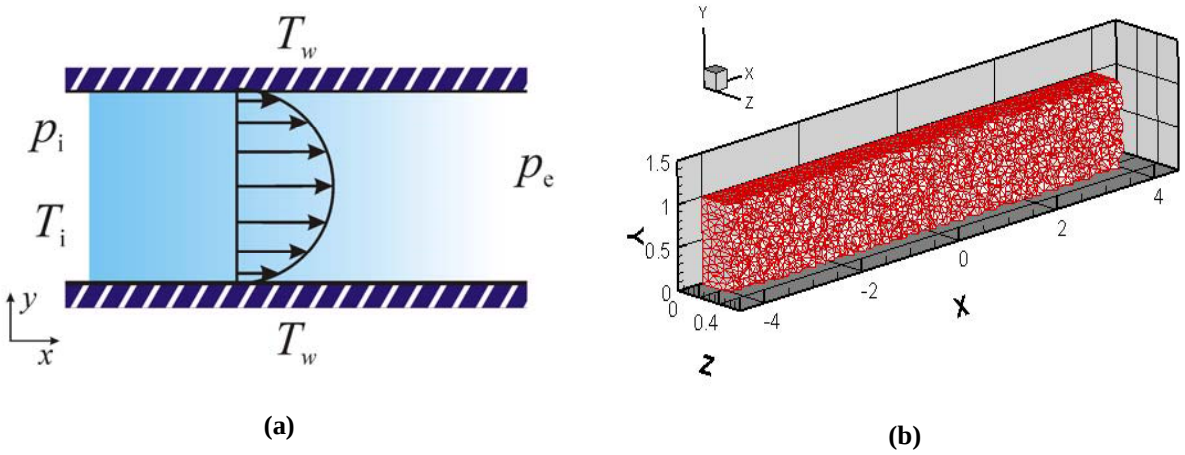
3.5 Poiseuille Flow

The implementation of subsonic boundaries within the current version of U-DSMC is tested by means of two classic flow scenarios. The first scenario is that of pressure driven Poiseuille flow. Due to the nature of the flow conditions, Poiseuille flow requires proper capture of both the pressure inlet as well as the downstream pressure exit. As a result of the internal flow conditions combined with subsonic upstream disturbances, subsonic Poiseuille flow is a demanding test case that will require accurate implementation of both the upstream fixed-pressure, fixed-temperature subsonic boundary condition as well as the downstream fixed-pressure condition. Two theoretical formulations are used to verify the U-DSMC handling of pressure-driven subsonic flow.

3.5.1 Geometry and Boundary Conditions

Poiseuille flow is a pressure-driven flow between two infinite stationary parallel plates. As illustrated in Figure 43 (a), the inlet surface is maintained at a fixed pressure p_i and

temperature T_i while the outlet is maintained at a fixed pressure p_e with the pressure ratio along the flow path given as $p_{ratio} = p_i / p_e$. The plate walls are fixed at a temperature T_w. The infinite parallel plates are modeled as a channel with fully diffuse upper and lower walls and the side walls are modeled as specularly reflecting thus effectively acting as symmetry planes. The grid used in the simulations in shown in Figure 43 (b).



(a) **(b)**

Figure 43. Geometry and boundary conditions for Poiseuille flow test case (a) and an example grid from the Poiseuille flow simulations (b).

3.5.2 Results

The computational parameters are shown in Table 3. The distance between the parallel plates was chosen as $h = 1\,\mathrm{m}$. The pressure applied at the inlet was chosen to be 5.34 Pa, while the outlet pressure has been fixed to 0.534 Pa thus resulting in a pressure ratio across the flow path equal to $p_{ratio} = 10$. The gas used for the current study is argon. The upper and lower domain boundaries, which correspond to the plate surfaces, are modeled as diffuse with full accommodation to the wall temperature of $T_w = 300\,\mathrm{K}$. The subsonic inlet boundary condition

was applied at the inlet surface with $\lambda_{coef} = 1$ and the time averaged face-normal velocity component was recalculated every 10 iteration steps. The subsonic outlet boundary condition was applied at the outlet surface with $\lambda_{coef} = 1$ with time averaging also occurring every 10 iteration steps. The computational domain is initially loaded with a background field corresponding to a pressure of 0.4 Pa.

Table 3. Flow parameters used in the Poiseuille flow test case.

$p_i\,(\mathrm{Pa})$	p_{ratio}	$n_i\,(\mathrm{m}^{-3})$	$h\,(\mathrm{m})$	Kn	Re	S
5.34	10	1.29×10^{21}	1	0.001-0.01	300-900	0.2-0.6

The progression of the simulation results, as driven by the implementation of the subsonic boundary conditions, is of significant importance in determining the viability of U-DSMC to model subsonic flows. The flow field development as a function of simulation time is shown in Figure 44. As seen in the left hand side of Figure 44, the development of the x-component velocity profile is composed of various stages. Initially the pressure gradient across the inlet surface is significant following:

$$\frac{dp}{dx} = \frac{p_j - p_i}{\Delta x} \approx \frac{0.4 - 5.34}{\lambda} \approx -5000\,\mathrm{Nm} \tag{3.16}$$

Due to this large localized pressure gradient, the subsonic boundary condition over compensate with a large slug of high speed particles. The slug continues to propagate through the computational domain. By means of inter-molecular collisions, the kinetic energy of the slug particles is transferred to thermal energy as the slug front interacts with the loaded background field, as shown in the right hand side of Figure 44. After the initial over-shoot period, the local pressure gradient at the inlet begins to settle toward the steady-state solution and as such the

injected particles have smaller x-component velocities. Likewise, the translational temperature of the flow field begins to settle toward the steady state value.

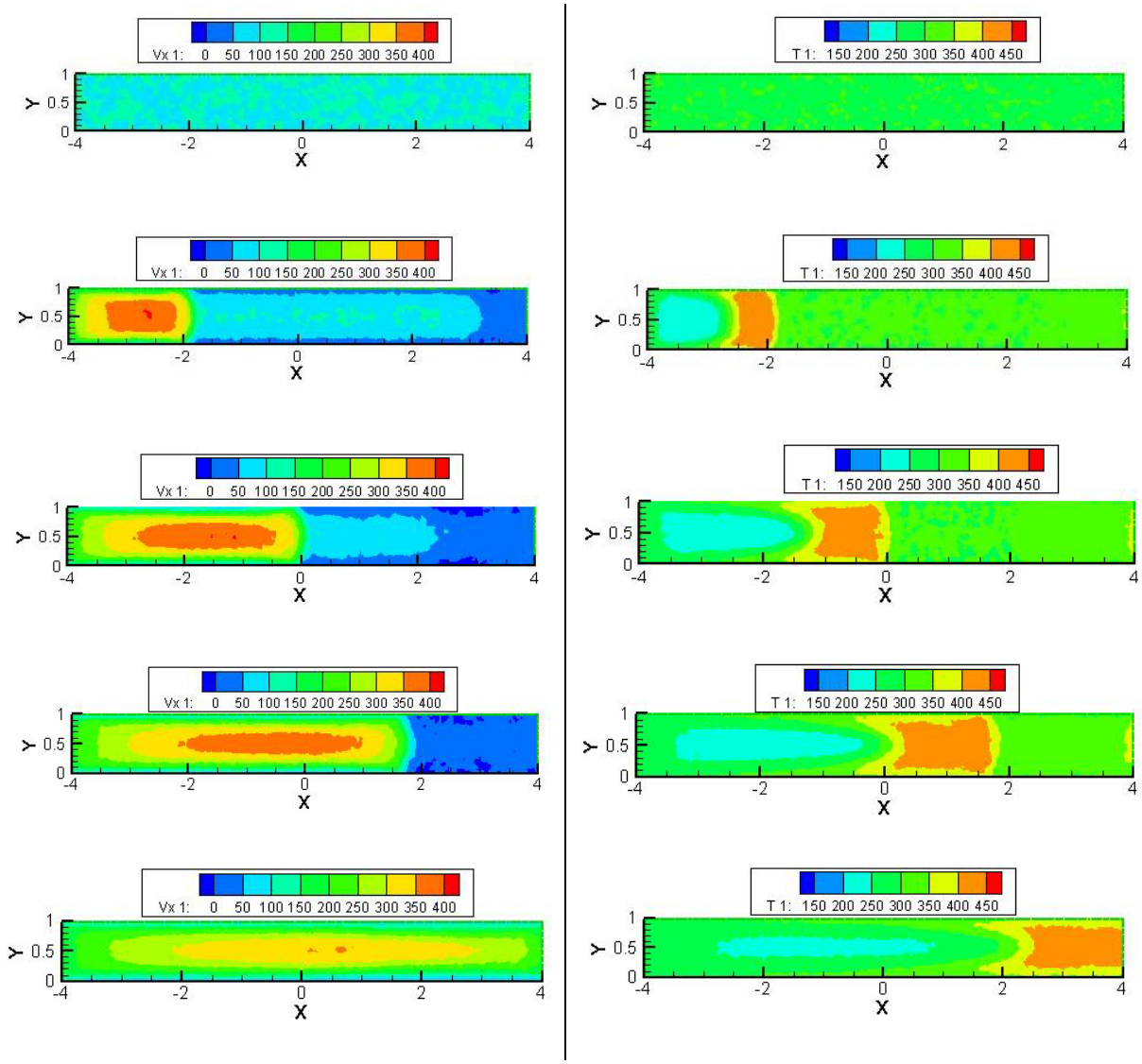


Figure 44. Flow field development as a function of simulation time for subsonic Poiseuille flow.

Following the initial transient response of the boundary conditions a steady state solution is reached. Although previous implementations of characteristic-based subsonic boundary conditions in structured DSMC have been reported to suffer from significant fluctuations of the

sample-based injection parameters, the averaging technique applied in the current implementation appears to dampen this fluctuation.

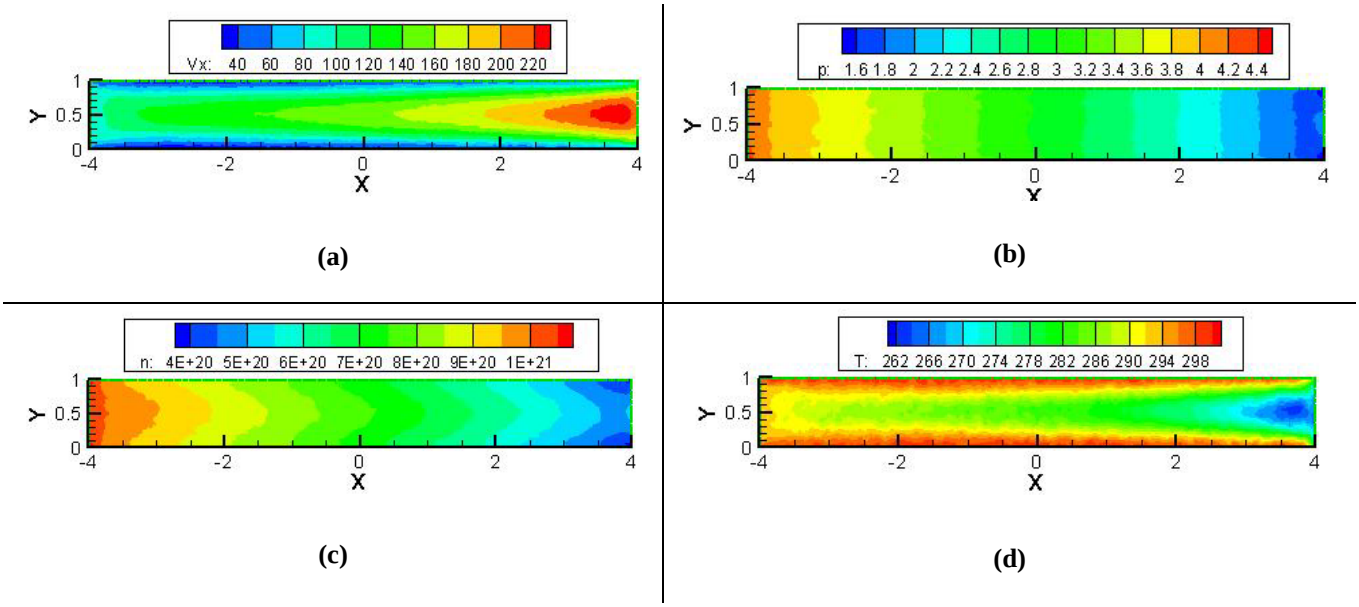


Figure 45. Time averaged steady state values of x-component velocity (a), pressure (b), number density (c) and translational temperature (d).

The steady-state flow field is shown in Figure 45. Figure 45 (a) displays the clear development of a velocity profile between the parallel plates. Similarly, the number density contours shown in Figure 45 (c) illustrate the development of a flow structure along the plate walls. Figure 45 (d) shows the contours of translational temperature at steady state. Once again a clear flow structure can be observed. However, it should be noted that the contours of Figure 45 (d) depict the thermal temperature of the molecules with respect to the local stream velocity therefore the contours do not indicate the development of true thermal boundary layers, as would be observed if the wall temperature was significantly different from the inlet stream temperature. However, the current implementation of U-DSMC would be excellent tool to use in investigation of thermal creep and subsonic heat transfer.

In order to quantitatively validate and verify the subsonic implementation, the velocity profile generated from the U-DSMC simulation is compared to analytical formulations based on the Navier-Stokes equations with modified velocity-slip boundary conditions at the walls. For fully developed parallel plate flow, such as Poiseuille flow, with low Reynolds numbers, the Navier-Stokes equations are simplified to:

$$\frac{dp}{dx} = \mu \frac{\partial^2 u}{\partial x^2} \tag{3.17}$$

where dp/dx is the pressure gradient in the x-direction, μ is the fluid's viscosity, and u is the x-component of the local fluid velocity. The first order slip wall boundary condition is taken from Maxwell (Liou and Fang, 2000) and for fully diffuse reflection at the wall can be written as:

$$u - u_w = Kn \frac{du}{dy} \tag{3.18}$$

which is seen to be first order in Kn. A second-order slip boundary condition was proposed by Beskok (1996) and for fully diffuse walls it can be written as:

$$u - u_w = \frac{Kn}{1 + Kn} Kn \frac{du}{dy} \tag{3.19}$$

For pressure driven Poiseuille flow, the first-order relation for the functionality of the local x-component velocity in fully diffuse Poiseuille flow is found to be:

$$u_{1st} = \frac{h^2}{2\mu} \frac{dp}{dx} \left[\left(\frac{y}{h} \right)^2 - \frac{y}{h} - Kn \right] \tag{3.20}$$

where h is the distance between the parallel plates. The second-order functionality of the local x-component velocity in fully diffuse Poiseuille flow is given as:

$$u_{2nd} = \frac{h^2}{2\mu}\frac{dp}{dx}\left[\left(\frac{y}{h}\right)^2 - \frac{y}{h} - \frac{Kn}{1+Kn}\right] \tag{3.21}$$

It is clearly seen in both relations that the velocity profile is a function of the local Knudsen number as well as the pressure gradient and the distance between the parallel plates. The relations above can be written in non-dimensional form with the introduction of a velocity scale. If the centerline velocity at the x-location of interest is taken as the velocity scale then the non-dimensional velocity profiles become:

$$\frac{u_{1st}}{u_c} = \frac{\left[-\left(\frac{y}{h}\right)^2 + \frac{y}{h} + Kn\right]}{\frac{1}{4} + Kn} \tag{3.22}$$

$$\frac{u_{2nd}}{u_c} = \frac{\left[-\left(\frac{y}{h}\right)^2 + \frac{y}{h} + \frac{Kn}{1+Kn}\right]}{\frac{1}{4} + \frac{Kn}{1+Kn}} \tag{3.23}$$

Comparison of the U-DSMC predicted velocity profiles with those of the relations above are shown in Figure 46. Although the U-DSMC results exhibit minor scatter in the near-wall region the excellent agreement between the overall U-DSMC velocity profile and the theoretical formulations is shown in Figure 46. It should be noted that for a Knudsen value as low as 0.002, such as the local value corresponding to the location from which the profile was generated, the flow does not exhibit noticeable slip and therefore the first and second-order slip models predict equivalent velocity profiles.

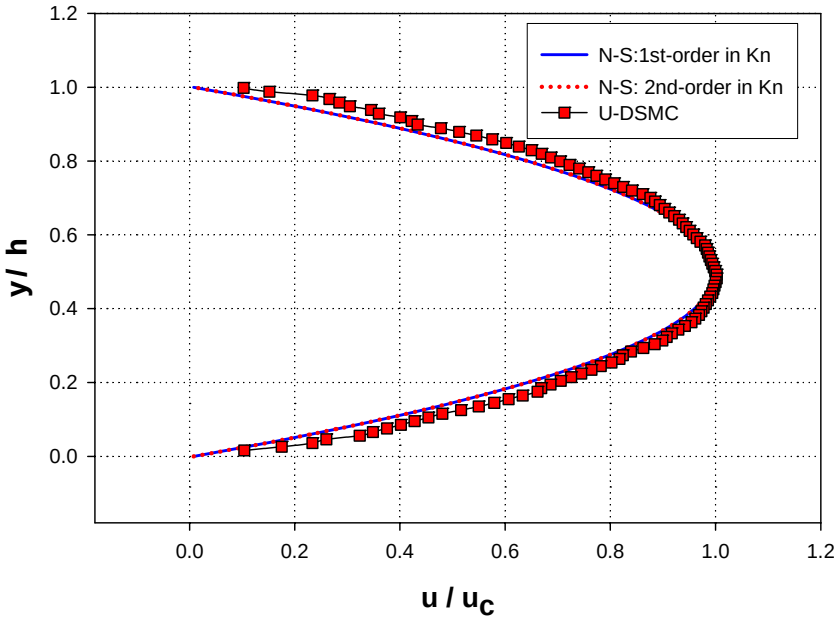


Figure 46. Comparison of U-DSMC predicted velocity profiles with theoretical formulations.

3.5.3 Conclusions

The study of Poiseuille flow using the current implementation of U-DSMC indicates that the implementations of subsonic inlet and exit boundary conditions are fully functional. The direct comparison of the velocity profile generated using U-DSMC with theoretical formulations displays excellent agreement and thus verifies the accuracy of the current implementation. Additionally, the lack of noticeable fluctuation of the sample-based inlet velocity indicates that the averaging technique applied in the subsonic algorithms aids in reducing transient fluctuation of the boundary injection parameters.

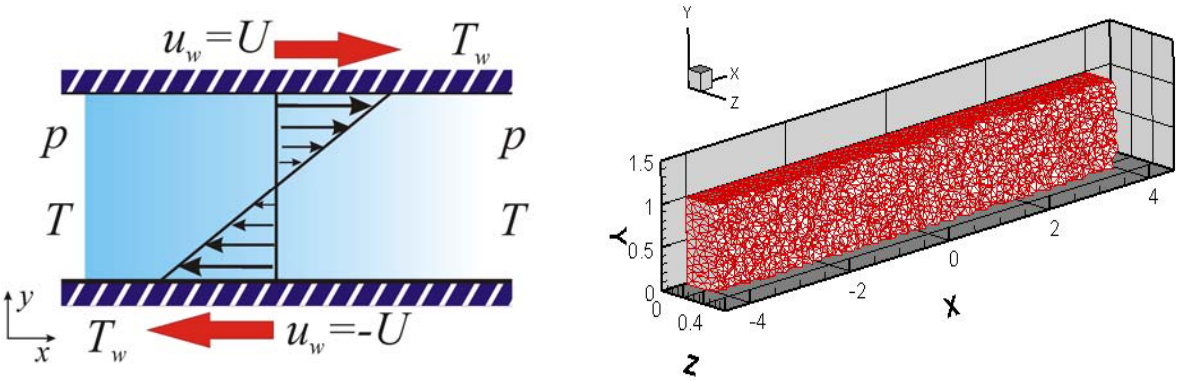
3.6 Couette Flow

The second classic flow scenario utilized to verify the subsonic implementation as well as the moving wall boundary condition is that of Couette flow. The flow conditions of Couette

flow are suitable for determining the overall performance of the current U-DSMC implementation to subsonic flow cases which are not driven by pressure gradients. As such, a theoretical formulation is used to verify the U-DSMC handling of shear-driven subsonic flow.

3.6.1 Geometry and Boundary Conditions

Couette flow is a shear-driven flow between two infinite moving parallel plates. The inlet surfaces are maintained at a fixed pressure p_i and temperature T_i while the local stream velocity must remain floating. The plate walls are fixed at a temperature T_w. The relevant geometry applied to the U-DSMC modeling of Couette flow is shown in Figure 47 (a). In the current test case, both the upper and lower plates are moving with a velocity of $u_w = U$ in the positive x-direction for the upper plate and negative x-direction for the lower plate. The grid used in the Couette flow application is shown in Figure 47 (b).



(a) **(b)**

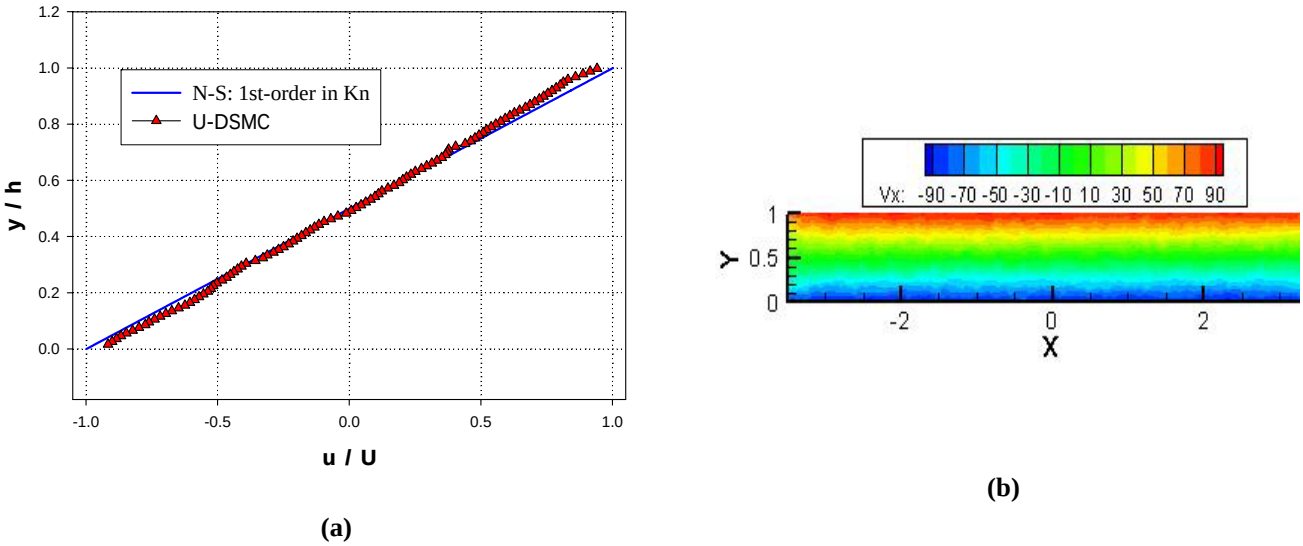
Figure 47. Geometry and boundary conditions for Couette flow test case (a) and an example grid from the Couette simulations (b).

3.6.2 Results

For validation purposes the velocity profile generated from the U-DSMC simulation are compared to analytical formulations based on the Navier-Stokes equations with modified velocity-slip boundary conditions at the walls. For fully diffuse Couette flow with low Reynolds numbers, negligible thermal creep effects, slip-velocity at the wall and motion of the top plate only, integration of the momentum equation results in (Karniadakis and Beskok, 2000):

$$\frac{u(y)}{U_\infty} = \frac{1}{1+2Kn}\left[\frac{y}{h} + Kn\right] \tag{3.24}$$

Extension of this relation to the case where both the top and bottom plate are in motion is achieved by a simple mapping of the reference frame. Comparison of the U-DSMC predicted velocity profile and the theoretical formulation is shown in Figure 48.



(a) **(b)**

Figure 48. Comparison of U-DSMC generated velocity profiles with a theoretical formulation (a) and velocity contours between the parallel plates (b) for Couette flow.

3.6.3 Conclusions

From the excellent agreement shown in Figure 48 it can be concluded that the current implementation of U-DSMC is readily applicable to subsonic flow scenarios that are shear-driven in nature. Consequently, the implementation of the moving wall boundary condition is verified as are the subsonic inlet boundary conditions.

4. U-DSMC Studies of Micro and Nanoflows

The U-DSMC code is applied to the investigation of four studies in order to investigate the nature of rarefied gaseous flows in micro and nano domains. The first application involves the expansion of argon from microtubes into vacuum. The U-DSMC simulations characterize the effects of Knudsen number, Reynolds number, aspect ratio and tube scale on the resulting plume. The second study characterizes the effects of Knudsen number, Reynolds number, speed ratio and scale on micro and nano nozzle expansion. The third application aids the design of a microsensor under development which will be used to study microjet expansion. The last study characterizes the effects of decreasing the scale on statistical fluctuations with regards to subsonic micro and nano flows.

4.1 Gaseous Expansion from Microtubes

The expansion of argon from microtubes into hard vacuum is extensively investigated using U-DSMC. The simulations are used to investigate the effects of Knudsen number, aspect ratio, Reynolds number and microtube scale on plume structure as well as the decay of the number density along the flow path. The following work is an extraction of the findings presented in Chamberlin and Gatsonis (2007) as well as those presented within a Keynote paper at the Fourth International Conference on Nanochannels, Microchannels and Minichannels (Chamberlin and Gatsonis, 2006 (b)).

4.1.1 Geometry and Boundary Conditions

All cases investigated in this section correspond to the geometry indicated in Figure 49. The simulation domains feature the internal microtube region and the near field plume expansion

region. Particle injection along the inlet of the microtube is modeled as a molecular flux of an equilibrium gas. To aid in the development of a corresponding equilibrium distribution in the region of the inlet surface at steady state, any particles that move upstream along the inlet surface are removed from the flow. Furthermore, the inlet flow constraints imposed allow for simple manipulation of the inlet Knudsen number and Reynolds number as is required for the investigation at hand.

The walls of the microtube are modeled as fully diffuse and the temperature of the wall is chosen to equal that of the inlet distribution. This yields the most general relation between the wall temperature and the stagnation chamber temperature and therefore should lend simplicity to experimental validation of the U-DSMC results.

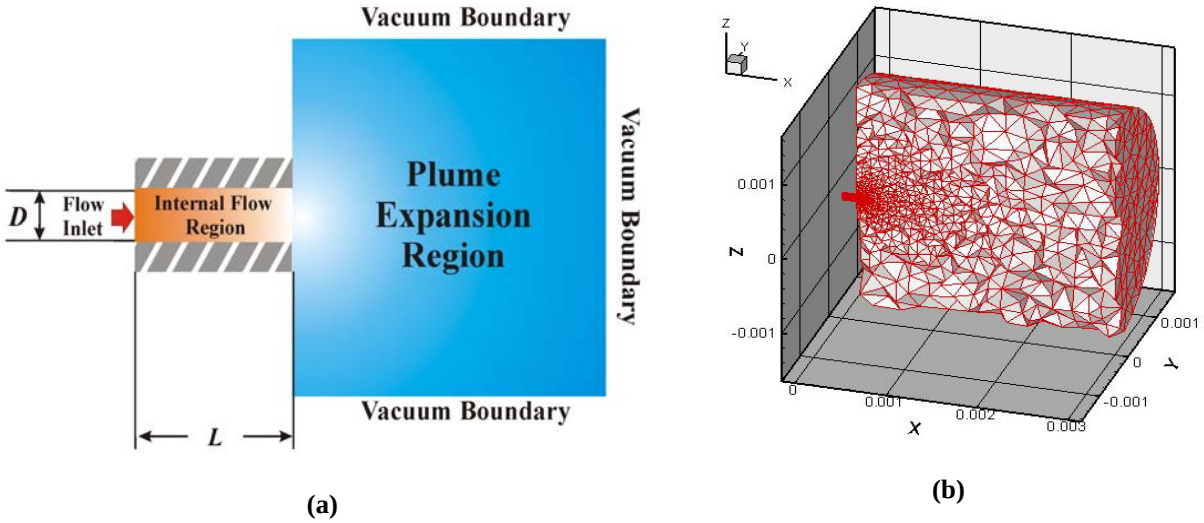


Figure 49. Geometry and boundary conditions applied in microtube study (a) and an example grid used for microtube simulations (b).

The final boundary conditions required for the simulations are the downstream plume region boundaries. To keep the results as general as possible a hard vacuum was chosen to model the boundaries of the plume region. Although a backpressure boundary condition would

be a better match for experimental investigation, the minimum achievable backpressure for each case investigated would vary with each facility. However, future investigations are planned which will study the effect of backpressure on the plume properties.

An example grid used in the microtube expansion simulations is shown in Figure 49 (b). Although the grid spacing varies for each case a few relations remain constant amongst all cases. The first relation that is guaranteed for all grids is that the local cell spacing is always smaller than the local mean free path. This is a widely accepted limit that, when coupled with a properly chosen time step and a sufficient number of simulated particles in each cell, ensures physical collision pair spacing and thus statistically accurate modeling of transport phenomena (Garcia and Wagner, 2000; Hadjiconstantinou, 2000). The second cell spacing criteria common to all grids is sizing bounded by geometric resolution. In certain cases investigated in this work the local mean free path is much larger that the microtube diameter. When this relation holds the cell spacing is set to the maximum size that still leads to a sufficient capturing of the tube curvature.

It should be noted that for small Reynolds number flows DSMC results exhibit significant numerical scatter. Furthermore statistical uncertainty is known to be large for small Reynolds number and speed ratio simulations (Hadjiconstantinou et al, 2003). In order to reduce the statistical error in the results shown here, extensive time averaging of the data has been carried out for each case studied. Additionally, the number of simulation particles in each computational collision cell is maintained at well over twenty, while macroscopic parameters are calculated over volumes featuring at least 100 simulation particles. Although the computational cost of these features is considerable the statistical error and random noise in the resulting data is drastically reduced, even when using a standard DSMC implementation.

4.1.2 Establishment of a Basis Parameter Set

As a means of establishing a set of basis parameter values simulations were made with an aspect ratio approaching the free jet limits. Case 1 has an inlet Knudsen number of 0.1, based on Eq. (3.11). Using Eq. (3.12), the resulting inlet Reynolds number for this case is $\mathrm{Re}_I = 0.5$. The orifice diameter is 100 μm and the aspect ratio is 1.5. The inlet injection is sampled from a distribution with $T_I = 300\mathrm{K}$, $n_I = 1.29 \times 10^{23}\,\mathrm{m}^{-3}$, and an axial drift of $V_I = 13.5\,\mathrm{m/s}$. The tube wall temperature is fixed at $T_W = 300\mathrm{K}$.

Three formulations are used to define basis values of the governing parameters which yield a standardized plume profile. The theoretical formulations are further used to serve as reference points for comparison purposes. Details of the formulations have been given in Section 3.4. The relations are repeated below for the simplified referencing.

[Ashkenas & Sherman]
$$\frac{\rho(R,\theta)}{\rho(R,0)} = cos^2\left(\frac{\pi\theta}{2\phi}\right) \tag{3.13}$$

where $\phi = 1.365$ for argon.

[Boynton]
$$\frac{\rho(R,\theta)}{\rho(R,0)} = \cos^{\frac{2}{\gamma-1}}\left(\frac{\pi\theta}{2\theta_{\mathrm{lim}}}\right) \tag{3.14}$$

[Albini & Hubbard]
$$\frac{\rho(R,\theta)}{\rho(R,0)} = \cos^{\frac{1}{\gamma-1}}\left(\frac{\pi\theta}{2\theta_{\mathrm{lim}}}\right) \tag{3.15}$$

In Figure 50 the close agreement between the U-DSMC results and the formulations is displayed. Excellent agreement between the U-DSMC results and the three formulations is found for case 1. A point-to-point RMS error was calculated for the U-DSMC results with respect to each theoretical formulation following:

$$RMS_{pp} = \sqrt{\left\{\left\langle \frac{n(R,\theta)}{n_0} \right\rangle_{U-DSMC} - \left[\frac{n(R,\theta)}{n_0}\right]_{th}\right\}^2} \tag{4.1}$$

Over angular ranges of $-45^o \leq \theta \leq 45^o$ the point-to-point error between the U-DSMC results and Eq. (3.13) varies from 1% to 8%, while the difference between U-DSMC results and Eq. (3.14) ranges from 3% to 10%. The error for Eq. (3.15) is a bit higher for each point over the same range but is still less than 10%, varying from roughly 6% to 10%.

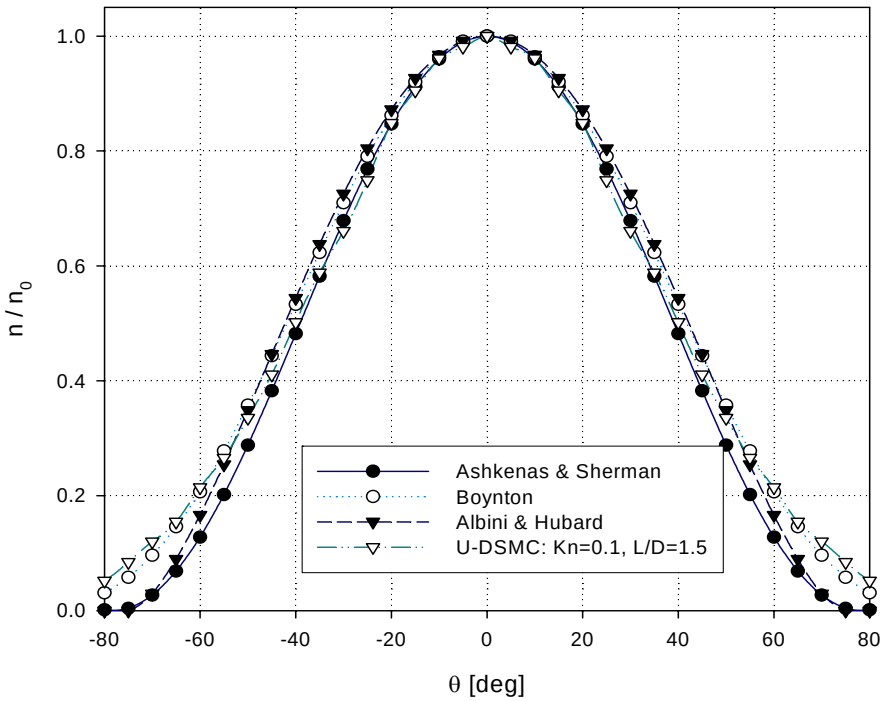


Figure 50. Comparison of U-DSMC results with theoretical formulations at an axial distance of x/D=10.

A value of $R = 1\,\mathrm{mm}$, which corresponds to an axial distance from the exit plane (x) equal to ten tube diameters, was used to sample the angular profile shown in Figure 50. The number density values at each angle were normalized by the centerline number density n_0. For all cases in this work the flow is comprised of a single species, argon, and thus the normalized density is equal to the normalized number density. Due to the excellent agreement between the resulting plume profile and the theoretical formulations, the parameter values of case 1 will be used as basis values in the following comparisons.

4.1.3 Effect of Knudsen Number

The second set of parameters simulated corresponds to an increase in the inlet Knudsen number by increasing the number density. The simulation parameters for case 2 are given in Table 4. The goal of this simulation is to provide insight into role that the Knudsen number plays in the plume development. To this end, the Reynolds number has been set to the value used in case 1. In order to hold the Reynolds number constant while increasing the Knudsen number the speed ratio has to be increased. It should be noted that within the present work the Reynolds number and Knudsen number are assumed to be the key nondimensional parameters of the flow. Due to the interdependence shown in Eq. (3.12), the speed ratio is assumed to be a secondary parameter. Future work is planned to fully test the validity of this assumption.

Table 4. Input parameters used for investigating the effect of Knudsen number on plume characteristics.

Case	$n_I\,(\mathrm{m}^{-3})$	$V_I\,(\mathrm{m/s})$	$D\,(\mu\mathrm{m})$	L/D	Kn	Re
1	1.29×10^{23}	13.5	100	1.5	0.1	0.5
2	1.29×10^{22}	135	100	1.5	1	0.5
3	1.29×10^{21}	1350	100	1.5	10	0.5

The resulting profile of case 2 is plotted in Figure 51. Although the U-DSMC data exhibits some numerical scatter a clear narrowing of the angular profile can be seen. For case 2, the error between the U-DSMC data and Eq. (3.13) varies between 7% and 23% over $-45^o \leq \theta \leq 45^o$, while the error between U-DSMC results and Eq. (3.14) ranges from 14% to 33%. It should be noted that Eq. (3.14) and Eq. (3.15) can be adjusted slightly by varying $\theta_{\lim}$ to improve the correlation with the U-DSMC results. However, even with adjustment, Eq. (3.14) and Eq. (3.15) fail to capture the plume shape seen in the U-DSMC results over an angular range greater than $-20^o \leq \theta \leq 20^o$ with an error less than 10%. This discrepancy indicates a deviation in the resulting plume shape from the standard shape commonly experienced in inertia-dominated continuum-regime expansion.

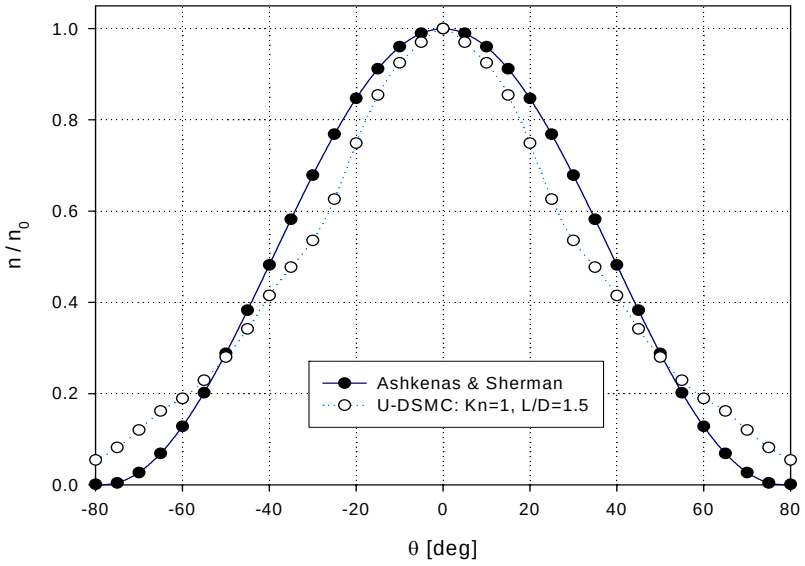


Figure 51. Comparison of U-DSMC results with the theoretical formulation of Eq. (3.13) for case 2 at x/D=10.

The third set of simulation parameters extends the Knudsen number investigation. The Knudsen number is increased to 10 by decreasing the inlet number density by a factor of 10. The simulation parameters are given in Table 4.

The resulting plume is shown in Figure 52. A clear narrowing of the angular profile can be seen. The error between the U-DSMC data and the theoretical formulations are significantly higher than those seen in case 2. Clearly the formulations fail to accurately capture the narrowing effect that occurs with increasing Knudsen number.

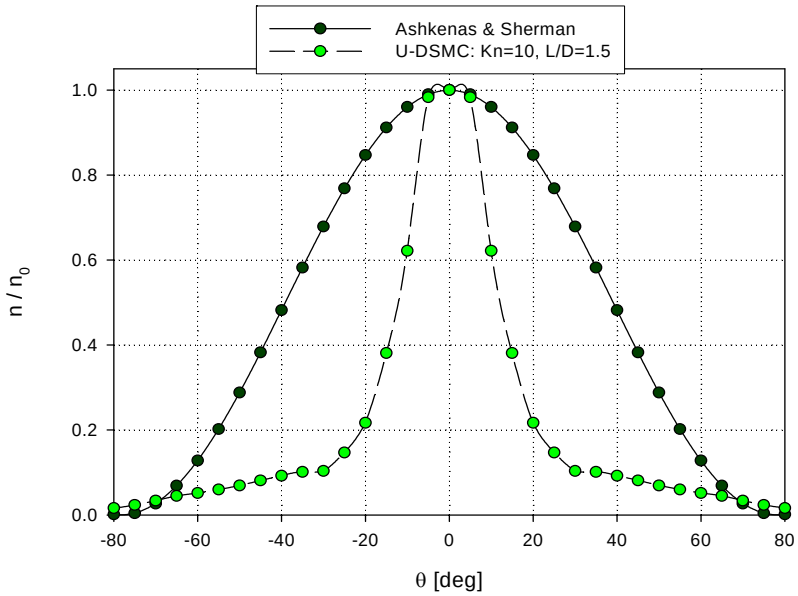
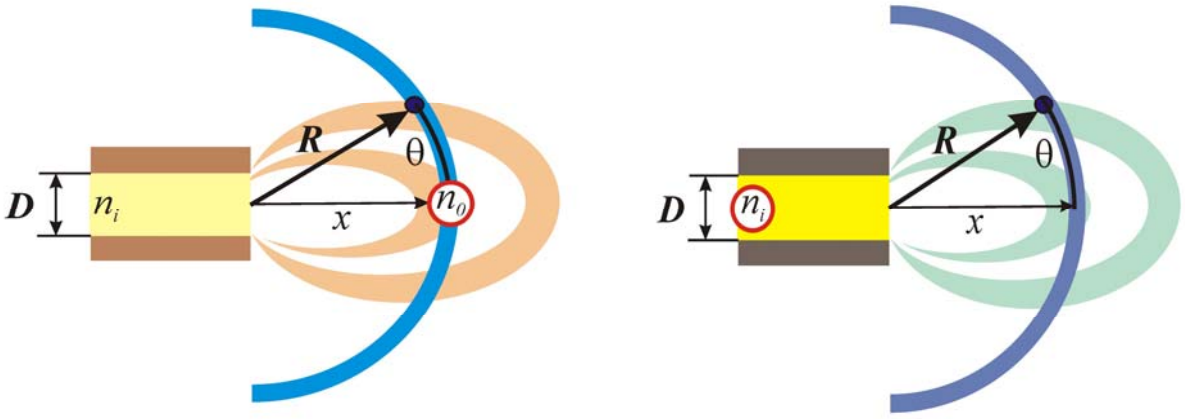


Figure 52. Comparison of U-DSMC results with the theoretical formulation of Eq. (3.13) for case 3.

The results of the first three simulations indicate that there is a relation between the Knudsen number of the flow and the resulting plume shape. In order to visualize the effect of Knudsen number, the angular number density is sampled for each case and scaled with respect to both the centerline value n_0 and the inlet value n_I, as illustrated in Figure 53.



(a) **(b)**

Figure 53. Illustration of the sampled curves, with plume shape defined using centerline normalization (a) and drop in relative number density drop defined using inlet value scaling (b).

A narrowing of the plume is seen to occur as the Knudsen number is increased. This effect is displayed in Figure 54, where the angular profiles, normalized with respect to the centerline value, are plotted together for case 1, case 2 and case 3. Although there is some numerical scatter in the results, a clear shifting of the plume shape is seen to occur between each case.

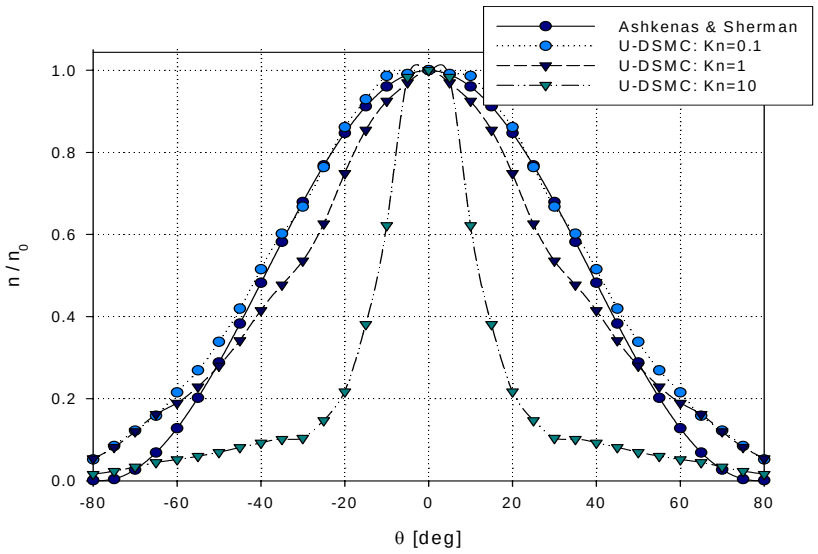


Figure 54. Plume narrowing with increasing Knudsen number.

By plotting the angular profiles normalized by the inlet number density, as in Figure 55, a second effect of the Knudsen number is illustrated. As the Knudsen number increases the effective drop in number density along the centerline decreases. This result can be attributed to collision based effects that decrease with increasing Knudsen number.

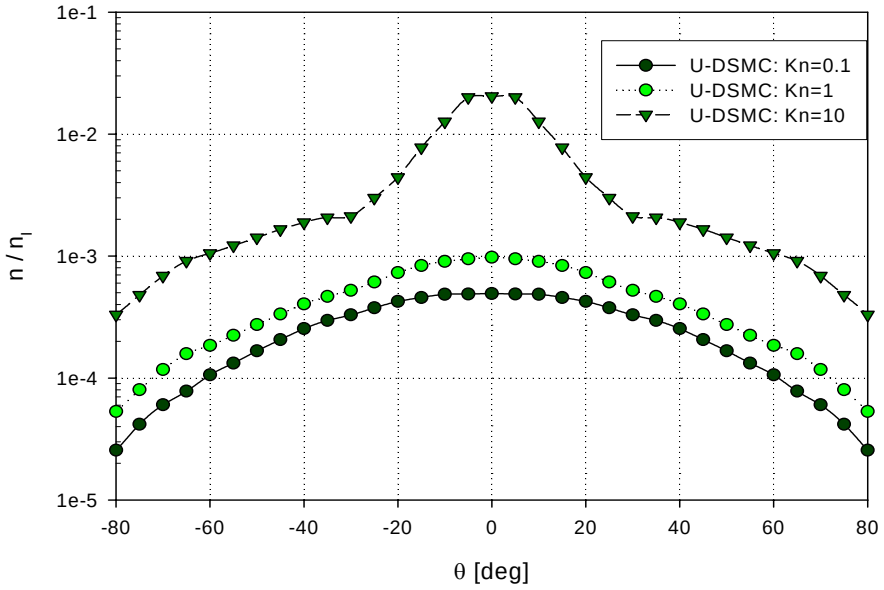


Figure 55. Decrease of the effective number density drop along plume centerline with increasing Knudsen number.

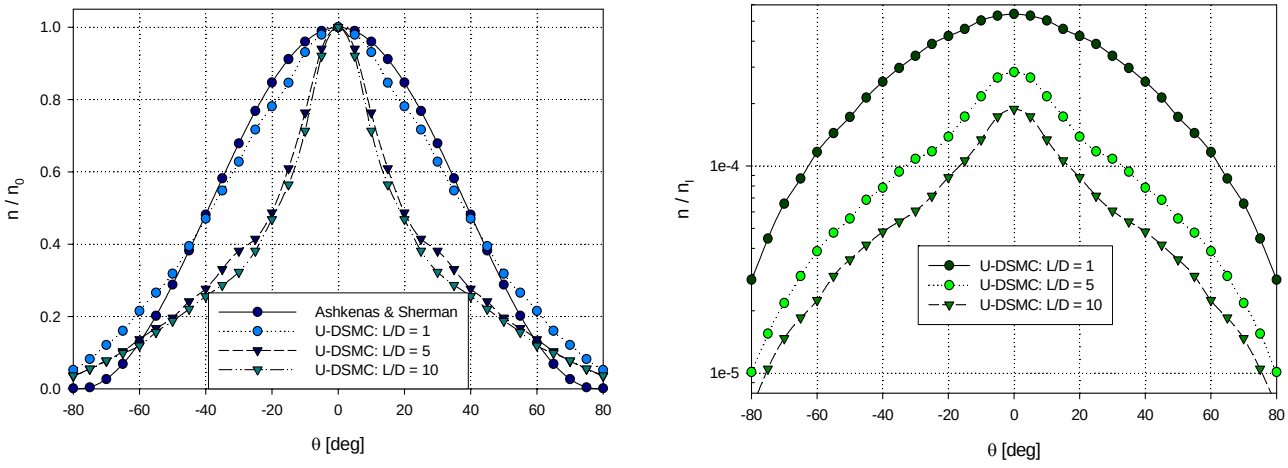
4.1.4 Effect of Aspect Ratio

The variations in plume characteristics with aspect ratio are investigated for $Kn = 1$ in cases 4-6, as shown in Table 5. Each microtube simulated has a diameter of $10\,\mu\mathrm{m}$. The aspect ratio is varied over a range from 1 to 10, with all other parameters kept constant.

Table 5. Selected parameters used for investigating the effect of aspect ratio on plume characteristics.

Case	$n_I\,(\mathrm{m}^{-3})$	$V_I\,(\mathrm{m/s})$	$D\,(\mu\mathrm{m})$	L/D	Kn	Re
4	1.29×10^{23}	5	10	1	1	0.02
5	1.29×10^{23}	5	10	5	1	0.02
6	1.29×10^{23}	5	10	10	1	0.02

The resulting angular profiles taken at an axial distance from the exit plane corresponding to $x/D=10$ are displayed in Figure 56 (a). For $L/D=1$ the resulting plume shape is very close to the formulation for a free jet. The slight shift from the free jet formulation is attributable to the Knudsen value as seen in the previous section. For larger aspect ratios a very distinct sharpening of the plume is seen to occur. Furthermore, it seems the severity of the plume narrowing increases with increasing aspect ratio. Additionally, the results suggest a nonlinear relation between the plume shape and the aspect ratio as seen by the unequal spacing between the three cases shown. Figure 56 (b) indicates that the axial number density drop increases with increasing aspect ratio.



(a) **(b)**

Figure 56. Angular profiles normalized with respect to the centerline value (a) and angular profiles normalized with respect to inlet conditions (b) for a range of aspect ratios.

4.1.5 Effect of Reynolds Number

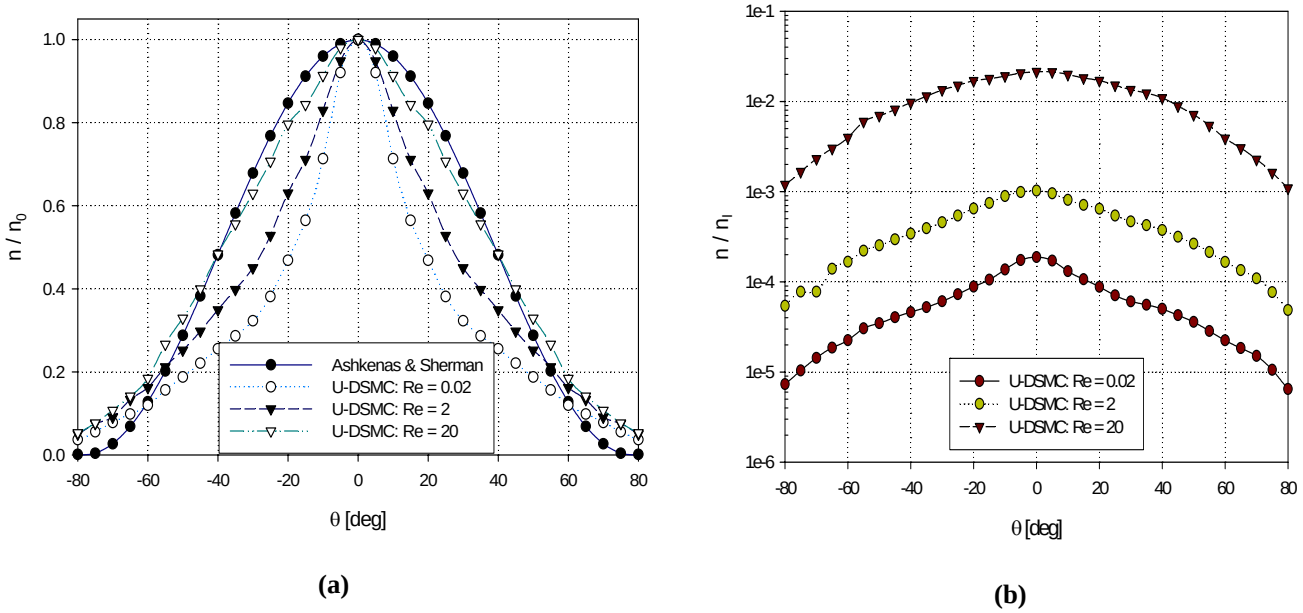
The variations in plume characteristics with inlet Reynolds number are investigated for a fixed Kn and L/D in cases 7-9 as given in Table 6. The Reynolds number is varied by increasing the axial velocity at the inlet.

Table 6. Selected parameters used for investigating the effect of Reynolds number on plume characteristics.

Case	$n_I\,(\mathrm{m}^{-3})$	$V_I\,(\mathrm{m/s})$	$D\,(\mu\mathrm{m})$	L/D	Kn	Re
7	1.29×10^{23}	5	10	10	1	0.02
8	1.29×10^{23}	500	10	10	1	2
9	1.29×10^{23}	5000	10	10	1	20

The resulting angular profiles taken at $x/D=10$ are displayed in Figure 57 (a). The base plume shape can be attributed to the combination of $L/D=10$ and $Kn=1$ as gathered from the previous sections. However from Figure 57 (b) the effect of the Reynolds number on the plume is rather clear. Increasing the Reynolds number widens the base plume shape in a fashion similar to decreasing the aspect ratio. Figure 57 (b) illustrates the effect that the Reynolds number has on the effective decrease in number density along the axis of the flow. As the Reynolds number increases the effective drop in the number density decreases. This result fits well with the interpretation of the Reynolds number as the ratio of inertial effects over viscous effects. Furthermore the similarity between the effect of increasing the Reynolds number or decreasing the aspect ratio also lends support to a viscous argument. As the aspect ratio decreases the relative number of gas-wall interactions decreases, as such the momentum transfer to the wall decreases and with it the effect of viscosity. Further investigation is needed

to rigorously define the underlying relation between the effects of aspect ratio and Reynolds number on the plume structure.



(a) **(b)**

Figure 57. Angular profiles normalized with respect to the centerline value (a) and angular profiles normalized with respect to inlet conditions (b) for a range of Reynolds numbers.

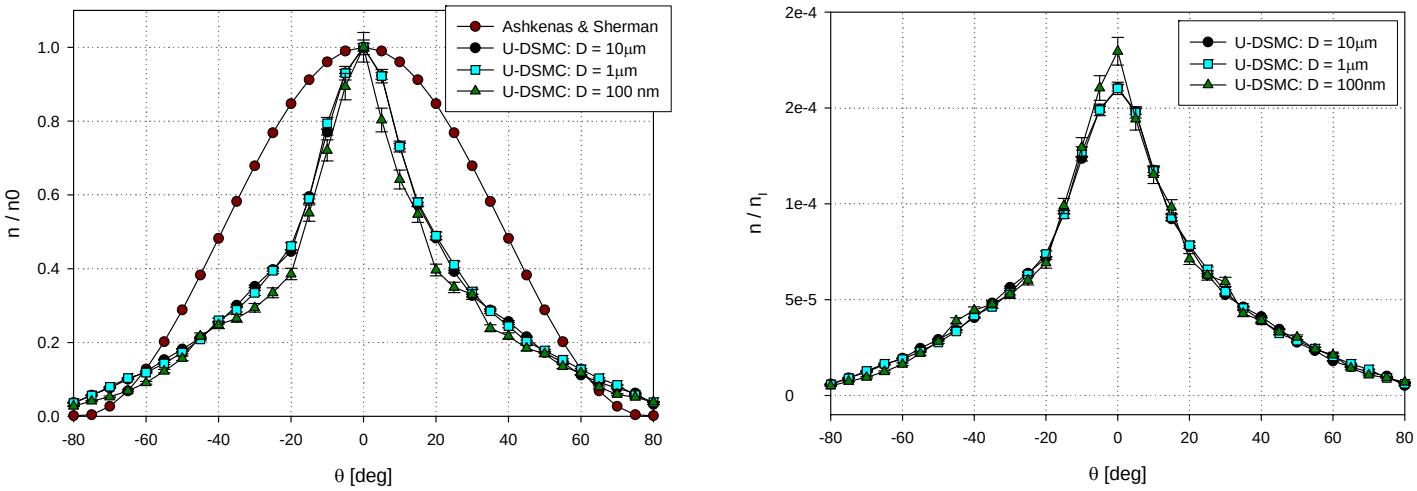
4.1.6 Effect of Scale

The variation in plume characteristics with the diameter of the microtube is investigated in order to determine if the base U-DSMC procedures capture scale induced phenomena. The simulation parameters are given in Table 7. The microtubes simulated all have an aspect ratio of 10. The inlet Knudsen number is kept constant for all three cases and the Reynolds number is fixed at 0.5. The scale of the geometry is varied over three decades.

Table 7. Selected parameters used for investigating the effect of geometric scale on plume characteristics.

Case	$n_I\,(\mathrm{m}^{-3})$	$V_I\,(\mathrm{m/s})$	$D\ (\mu\mathrm{m})$	L/D	Kn	Re
10	1.29×10^{23}	135	10	10	1	0.5
11	1.29×10^{24}	135	1	10	1	0.5
12	1.29×10^{25}	135	0.1	10	1	0.5

The resulting angular profiles taken at $x/D = 10$ are displayed in Figure 58 (a). The case of $D = 10\,\mu\mathrm{m}$ can be viewed as the basis of comparison since similar parameters were used in the two previous sections. The results shown in Figure 58 (a) indicate that the base DSMC procedures of Bird (1994) do not capture scale induced phenomena for the scales investigated. Since all three cases overlap within the numerical scatter, when accounting for the breakdown in symmetry of the plume for the 100 nm case, it can be inferred that U-DSMC, as currently implemented, is a scalable method. As such, microscale tube expansion simulations can be carried out on macroscaled geometries as long as the Knudsen number and Reynolds number are matched properly. Figure 58 (b) further illustrates the negligible effect of scale. Once again, the results of the three scales are nearly indistinguishable. The minor discrepancy between the peaks can be attributed to numerical error induced from the small particle weight required to insure a maximum number of simulated particles in the domain at the smallest scale.



(a) **(b)**

Figure 58. Angular profiles normalized with respect to centerline value (a) and angular profiles normalized with respect to inlet conditions (b) for a range of microtube scales at x/D = 10.

The results seen above indicate the inherent limitations of the algorithms that define the current implementation of the U-DSMC method. It is seen that scale-based physical phenomena will not alter the U-DSMC results. Therefore, real effects, such as surface roughness or chemical potentials at the walls, which may alter experimental results as scales decrease, will not be captured using standard U-DSMC implementation. The only scaling effect anticipated to occur in U-DSMC is an increase in statistical scatter as the limitations of general DSMC methodology, such as the requirement of molecular chaos, begin to break down as scales decrease and thus the number of real particles in the domain decrease. Further work is required over a wider range of geometries, with the Knudsen number and Reynolds number fixed, to determine the true nature of scaling U-DSMC.

4.1.7 Conclusions

The expansion of argon from microtubes into hard vacuum has been extensively investigated using the U-DSMC code. Simulation results for cases with an aspect ratio of 1.5 have been shown to compare well with commonly accepted theoretical formulations of free jet expansion. The discrepancies between the theoretical formulation and the U-DSMC results have been found to increase with increasing Knudsen numbers and aspect ratios, and with decreasing Reynolds numbers. These trends correspond to the breakdown of the assumptions used in the derivation of the three theoretical models.

U-DSMC investigations of the effect of Knudsen number, aspect ratio, Reynolds number and microtube scale on plume structure have been made. The plume profile has been found to narrow with increasing Knudsen numbers, as well as increasing the aspect ratio. However, the plume shape has also been found to narrow with decreasing Reynolds number.

The relative number density drop along the flow axis has been found to decrease with increasing Knudsen number and increasing Reynolds number. Conversely, a decrease in the relative number density drop has been observed for decreasing aspect ratio. From these investigations a similarity between the effects of aspect ratio and Reynolds number is seen. Furthermore, results obtained to date imply that the current U-DSMC implementation is a scalable method. Results of this study indicate that modeling gas expansion from tubes using U-DSMC is scalable with Reynolds number and Knudsen number.

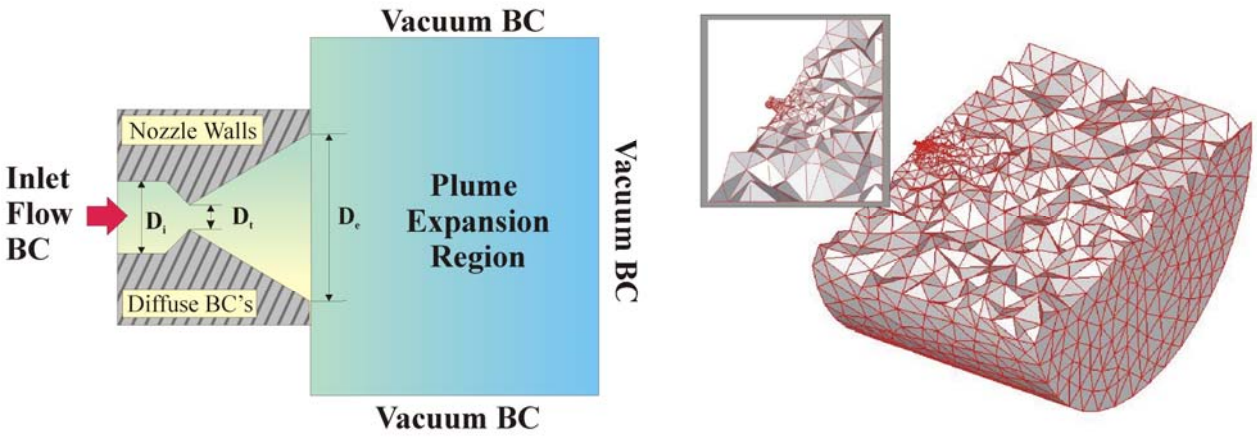
4.2 Expansion from Micro and Nano Nozzles

The expansion of helium from micronozzles into vacuum has been extensively investigated using the U-DSMC code. Investigations into the effect of Reynolds number, Knudsen number, speed ratio and scale have been carried out. The following work is an extraction of the findings presented in Chamberlin and Gatsonis (2006 (c)).

4.2.1 Geometry and Boundary Conditions

For this preliminary investigation, the chosen nozzle geometry corresponds to that shown in Figure 59 (a). The geometry is fixed for all cases investigated. Each nozzle features a conical half-angle of 20 degrees and an area ratio of 4. The simulation domains used in this study all feature an internal nozzle region, shown on the left side of Figure 59 (a) and the near field plume expansion region, as seen in the right side of Figure 59 (a).

The particle injection along the inlet of the nozzle is modeled as a molecular flux of an equilibrium gas. As such, the pressure, temperature and drift velocity of the equilibrium distribution at the boundary are chosen at initialization and enforced throughout the simulation. To aid in the development of a corresponding equilibrium distribution in the region of the inlet surface at steady state, any particles that move upstream along the inlet surface are removed from the flow. The inlet flow constraints allow for simple manipulation of the inlet Knudsen number and Reynolds number as is required for this investigation.



(a) **(b)**

Figure 59. Geometry and boundary conditions used in the nozzle simulations (a). An example of the unstructured grids used in the nozzle simulations, illustrating the localized grid spacing (b).

The walls of the nozzle are modeled as fully diffuse and the temperature of the wall is chosen to equal that of the inlet distribution. To keep the results as general as possible a hard vacuum was chosen to model the boundaries of the plume region. Although a backpressure boundary condition would be a better match for experimental investigation, the minimum achievable backpressure for each case investigated would vary with each facility. However, future investigations are planned which will study the effect of backpressure on the plume properties.

An example of the unstructured grids used in the nozzle simulations is shown in Figure 59 (b). Although the grid spacing varies for each case, a few relations remain constant amongst all cases. The first relation that is guaranteed for all grids is that the local cell spacing is always smaller than the local mean free path. The second cell spacing criteria common to all grids is sizing bounded by geometric resolution. In certain cases investigated in this work the local mean free path is much larger than that the nozzle throat diameter. When this relation holds the cell

spacing is set to the maximum size that still leads to a sufficient capturing of the nozzle curvature.

It should be noted that for small Reynolds number flows U-DSMC results exhibit significant numerical scatter. In order to reduce the statistical error in the results shown here, extensive time averaging of the data has been carried out for each case studied. Additionally, the number of simulation particles in each computational collision cell is maintained at well over twenty, while macroscopic parameters are calculated over volumes featuring at least 100 simulation particles.

4.2.2 Characterization of Fundamental Parameters

The focus of this study is to identify the key parameters governing the expansion of gas from micronozzles and nanonozzles. A secondary goal of this work is to characterize the effects of each key parameter on the resulting plume structure. The parameters under consideration for this study are the inlet Knudsen number (Kn), Reynolds number (Re) and speed ratio (S). The definitions used within this work were derived for the VHS model and take the form (Bird, 1994):

$$Kn = \frac{\lambda}{D} = \frac{1}{\sqrt{2}\pi {d_{ref}}^2 n \left(T_{ref}/T\right)^{\omega-1/2}} \frac{1}{D} \tag{3.11}$$

$$Re = \frac{2(5-2\omega)(7-2\omega)}{15\pi} \frac{S}{Kn} \tag{3.12}$$

$$S = \frac{V}{\sqrt{2kT/m}} \tag{4.2}$$

For the purpose of identifying and characterizing the effects of the key parameters a parametric study has been carried out over a range of parameter values. The resulting plume profiles are compared for each parameter set.

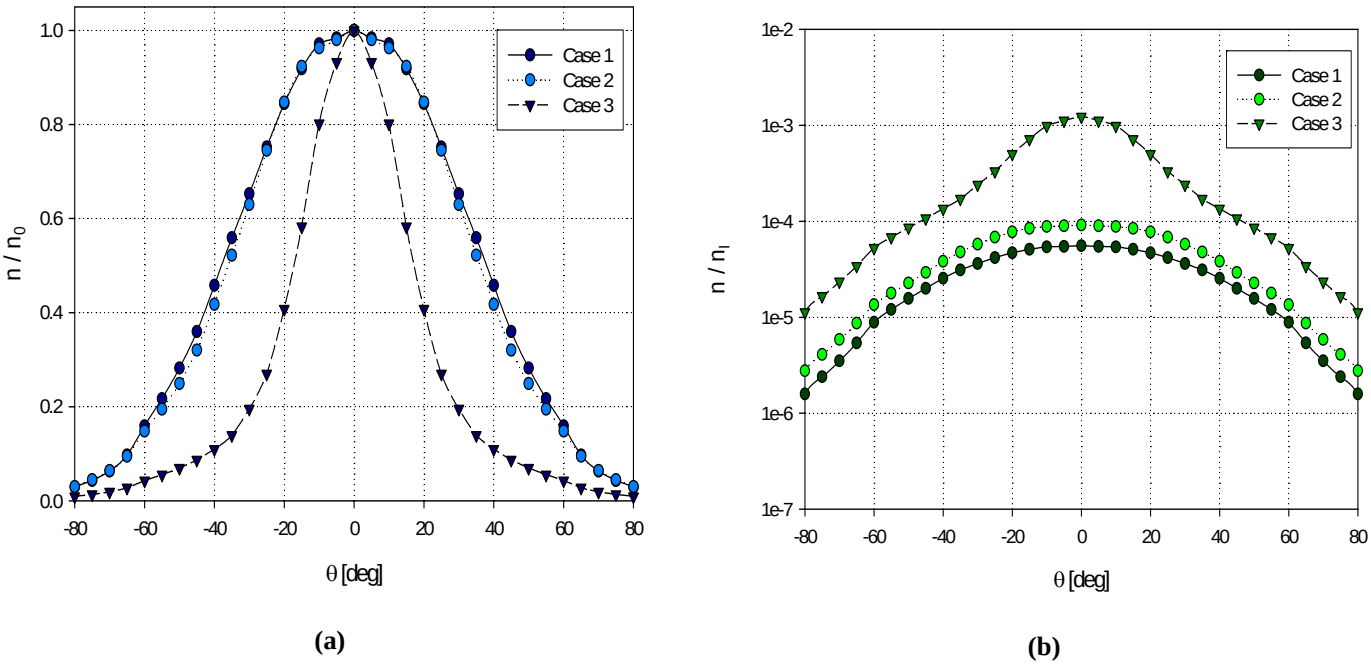
The first parameter set under investigation is chosen such that each case shares a common inlet Reynolds number of 0.5. When using the VHS definitions, varying the Knudsen number while fixing the Reynolds number requires that the speed ratio also vary. Unfortunately the relation between the commonly used governing parameters undermines the isolation of each potential key parameter. However, by carrying out several parameter set comparisons with each potential key parameter fixed one can gain insight into the significance of the fixed parameter.

Table 8. Simulation parameters used in fixed Reynolds number investigations.

	$D_t\ (\mu\mathrm{m})$	$n_1\ (\mathrm{m}^{-3})$	$V\ (\mathrm{m/s})$	$T\ (\mathrm{K})$	*Re*	*Kn*	*S*
Case 1	250	1.65e23	35	300	0.5	0.1	0.03
Case 2	250	1.65e22	350	300	0.5	1	0.3
Case 3	250	1.65e21	3500	300	0.5	10	3

The parameter values used for the first set of simulations are given in Table 8. For the purpose of the current study, the number density is sampled as a function of the radial distance from the nozzle exit plane as well as the angle from the nozzle axis, $n(R,\theta)$. The resulting plume profiles are seen in Figure 60. Figure 60 (a) is a comparison plot of the number density, normalized by the centerline value $n_0 = n(R,0)$, at a radial distance of 20 exit diameters (40 throat diameters). From the comparison plot of Figure 60 (a) a clear narrowing of the plume can be seen as both the Knudsen number and speed ratio are increased. Figure 60 (b) is a comparison plot of the number density, also at a radial distance of 20 exit diameters, scaled using

the inlet value $n_I = n\left(x_{Inlet}, 0\right)$. From Figure 60 (b) it can be seen that the relative drop in the number density along the flow path decreases with increasing Knudsen number and speed ratio.



(a) **(b)**

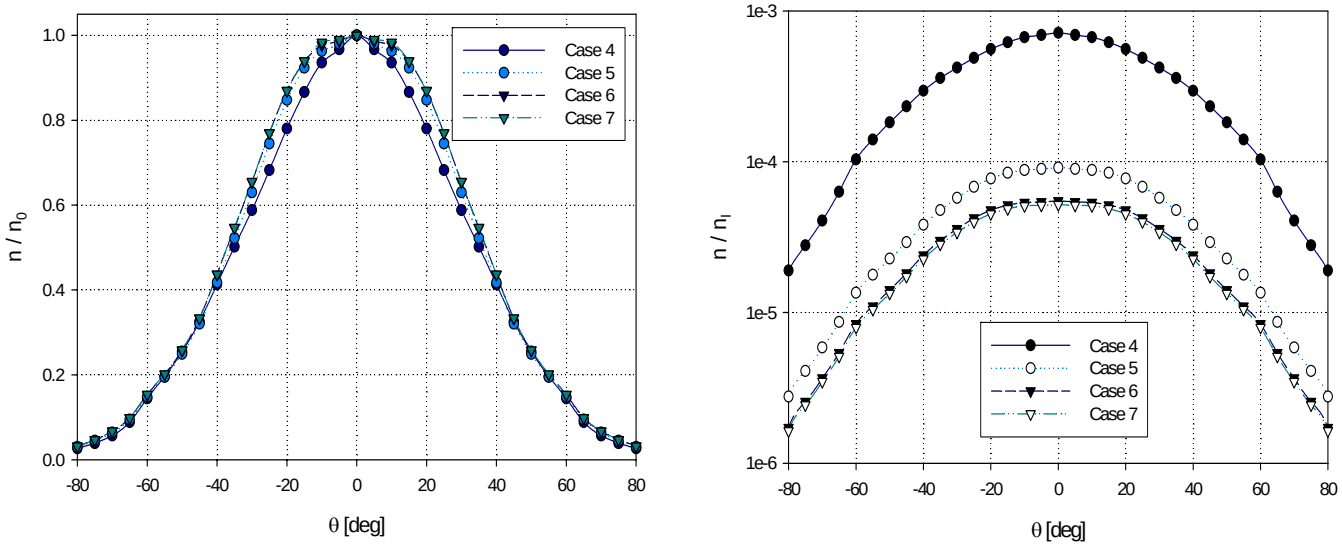
Figure 60. Plume profiles for parameter sets with a common inlet Reynolds number of 0.5, sampled at a radial distance of 40 throat diameters from the nozzle exit plane.

The second parameter set investigated is chosen such that each case shares a common inlet Knudsen number. The chosen value for the common Knudsen number is 1, which is well within the transitional regime. The parameter values used for the second set of simulations are given in Table 9.

Table 9. Parameter values used for fixed Knudsen number (Kn) simulations.

	D_t (μm)	n_I (m^{-3})	V (m/s)	T (K)	Re	Kn	S
Case 4	250	1.65e22	3500	300	5	1	3
Case 5	250	1.65e22	350	300	0.5	1	0.3
Case 6	250	1.65e22	35	300	0.05	1	0.03
Case 7	250	1.65e22	3.5	300	0.005	1	0.003

The resulting plume profiles are seen in Figure 61. In Figure 61 (a) a slight narrowing of the plume can be seen as both the Reynolds number and speed ratio are increased. From Figure 61 (b) it can be seen that the relative drop in the number density along the flow path clearly decreases with increasing Reynolds number and speed ratio.



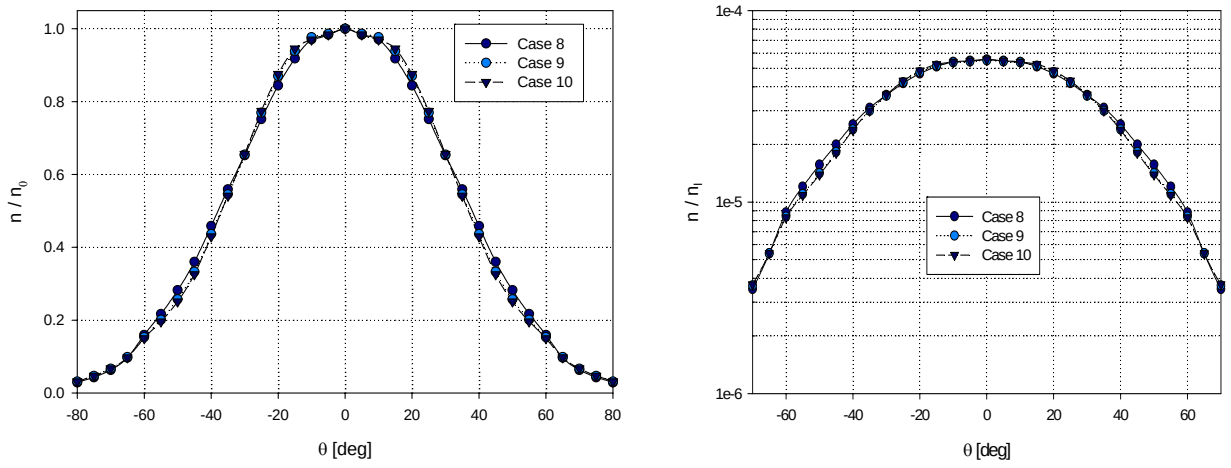
(a) **(b)**

Figure 61. Plume profiles for parameter sets with a common inlet Knudsen number of 1, sampled at a radial distance of 40 throat diameters from the nozzle exit plane.

Table 10. Parameter values used for fixed speed ratio (S) simulations.

	$D_t\ (\mu\text{m})$	$n_1\ (\text{m}^{-3})$	$V\ (\text{m/s})$	$T\ (\text{K})$	*Re*	*Kn*	*S*
Case 8	250	1.65e23	35	300	0.5	0.1	0.03
Case 9	250	1.65e22	35	300	0.05	1	0.03
Case 10	250	1.65e21	35	300	0.005	10	0.03

The third parameter set under investigation is chosen such that each case shares a common inlet speed ratio of 0.03. The parameter values used for the third set of simulations are given in Table 10. The resulting plume profiles are seen in Figure 62. In Figure 62 (a) and Figure 62 (b), the comparison plots seem to indicate that the plume profile is not significantly altered by increasing the Knudsen number or decreasing the Reynolds number provided that the speed ratio is fixed.



(a) **(b)**

Figure 62. Plume profiles for parameter sets with a common inlet speed ratio of 0.03, sampled at a radial distance of 40 throat diameters from the nozzle exit plane.

The cause for the fixed plume profile is uncertain but is most likely attributed to one of two possibilities. Either the speed ratio is the dominate parameter effecting the plume structure or the effects incurred as the Knudsen number increases and the Reynolds number decreases cancel each other out. Further study is required to ascertain the cause of the static plume profile seen in this parameter set.

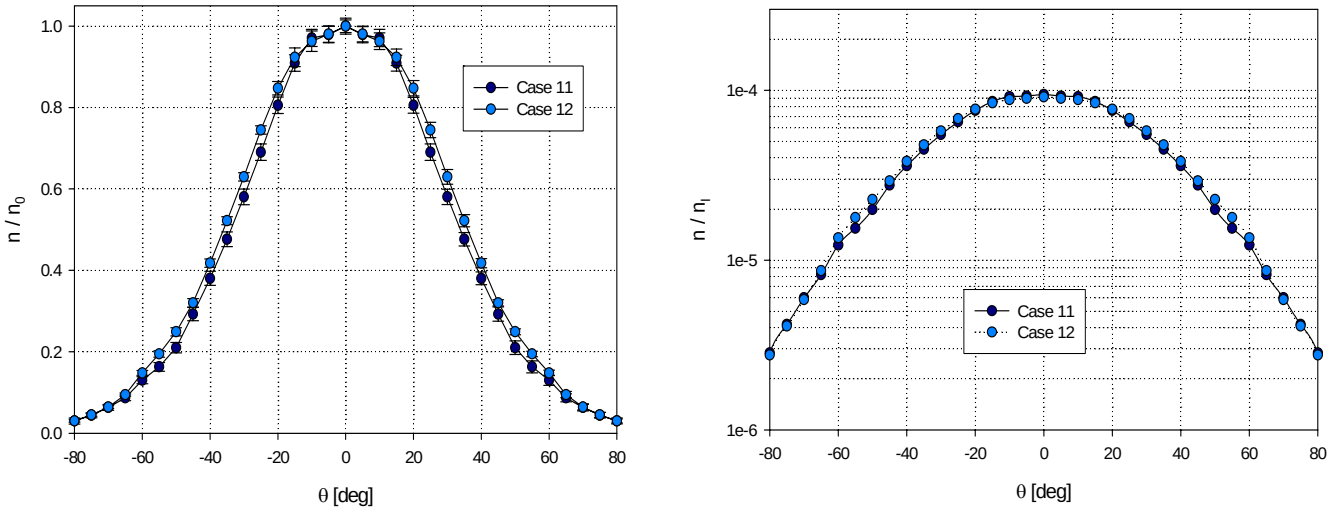
The final data set investigated in this study isolates the effect of nozzle scale on the resulting plume profile predicted by the current U-DSMC implementation. Two nozzle scales are simulated. The first has a throat diameter of 250 μm while the second has a throat diameter of only 250 nm. For both cases the nozzle geometry is identical as are the values of the inlet Knudsen number, Reynolds number and speed ratio. The simulation parameters are given in Table 11.

Table 11. Parameter values used for scale simulations.

	D_t	$n_1\ (\mathrm{m}^{-3})$	$V\ (\mathrm{m/s})$	$T\ (\mathrm{K})$	*Re*	*Kn*	*S*
Case 11	250 μm	1.65e22	350	300	0.5	1	0.3
Case 12	250 nm	1.65e25	350	300	0.5	1	0.3

The resulting plume profiles are seen in Figure 63. In Figure 63 (a) and Figure 63 (b) the comparison plots indicate that the plume profile is not significantly altered by decreasing the nozzle scale provided that the Knudsen number, Reynolds number and speed ratio are fixed. This result is expected considering that the current U-DSMC implementation is scale insensitive and does not capture the microscopic effects, such as the molecular potential interactions between the gas and surface molecules, that would become more prevalent at the reduced scales

seen here. Figure 63 (a) illustrates one effect of reducing the scale of the U-DSMC simulation domain which is a significant increase in statistical fluctuations in the sampling of macroscopic variables. For the simulation sets shown above the results were obtained using time-averaged data which were averaged over twenty data sets (of 100 iteration intervals) in order to obtain number density data where the statistical noise was reduced to a point where 95% confidence interval error bars were smaller than the plotted point markers. For the nanoscaled nozzle, data averaging was carried out over 100 data sets yet the the statistical scatter amongst the plotted points is comparatively large. Future work is planned that will lend further insight into the scale-induced increase in statistical scatter in U-DSMC simulation at the nanoscale.



(a) **(b)**

Figure 63. Plume profiles for scale parameter sets with a common inlet speed ratio, Knudsen number and Reynolds number.

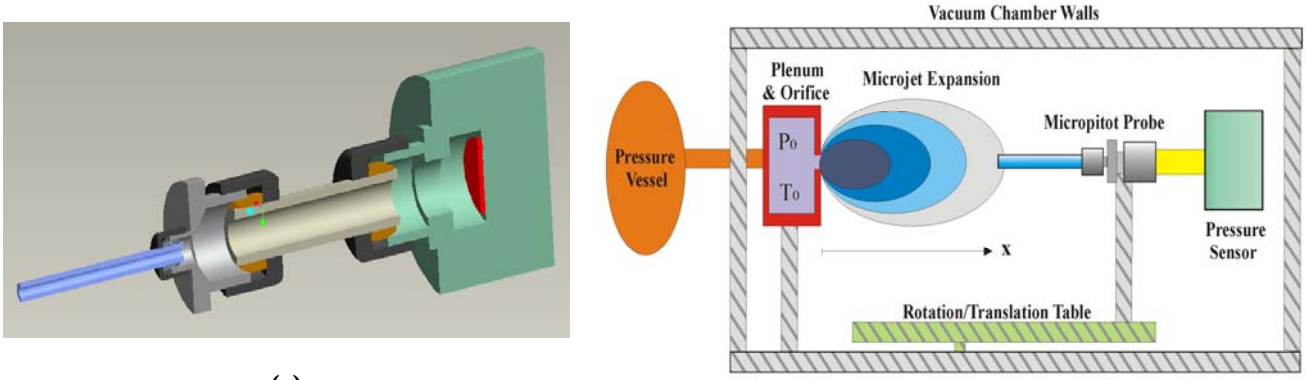
4.2.3 Conclusions

A parametric study has been carried out over a range of Knudsen number, Reynolds number and speed ratio values. The resulting plume profiles were compared for each set of parameters. From the comparisons it has been found that the plume profile narrows as both the Knudsen number and speed ratio are increased for a fixed Reynolds number. Furthermore, the relative drop in the number density along the flow path was found to decrease with increasing Knudsen number and speed ratio for a fixed Reynolds number. A slight narrowing of the plume was seen as both the Reynolds number and speed ratio were increased for a fixed Knudsen number. While the relative drop in the number density along the flow path clearly decreases with increasing Reynolds number and speed ratio for a fixed Knudsen number. It was further observed that the plume profile is not significantly altered when increasing the Knudsen number while decreasing the Reynolds number with the speed ratio fixed. With the exception of an increase in statistical fluctuations, the current U-DSMC implementation has been found to be scale insensitive.

4.3 Micropitot Probe Pressure Measurement Predictions

A two-stage numerical modeling approach is used to guide the design of a micropitot probe, shown in Figure 64 (a). The micropitot probe consists of a stainless steel tube with an outer diameter of $OD = 150\,\mu\mathrm{m}$ and an inner diameter of $D_p = 63.5\mu$m coupled to a pressure sensor. The U-DSMC simulations model the experimental test scenario shown in Figure 64 (b) in which the micropitot probe is used to measure the pressure field of a microjet. In the first modeling stage, the expansion of nitrogen from the microjet into vacuum is modeled using the U-DSMC code. Local free stream conditions are extracted from the resulting microjet plume

and are used in the second stage of the investigation as boundary conditions for a series of micropitot probe simulations. From the results of stage two, predictions of the pressure within the probe are obtained for a range of axial distances from the exit plane of the microjet. The following work is an extraction of the findings presented in Chamberlin and Gatsonis (2006 (a)).

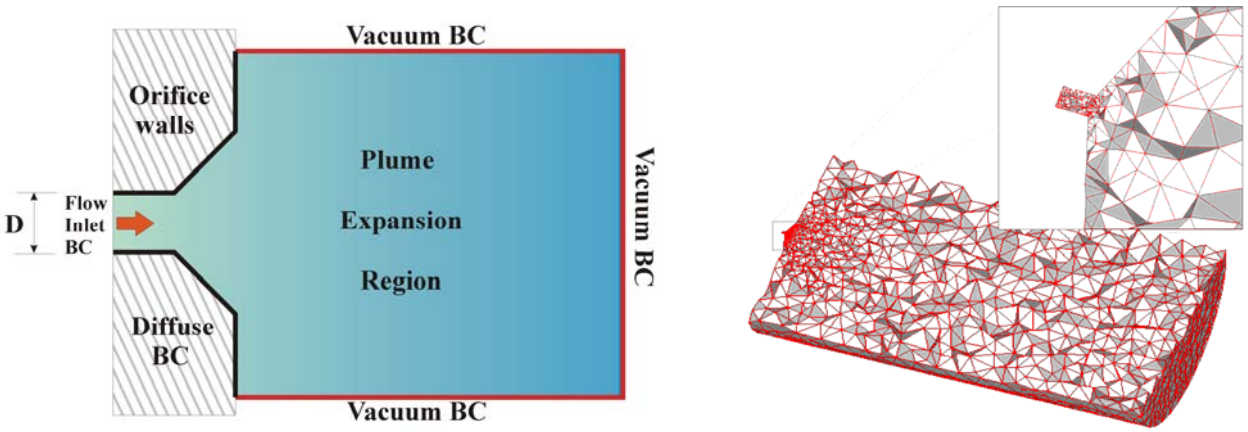


(a) **(b)**

Figure 64. Micropitot probe design (a) and a schematic of the experimental test scenario (b).

4.3.1 Simulation of Microjet Expansion

Within the first stage of the investigation the expansion from a microjet corresponding to anticipated geometry and inlet conditions of the experimental test case are carried out using U-DSMC. The U-DSMC simulations for the microjet expansion are carried out on a domain that corresponds to the geometry shown in Figure 65 (a). The simulation domain features the internal flow region, shown on the left side of the figure, and the external plume expansion region, shown on the right side of Figure 65 (a). The grid spacing varies throughout the domain, as shown in Figure 65 (b), with the nominal cell sizing corresponding to a fraction of the local mean free path.



(a) (b)

Figure 65. Microjet simulation domain and boundary conditions (a) and unstructured mesh of the microjet simulation domain, featuring localized cell spacing (b).

For the microjet expansion simulation the particle injection along the inlet of the orifice is modeled as a molecular flux of an equilibrium gas. As such, the pressure, temperature and drift velocity of the equilibrium distribution at the boundary are chosen at initialization and enforced throughout the simulation. To aid in the development of a corresponding equilibrium distribution in the region of the inlet surface at steady state, any particles that move upstream along the inlet surface are removed from the flow. The inlet flow constraints match anticipated plenum region conditions. In keeping with the anticipated test conditions, the plenum temperature is set to the ambient laboratory value of $300\,\mathrm{K}$, the inlet number density used was $1.29\times10^{23}\,\mathrm{m}^{-3}$, and the drift velocity applied was $15\,\mathrm{m/s}$. The walls of the orifice are modeled as fully diffuse and the temperature of the wall is also chosen to equal that of the ambient laboratory. This yields the most general relation between the wall temperature and the stagnation chamber temperature and therefore should lend simplicity to experimental validation of the U-DSMC results.

The final boundary conditions required for the simulations are the downstream plume region boundaries. To keep the results as general as possible a hard vacuum was chosen to model the boundaries of the plume region. Although a backpressure boundary condition would be a better match for experimental investigation, the minimum achievable backpressure for the intended test facility during microjet operation is unknown at this time. However, future investigations are planned which will include the effect of backpressure on the plume properties.

Microjet Expansion Results

The investigation begins with a comparison between simulation results of microjet expansion profiles and three theoretical models developed for supersonic free jets, given in detail in Section 3.4. The purpose of the comparisons is twofold. The primary motivation is that of plume characterization, that is, to define the shape of the plume with respect to the theoretical standards. The secondary motivation lies in the need to determine which of the three standard models most accurately captures the U-DSMC results for the current set of parameters, thereby setting the precedent for future investigations.

[Ashkenas & Sherman] $$\frac{\rho(R,\theta)}{\rho(R,0)} = cos^2\left(\frac{\pi\theta}{2\phi}\right) \tag{3.13}$$

where $\phi = 1.662$ for nitrogen.

[Boynton] $$\frac{\rho(R,\theta)}{\rho(R,0)} = \cos^{\frac{2}{\gamma-1}}\left(\frac{\pi\theta}{2\theta_{\lim}}\right) \tag{3.14}$$

[Albini & Hubbard] $$\frac{\rho(R,\theta)}{\rho(R,0)} = \cos^{\frac{1}{\gamma-1}}\left(\frac{\pi\theta}{2\theta_{\lim}}\right) \tag{3.15}$$

Since Equations (3.13)-(3.15) were formulated from continuum solutions (typically having Knudsen numbers less than 0.01) where the Reynolds number was relatively large, it is of value to note the Knudsen number and Reynolds number of the microjet flow modeled here. The inlet injection is sampled from a distribution with $T_I = 300\,\mathrm{K}$, $n_I = 1.29 \times 10^{23}\,\mathrm{m}^{-3}$, and an axial drift of $V_I = 15\,\mathrm{m/s}$. The orifice diameter is taken as 100μm. The resulting Knudsen number is then calculated to be $Kn = 0.1$ and as such the flow in near-field plume is expected to show characteristics fitting that of the near-continuum regime. The inlet Reynolds number is found to be $\mathrm{Re}_I = 0.5$. Although this value is small in absolute magnitude, it is relatively large in terms of microscale gas flows.

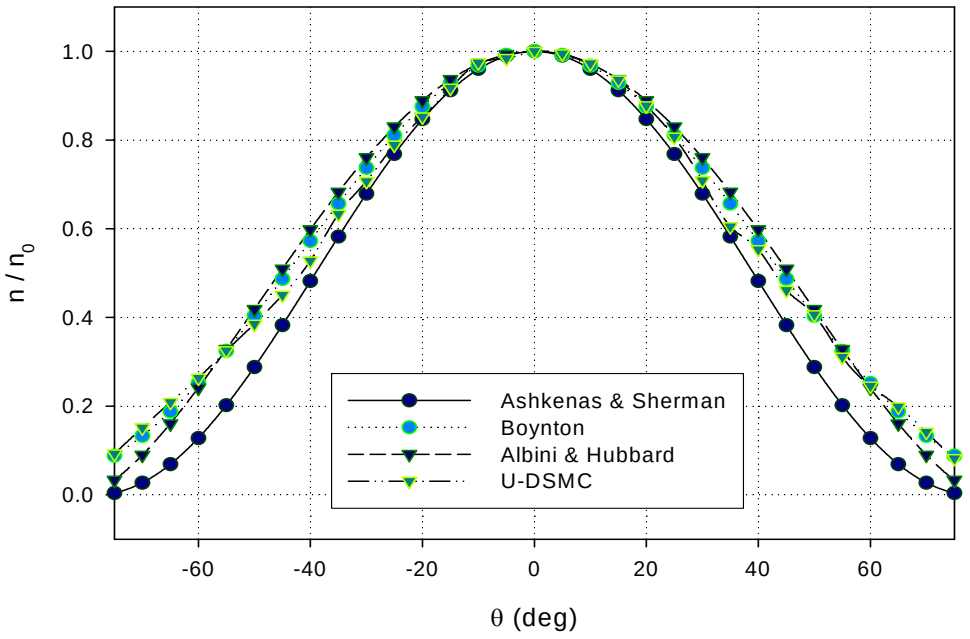


Figure 66. Comparison of angular number density at an axial distance of 5mm from the orifice exit plane.

Figure 66 shows the normalized density at a distance of $R = 5\,\mathrm{mm}$ downstream from the exit plane (which corresponds to fifty orifice diameters). The number density values at each angle were normalized by the centerline number density n_0. Since the flow is comprised of a single species, nitrogen, the normalized density is equal to the normalized number density. Figure 66 illustrates the excellent agreement that is found between the U-DSMC results and the theoretical predictions. For polar angles of $-45^o \le \theta \le 45^o$ the point-to-point RMS error between the U-DSMC results and theory varies from 1% to 15% for Eq. (3.13), from 1% to 8% for Eq. (3.14) and from 1% to 13% for Eq. (3.15). From these comparisons it is seen that the formulation of Boynton [Eq. (3.14)] gives the closest approximation to the U-DSMC results. Furthermore, the comparisons indicate that the plume shape fits the standards rather closely, thus setting a basis configuration for future studies.

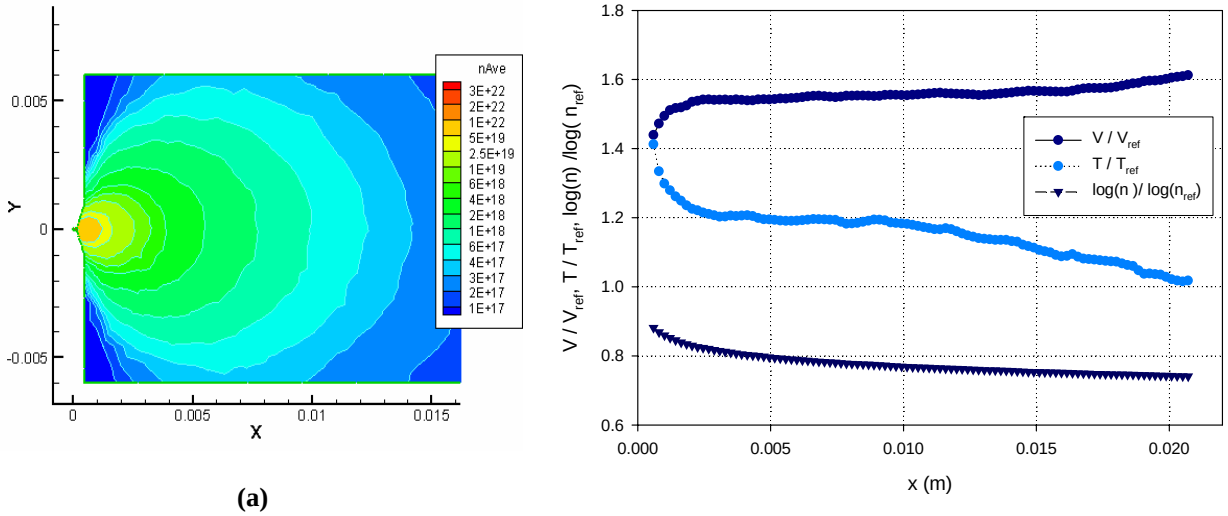


Figure 67. Number density contours of U-DSMC results (a) and centerline values of axial velocity (V), temperature (T), and number density (n). The normalization constants have values of V_{ref} = 353 m/s, T_{ref} = 35K, and n_{ref} = $1.29\times10^{23}\mathrm{m}^{-3}$ (b).

A sense of the overall structure of the plume can be ascertained from the number density contours obtained from the U-DSMC simulations as shown in Figure 67 (a). The number density contours indicate that the expansion occurs uniformly from the orifice exit plane as indicated by the circular shaping of the plume. Figure 67 (b) shows the centerline values of the plume temperature, number density, and velocity. Local free stream conditions are extracted from the U-DSMC centerline data and are used in the second stage of the investigation as boundary conditions for a series of micropitot probe simulations. The resulting free stream parameters are tabulated in Table 12.

Table 12. Centerline values of axial velocity (V), temperature (T), and number density (n) at various axial distances from microjet exit plane.

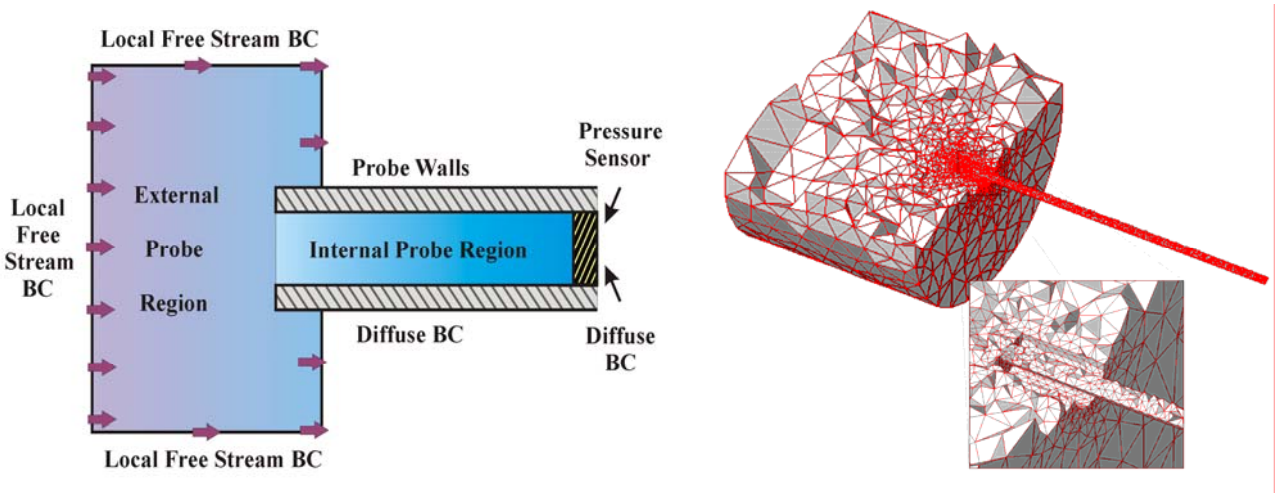
x (m)	V (m/s)	T (K)	n (m^{-3})
0.0008	520	46.7	1.19e20
0.0015	536	44.0	3.40e19
0.0025	544	42.5	1.06e19
0.005	544	41.8	2.49e18
0.0075	547	41.7	1.02e18
0.0125	550	40.2	3.91e17
0.0175	556	37.6	1.89e17
0.02	566	35.7	1.42e17

4.3.2 Micropitot Probe Simulations

Following the attainment of the local free stream conditions from the microjet expansion of the previous section, predictions of micropitot probe performance are obtained from segmented local simulations. The current design of the micropitot probe features a simplistic union of a pressure sensor (model yet to be determined) and a stainless steel tube. The tubing is available with inner diameters ranging down to roughly 60μm with a minimum outer diameter

of $150\mu m$. The length of the tube is variable and is currently chosen to be $2.5mm$. The details of the union required to affix the tube and the pressure sensor are also variable at this time (will be dependant upon sensor selection). As such the sensor will be assumed to be located at the end of the tube with negligible sensor volume. The actual sensor volume along with any required union volume will be addressed in future investigations as the design process progresses.

The U-DSMC simulations for the micropitot probe entrance and internal regions are carried out on a domain that corresponds to the geometry shown in Figure 68. The simulation domain features the external flow region, shown on the left side of the figure, and the internal probe region, shown on the right side of Figure 68 (a). The grid spacing varies throughout the domain, as shown in Figure 68 (b). As with the microjet simulation domain, the grid spacing for the probe simulations adhere to minimum spacing constraints required for physical collision modeling. The grid spacing is further constrained by the geometric resolution required to accurately capture the structure and dimensionality of the intended device.



(a) **(b)**

Figure 68. Micropitot probe simulation geometry and boundary conditions (a) and unstructured mesh of the micropitot probe geometry (b).

The domain size has been chosen such that the flow disturbance ahead of the probe entrance is contained within the external flow region, while simultaneously minimizing the distance at which the local free stream conditions are applied. This balance, along with the finite radius of the probe simulation domain, leads to the need for a minor numeric approximation for the inlet conditions. The approximation made for the probe model assumes that the radial divergence of the local free stream has negligible effect over the radius of the probe domain for the axial locations investigated. Further, it is assumed that the local free stream undergoes negligible deviations from the imposed free stream conditions across the span between the free stream boundary and the probe inlet. Figure 69 illustrates the overlap of the microjet simulation region and the micropitot probe domain.

For the micropitot probe simulations the particle injection along the free stream boundaries of the domain is modeled as a molecular flux of an equilibrium gas with the pressure, temperature and drift velocity set to that of the local free stream conditions obtained from the microjet expansion simulation. The walls of the micropitot probe are modeled as fully diffuse and the temperature of the wall is set to the stagnation temperature of the flow. The inner diameter of the probe is $D_p = 63.5\mu\text{m}$ while the outer diameter is $150\mu\text{m}$. The length from the entrance to the pressure sensor surface is $L_p = 2.5\text{mm}$. The pressure sensor boundary of the probe is modeled as fully diffuse, with a temperature equal to that of the flow stagnation temperature. The resulting pressure on the sensor boundary is monitored throughout the simulation. The sensor pressure is obtained directly from the momentum transfer that occurs during molecular impingement with the simulated sensor surface.

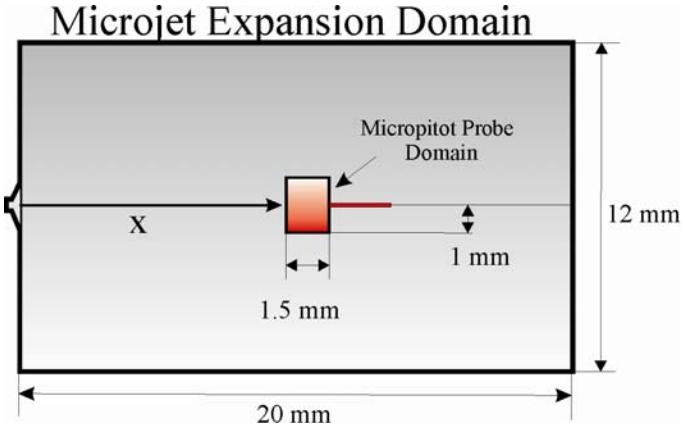


Figure 69. Schematic illustrating the overlap of the micropitot probe domain with that of the microjet expansion domain.

Micropitot Probe Results

The resulting pitot pressure predictions are shown in Figure 70 as a function of axial distance from the orifice exit plane. The U-DSMC results are plotted together with predictions from two theoretical models in order to establish a basis for future comparison as well as to determine the range of applicability of each theory.

The first relation used is developed by Edwards and Quan (1966) from a solution of the Boltzmann equation for a pitot tube immersed in hypersonic adiabatic rarefied flow. The pressure at the sensor end of the tube is given as:

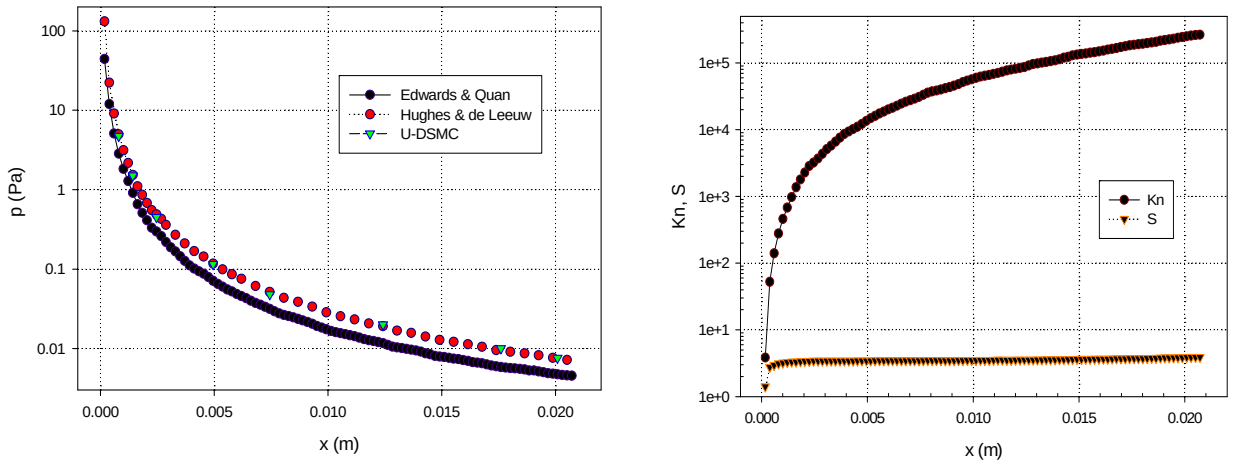
$$p = p_0 S^2 \left[(1.12S + 1.19)/\delta - \chi S^2 \left(0.75S + 2 - 4.2\delta + 2.9\delta^2 \right) \right] \tag{4.3}$$

where,

$$\delta = \left[\frac{\gamma + 1}{4(\gamma - 1)} \right]^{1/2} \tag{4.4}$$

$$\chi = \frac{8\sqrt{2}}{15\pi S \cdot Kn} \tag{4.5}$$

In equations (4.3)-(4.5), γ is the ratio of specific heats, $p_0 = n_0 k_B T_0$ is the freestream static pressure, S is the speed ratio of the free stream and Kn is the Knudsen number based on the free stream mean free path and the inner diameter of the probe.



(a) **(b)**

Figure 70. Micropitot probe pressure sensor predictions as a function of axial distance (x in Figure 69) from the microjet exit plane (a) and free stream values of *Kn* and *S* versus axial distance from the microjet exit plane (b).

Eq. (4.3) is derived for the near-free-molecule regime where the distribution function of the internal flow is taken as the sum of two separate distribution functions, one characteristic of the free stream the other characteristic of the molecules reflecting from the tube walls. It should be noted that Eq. (4.3) is the reduced relation that holds for the assumption of a long tube. This assumption is taken to hold for the present case since the aspect ratio of the current pitot probe design is larger than 40. It should also be noted that, as seen in Figure 70 (b), the local free stream conditions for each plume location of interest match the requirement of a hypersonic speed ratio, however the local Knudsen number is well beyond the free-molecular limit and thus

beyond the intended range of the theory. This discrepancy is most likely the cause of the slight bias between the theory of Edwards and Quan and the U-DSMC results, as seen in Figure 70 (a). Future investigations into the trends of pitot probe pressure as a function of Knudsen number, speed ratio and Reynolds number should clarify the cause of the bias.

The second relation used for comparison to the simulation results is a formulation developed by Hughes and de Leeuw (1965) for free-molecule impact pressure probes at an arbitrary angle of attack. The model assumes a drifting Maxwellian distribution for the free stream with speed ratio S, temperature T_1, and pressure $P_1 = n_1 k T_1$ at an angle of attack α with a tube having a diameter to length ratio of A. Perfectly diffuse reflection is assumed along the boundary. The flow conditions in the sensor chamber are designated by a temperature T_2 and the equilibrated pressure inside the volume, $P_2 = n_2 k T_2$, is presented in terms of the pressure ratio $R(S, \mathrm{A}, \alpha)$ by:

$$R(S, \mathrm{A}, \alpha = 0) = \frac{2}{W(\mathrm{A})} \int_0^{2\pi} K(\zeta, \mathrm{A}) F(\zeta, S, 0) d\zeta \tag{4.6}$$

where,

$$R(S, \mathrm{A}, \alpha) = (P_2 / P_1)(T_1 / T_2)^{1/2} \tag{4.7}$$

In Eq. (4.6) $K(\zeta, \mathrm{A})$ captures the response of the cylindrical tube to a molecular beam and $F(\zeta, S, 0)$ is a function that captures the drifting Maxwellian beam flux density. At equilibrium the flux of molecules that exit the chamber of the sensor is equal to the flux entering the tube.

The Hughes and de Leeuw model was implemented in a computer code in the work of Maynard (1996) and used for the comparisons shown in Figure 70. Once again it should be noted that the plume points of interest exhibit Knudsen numbers corresponding to the free molecular regime and speed ratios that indicate supersonic velocities, as seen in Figure 70 (b).

Since the flow field properties are consistent with the governing assumptions of the Hughes and de Leeuw theory excellent agreement is seen between the theoretical predictions and the U-DSMC results, as shown in Figure 70, for the simple geometry of the current pitot probe design.

4.3.3 Conclusions

From the resulting pressure predictions shown in Figure 70, several conclusions can be made. First, for the conditions of the present case, the theory of Hughes and de Leeuw can be used to determine the sensor pressure for the micropitot probe with acceptable accuracy. However, the geometric limits of the applicability of the theory have not been pushed by the current design and as such no conclusions can be made for the accuracy of the formulation for more complex geometries. The second conclusion which may be drawn is that the theory of Edwards and Quan is not acceptably accurate for the conditions of the current case. However, its range of applicability may have been overextended in the current U-DSMC study and as such future investigations will be needed to determine the applicability range in terms of the Knudsen number for both the Edwards and Quan theory and the Hughes and de Leeuw theory.

A further conclusion which can be drawn is that the required pressure range of a sensor implemented in the current experimental configuration is from 100 to 0.005 Pa (or 0.75 to 3.75e-5 torr). Furthermore, the U-DSMC results indicate that the effect of the blunt entrance geometry on the local probe-external flow field is negligible in the investigated Knudsen range. Additionally, it is seen from comparison with the Hughes and de Leeuw theory that the chosen length of the microtube does not markedly impede the transition of the external flow through the internal region in a manner not predicted by the theory. However, further investigations are

required, once sensor selection has been made, to determine whether the pressure sensor housing will have a significant effect on the microjet flow field and the resulting internal pressure.

4.4 Quantifying Scale-Induced Statistical Fluctuations in U-DSMC

The work presented in Sections 4.1 and 4.2 has uncovered scale-induced increases in the statistical fluctuations of U-DSMC results as the characteristic dimension approaches the nanoscale. In order to further quantify this effect, Poiseuille flow is simulated over a wide range of scales. For each simulation set the error between the theoretically predicted velocity profiles and those generated by U-DSMC is calculated. From this error a relation between statistical fluctuations in U-DSMC data and the characteristic scale is shown.

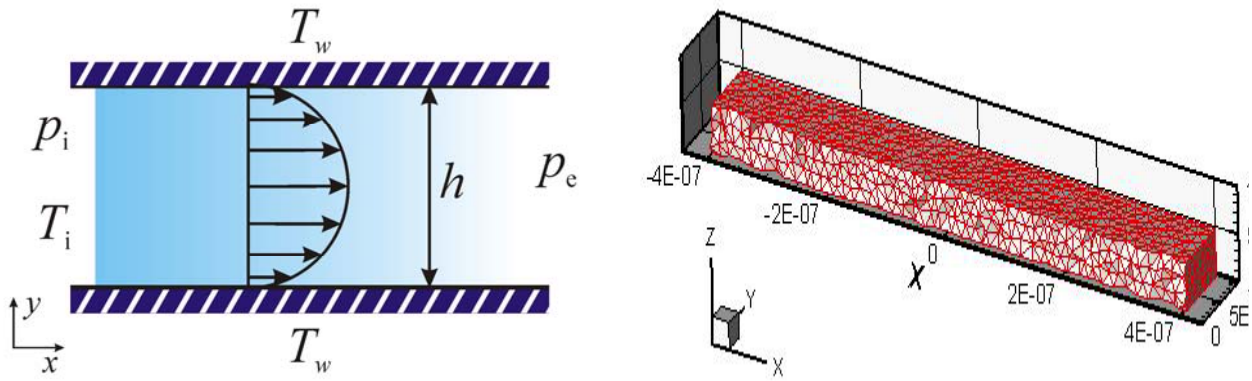
4.4.1 Scale-induced Statistical Fluctuations in Poiseuille Flow

Poiseuille flow is used as a test case in order to quantify the scale-induced statistical scatter that occurs as the characteristic domain length approaches the submicron range. To isolate the effect of scale on macroscopic sampling both surface roughness and detailed gas-surface potential interactions are neglected in this study. Furthermore, the inlet Knudsen number and pressure ratio will be fixed over all cases such that the scalability of U-DSMC, as shown in Sections 4.1 and 4.2, can be utilized in order to produce scale-based similarity within the cases.

Geometry and Boundary Conditions

Poiseuille flow is a pressure-driven flow between two infinite stationary parallel plates. The inlet surface is maintained at a fixed pressure p_i and temperature T_i while the outlet is maintained at a fixed pressure p_e with the pressure ratio along the flow path given as

$p_{ratio} = p_i / p_e$. The plate walls are fixed at a temperature T_w. The geometry corresponding to Poiseuille flow is illustrated in Figure 71 (a). Within this investigation the infinite parallel plates are modeled as a channel with fully diffuse upper and lower walls and the side walls are modeled as specularly reflecting thus effectively acting as symmetry planes. An example grid used in the simulations in shown in Figure 71 (b).



(a) **(b)**

Figure 71. Geometric configuration (a) and example grid (b) used in Poiseuille scaling studies.

Velocity Profile Comparison

As a means of determining the functionality between scale-induced statistical fluctuations a number of simulations have been carried out over a wide range of channel heights. The relevant parameters for each case are listed in Table 13. For each case studied, the grid structure has been scaled in such a manner as to constrain the number of cells in the domain to within 1% of the group average. In doing so, the geometric resolution of the domain is nearly equal for all cases. Therefore the macroscopic sampling volumes scale uniformly with each case. As such, isolation of the effect of scale on the number of molecules in each sample volume is achieved.

Table 13. Selected parameters used in Poiseuille scale simulations.

	h (m)	p_i (Pa)	n_I (m^{-3})	T_i (K)	p_{ratio}	Kn	T_w (K)
Case 1	1e-3	536	1.29e23	300	10	0.01	300
Case 2	1e-6	5.36e5	1.29e26	300	10	0.01	300
Case 3	100e-9	5.36e6	1.29e27	300	10	0.01	300
Case 4	25e-9	1.07e7	2.59e27	300	10	0.01	300
Case 5	50e-9	2.14e7	5.18e27	300	10	0.01	300
Case 6	10e-9	5.36e7	1.29e28	300	10	0.01	300

In order to establish the scale-induced error resulting from increased statistical fluctuation two error approximation schemes are used. The first method is an application of the definition of the root mean-squared error. As such, the error associated with this method is an internal measure of the statistical variation of the U-DSMC data from its own local time-average. In general form, the local-mean based error relation can be written as:

$$RMS = \sqrt{MSE} \tag{4.8}$$

with,

$$MSE = var\left[\left\langle x\right\rangle\right] = var\left[\frac{1}{N}\sum_{i=1}^{N} x_i\right] = \frac{1}{N}\sum_{i=1}^{N}\left(x_i - \left\langle x\right\rangle\right)^2 \tag{4.9}$$

Application of the relation for *RMS* to the U-DSMC results occurs in a post-processing manner by means of the data manipulation tools of Tecplot. Within the current study a fixed number of data sets are used for each case. The number of data sets used for time-averaging within this study has been set to 10. Therefore the local time-averaged data take the form:

$$\langle x \rangle = \frac{1}{10} \sum_{i=1}^{10} x \tag{4.10}$$

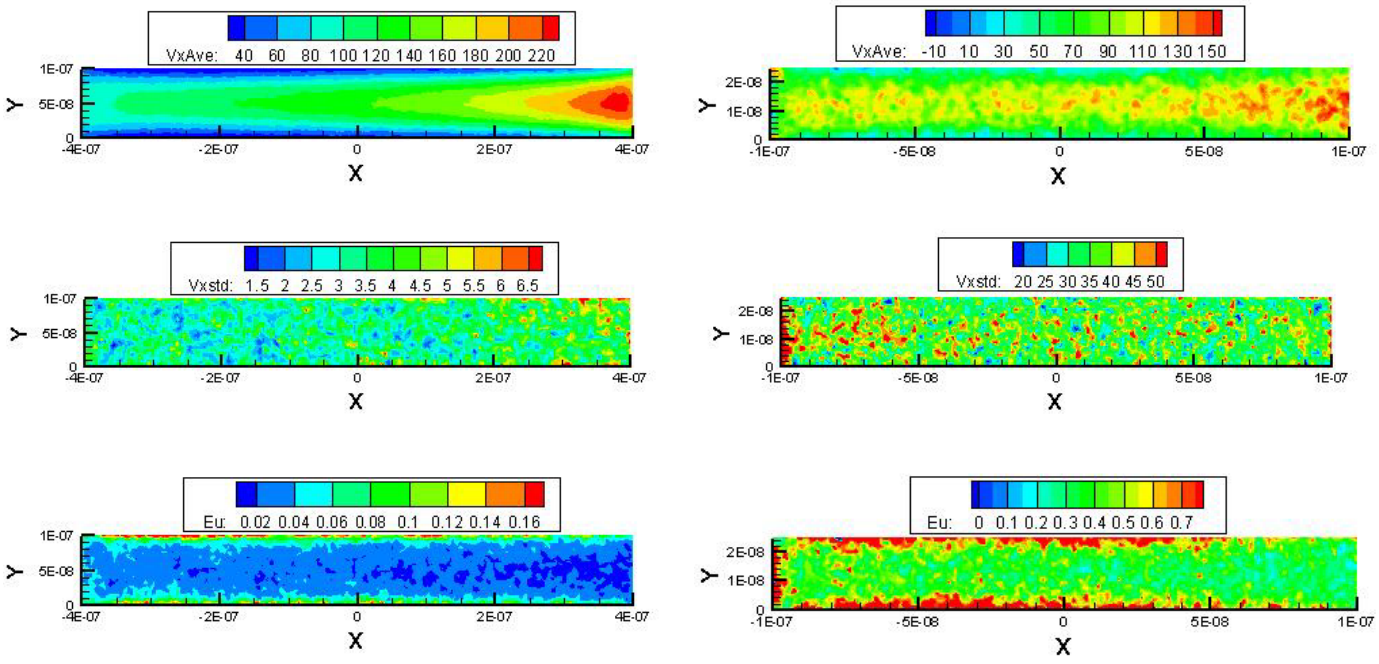
and the sample standard deviation from the local time-averaged value is calculated using:

$$x_{STD} = \sqrt{\frac{\sum_{i=1}^{N} \left(x_i - \langle x \rangle \right)}{N-1}} = \sqrt{\frac{\sum_{i=1}^{10} \left(x_i - \langle x \rangle \right)}{9}} \tag{4.11}$$

An additional value that is often useful in visualizing the local scatter in U-DSMC results is a mean weighted version of the sample standard deviation and takes the form:

$$E_x = \frac{x_{STD}}{\langle x \rangle} \tag{4.12}$$

Within this study, the U-DSMC data has been time-averaged and the values of mean, sample standard deviation, and mean weighted sample standard deviation have been calculated for the x-component velocity for each case. Figure 72 illustrates how these properties can be used to visualize the scale-induced scatter. From Figure 72 several key distinctions between the h = 100 nm case and the h = 25 nm case can be seen. The first noteworthy difference is shown in the mean velocity contours in the top plots of Figure 72. The smooth contours of the h = 100 nm case illustrate that the flow field is well captured and that the velocity varies smoothly and continuously across the domain. The sharp and blotchy contours of the h = 25 nm case illustrate that the flow field is not well captured and that the velocity varies discontinuously across the domain.



(a) **(b)**

Figure 72. Contours of x-component velocity, standard deviation in x-component velocity over data sample, and mean-weighted deviation in x-component velocity for Kn = 0.01 with *h* = 100nm (a) and *h* = 25nm (b).

The middle plots of Figure 72 illustrate the range of the local sample standard deviations in the x-component velocity for each case. The contour range of the h = 100 nm case shows that the standard deviation between each local velocity value for the data sets is modest. However, the contour range of the h = 25 nm case illustrates that the standard deviation between each local velocity value for its corresponding data sets is significantly larger than the h = 100 nm case. In the lower plots of Figure 72, a limitation of using local-mean error as a measure of the statistical fluctuations within a data set is seen. Although it has been clearly shown that the h = 100 nm case is captured with far less scatter, using the estimated values from within the U-DSMC data can lead to an internally biased measure of the error associated with the scale-induced statistical fluctuations.

In order to establish a consistent and unbiased measure of the scale-induced statistical fluctuation in U-DSMC predictions, error values based on comparison with theoretical solutions for Poiseuille flow have been developed. Using both first and second-order slip formulations, for near-continuum Knudsen values, the velocity profiles predicted for the continuum to slip regime can be generated for use as basis values. As derived in Section 3.5, the first and second order non-dimensional velocity profiles for Poiseuille flow are given by:

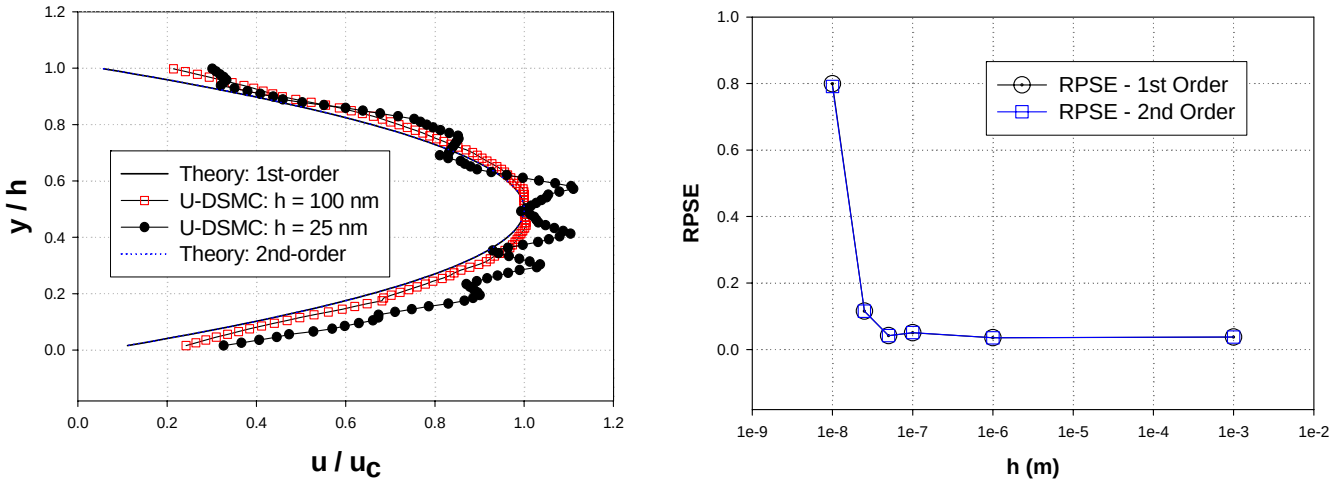
$$\frac{u_{1st}}{u_c} = \frac{\left[-\left(\frac{y}{h}\right)^2 + \frac{y}{h} + Kn\right]}{\frac{1}{4} + Kn} \tag{3.22}$$

$$\frac{u_{2nd}}{u_c} = \frac{\left[-\left(\frac{y}{h}\right)^2 + \frac{y}{h} + \frac{Kn}{1+Kn}\right]}{\frac{1}{4} + \frac{Kn}{1+Kn}} \tag{3.23}$$

Comparison of the U-DSMC predicted velocity profiles and those of the relations above are shown in Figure 73 (a). An error relation is formed between the theoretical profiles and the U-DSMC results using:

$$RPSE = \sqrt{\frac{1}{\eta}\sum_{i=1}^{\eta}\left(\langle x\rangle / \langle x\rangle_c - \xi / \xi_c\right)^2} \tag{4.13}$$

where η is the number of curve points considered, $\langle x\rangle$ is the U-DSMC data set average of the local quantity, in this case the x-component velocity, $\langle x\rangle_c$ is the centerline value of $\langle x\rangle$, ξ is the theoretically predicted local quantity and ξ_c is the centerline value of this quantity. The scale-induced error, as quantified using Eq. (4.13) is shown in Figure 73 (b).



(a) **(b)**

Figure 73. Comparison between U-DSMC predictions of x-component velocity profiles and theory for Kn = 0.01 with h = 100nm and h = 25nm (a). Plot of the error between the theoretical velocity profile prediction and U-DSMC results as a function of domain scale (b).

Figure 73 (a) clearly illustrates the velocity profiles under consideration in this study. From Figure 73 (a) the excellent agreement between the theoretical relations and the U-DSMC result for the h = 100 nm is well illustrated. Figure 73 (a) also clearly illustrates the statistical scatter in the velocity profiles for the h = 25nm case. Figure 73 (b) is a plot of the relative error between the U-DSMC results and the theoretically predicted values for all the cases studied. From Figure 73 (b) the scaled-induced error is seen to increase sharply as the channel height is decreased toward 1nm. This effect can be seen to correlate with the decrease in the number of simulated particles within the domain as the number of real molecules decreases over the simulation volume. Noting that the simulation set imposed a fixed Knudsen number of 0.01 for all cases, and also noting that the particle weight (the number of real particle represented by each simulated particle) was not allowed to reach unphysical values (a minimum of 1 was enforced)

the increase in error can be related to the decrease in the total number of real particles in the volume of interest following:

$$N_{tot} = nV = \frac{L^3}{\sqrt{2}\,\pi\, d^2\, Kn\, L} = \frac{L^2}{\sqrt{2}\,\pi\, d^2\, Kn} \tag{4.14}$$

where d is the molecular diameter, n is the overall number density, V is the domain volume and L is the characteristic length of the domain. From this relation it can be seen that for a fixed Knudsen number the total number of particles in the domain is proportional to L^2. Thus, as the length decreases by a factor of 10 the total number of particles drops by a factor of 100. As such, the scale-induced error is nonlinear and as shown in Figure 73 (b) and increases as $1/L^2$ as the characteristic length is decreased toward 1nm.

4.4.2 Conclusions

U-DSMC studies of scale-induced statistical fluctuations in Poiseuille flow have shown that the error between theoretical formulations and U-DSMC velocity profiles increase as the inverse of the square of the characteristic length, as the characteristic length approaches a value of 1nm. Furthermore, the studies have established a number of error measurement relations as well as introduced means of visually depicting statistical fluctuations in U-DSMC results.

5. CONCLUSIONS & RECOMMENDATIONS

5.1 Summary

The implementation of the three dimensional direct simulation Monte Carlo methodology on unstructured Delaunay meshes (U-DSMC) has been achieved. Extensive geometric flexibility as well as improved ease-of-use was achieved through the creation of an interface between U-DSMC and COMSOL. The verification and validation of the U-DSMC implementation was shown using a series of fundamental flow cases. The order of error associated with the implementation was also studied using a fundamental flow configuration. For the case of near-continuum heat transfer between parallel plates the current U-DSMC implementation exhibits first order error in both space and time as seen from the studies of cell spacing and time step size. The study also revealed an inverse first order error associated with the number of simulation particles in each Delaunay cell.

The expansion of argon from microtubes into vacuum has been extensively investigated using the U-DSMC code. U-DSMC investigations of the effect of Knudsen number, aspect ratio, Reynolds number and microtube scale on plume structure have been made. The plume profile has been found to narrow with increasing Knudsen numbers, as well as increasing the aspect ratio. However, the plume shape has also been found to narrow with decreasing Reynolds number. The relative number density drop along the flow axis has been found to decrease with increasing Knudsen number and increasing Reynolds number. Conversely, a decrease in the relative number density drop has been observed for decreasing aspect ratio. From these investigations a similarity between the effects of aspect ratio and Reynolds number is seen. Furthermore, results obtained to date imply that the current U-DSMC implementation is a

scalable method. Results of this study indicate that modeling gaseous expansion from tubes using U-DSMC is scalable with Reynolds number and Knudsen number.

A parametric study has been carried out over a range of Knudsen number, Reynolds number and speed ratio values. The resulting plume profiles were compared for each set of parameters. From the comparisons it has been found that the plume profile narrows as both the Knudsen number and speed ratio are increased for a fixed Reynolds number. Furthermore, the relative drop in the number density along the flow path was found to decrease with increasing Knudsen number and speed ratio for a fixed Reynolds number. A slight narrowing of the plume was seen as both the Reynolds number and speed ratio were increased for a fixed Knudsen number. While the relative drop in the number density along the flow path clearly decreases with increasing Reynolds number and speed ratio for a fixed Knudsen number. It was further observed that the plume profile is not significantly altered when increasing the Knudsen number while decreasing the Reynolds number with the speed ratio fixed. With the exception of an increase in statistical fluctuations, the current U-DSMC implementation has been found to be scale insensitive.

Results of a numerical study using the U-DSMC code were employed to guide the design of a micropitot probe intended for use in analyzing rarefied gaseous microjet flow. The flow conditions considered correspond to anticipated experimental test cases for a probe that is currently under development. The expansion of nitrogen from an orifice with a diameter of 100μm was modeled using U-DSMC. From these results, local ‘free stream’ conditions were obtained for use in U-DSMC simulations of the flow in the vicinity of the micropitot probe. For the conditions of the current design, the theory of Hughes and de Leeuw (1965) can be used to determine the sensor pressure for the micropitot probe with acceptable accuracy. Also, the

required pressure range of a sensor implemented in the current experimental configuration is from 100 to 0.005 Pa (or 0.75 to 3.75e-5 torr). Furthermore, the U-DSMC results indicate that the effect of the blunt entrance geometry on the probe-local external flow field is negligible in the investigated Knudsen range. Additionally, it is seen that the chosen length of the microtube does not markedly impede the transition of the external flow through the internal region in a manner not captured by the Hughes and de Leeuw theory.

The effect of scale on the statistical fluctuation of the U-DSMC data was studied using Poiseuille flow. The error in the predicted velocity profile was calculated with respect to both first and second-order slip formulations. Simulations were carried out for a range of channel heights and the error between the U-DSMC predictions and theory were calculated for each case. From this error, a functional dependence was shown between the scale-induced statistical fluctuations and the decreasing channel height for fixed Knudsen number and pressure ratio flow. U-DSMC studies have shown that the error between theoretical formulations of velocity profiles and U-DSMC velocity profiles increase as the inverse square of the characteristic length, as the characteristic length approaches a value of 1nm. Furthermore, the studies have established a number of error measurement relations as well as introduced means of visually depicting statistical fluctuations in U-DSMC results.

5.2 Recommendations for Future Work

It is the author's contention that the effort applied over the course of this work represents a transitional phase in the development of U-DSMC. The current implementation of U-DSMC was built upon the foundation of Hammel (2002) and Kovalev (2000) as well as the contributions of Spirkin (2006). Therefore, it is expected that future studies and additional

algorithmic development lie ahead for U-DSMC. The following listing covers key points that would significantly extend the efforts of this work as wells as those incorporated in the work mentioned above.

- Algorithmic additions and modifications:

1. Re-implementation of vibrational energy modeling.

   Within the work of Hammel (2002), the initial implementation of methodologies which model vibrational energy of polyatomic molecules was shown. However, the current version of the code does not contain a working algorithm to model vibrational energy. Therefore, re-implementation of the surviving subroutines into a complete algorithm would extend the ability of U-DSMC to model high temperature flows in which the vibrational modes are excited.

2. Implementation of chemical reaction modeling.

   It has been shown in this work that U-DSMC can be applied to a wide range of applications. Several potential applications of the method would require the modeling of chemical reactions within a gas mixture. Therefore the implementation of chemical reaction modeling would greatly enhance the codes applicability for both re-entry flows as well as lab-on-chip MEMS devices.

3. Implementation of localized particle weighting.

   Within many of the studies carried out within this work the results suffered from localized scatter due to an insufficient number of simulation particles in collision and sampling volumes within finely meshed regions of the flow. Frequently there is a requirement to size Delaunay cell edges based on geometric resolution rather than flow gradient. Localized particle weighting could compensate for the reduced

number of simulation particles that inhabit these cells, thus resulting in reduced statistical fluctuation as well as an overall decrease in computational effort.

4. Implementation of detailed gas-surface modeling.

   It was shown within this work that the detailed interaction between gas molecules and solid surfaces become increasingly important as the characteristic length scale is decreased. Therefore, detailed modeling of gas-surface interactions is required for submicron scaled flows. Coupling the U-DSMC method with Molecular Dynamics simulations could be one method of capturing the detailed interactions that occur along the solid boundaries of nanoscaled flows.

5. Parallelization of U-DSMC.

   In order to take full advantage of the geometric flexibility of the U-DSMC method, parallelization of the underlying algorithms should be carried out. Parallelization of U-DSMC would greatly extend its range of applicability and thus allow for complete system modeling of both high-altitude aircraft and spacecraft as well as lab-on-chip devices.

- Additional efforts into micro and nanotube expansion as well as micro and nanonozzle expansion:

  1. Additional U-DSMC studies.

     The studies carried out within this work have uncovered a functional relation between the plume characteristics and several key parameters. However, the data collected to date from the U-DSMC simulations is not comprehensive enough to

establish the functionality of this relation. Therefore further studies should be carried out in order to provide the data needed to define the functionality.

2. Development of a refined plume model that is a function of *S*, *Kn*, and *Re*.

   From the data collected within the additional U-DSMC studies, a functional relation between the plume characteristics and the key parameters could be developed. This relation would greatly extend the analytical predication capabilities available for use in expansion applications.

3. Further study using U-DSMC to determine the effects of gas composition.

   The effects of mass, internal degrees of freedom, and mixture composition should be studied in detail using U-DSMC to determine if there are any additional functional dependencies within the plume expansion on these parameters.

- Extension and experimentation of the micropitot probe development:

1. Sensor selection.

   In order to further the computational predications of U-DSMC, component selection for the micropitot probe should be made and system design should be finalized.

2. U-DSMC simulations of micropitot probe system.

   Once a finalized system design has been developed, revised simulations should be carried out in order to determine the applicability of the theoretical models as well as to determine the response time of the pressure measurements.

3. Additional U-DSMC simulations featuring background pressure.

In order to fully predict the experimental measurements using U-DSMC the effects of background pressure would need to be investigated. Determining the range of background pressures should be possible presently as the proposed experimental facility is currently well underway toward its fully-operational status.

- Further study of scale-induced effects using U-DSMC:
  1. Numerical studies utilizing other fundamental flow scenarios.

     Further U-DSMC study should be carried out using a combination of fundamental flow scenarios such as Couette flow or flow over a flat plate, in order to further determine the functionality of scale-induced statistical fluctuations in sampling of macroscopic properties in flow scenarios where the characteristic length scale approaches 1nm.
  2. Application of detailed surface interactions to determine the applicability of the diffuse model.

     Upon the completion of a detailed surface model, the range of applicability of the diffuse model could be determined by means of several comparison cases with results from U-DSMC using both methods. From these comparisons, the length scale where diffuse modeling becomes inappropriate could be determined.

REFERENCES

Albini, F. A., "Approximate Computation of Underexpanded Jet Structure," *AIAA Journal*, Vol. 3, 1965, pp. 1535-1537.

Alexander, F. J., Garcia, A. L., and Alder, B. J., "Cell size dependence of transport coefficients in stochastic particle algorithms," *Phys. Fluids*, Vol. 10, No. 6, 1998, pp.1540-1542.

Allegre, J., Raffin, M., Choun, A., and Gottensdiener, L. "Rarefied Hypersonic Flow over a Flat Plate with Truncated Leading Edge," *19th Symposium on Rarefied Gas Dynamics*, 1993.

Ashkenas, H. and Sherman, F. S., "The Structure and Utilization of Supersonic Free Jets in a Low Density Wind Tunnel," *Proceedings of the 5th RGD*, Vol. 2, pp.84-105, Academic Press, New York, 1966.

Balakrishnan, J., Boyd, I. D., and Braun, D. G., "Monte Carlo Simulation of Vapor Transport in Physical Vapor Deposition of Titanium," *Journal of Vacuum Science an Technology A*, Vol. 18, 2000, pp. 907-916.

Bergemann, F. and Boyd, I. D., "New Discrete vibrational energy model for the direct simulation Monte Carlo method," *Progress in Aeronautics and Astronautics*, Vol. 158, 1994, pp. 174-183.

Beskok, A. "Simulations and Models for Gas Flows in Microgeometries" PhD Thesis, Princeton University, 1996.

Bird, G. A., "Monte Carlo simulations in an engineering context," *Progr. in Astro. Aero.*, Vol. 74, 1981, pp. 239-255.

Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford Univ. Press, Oxford, 1994.

Borgnakke, C., and Larsen, P. S., "Statistical collision model for Monte Carlo simulation of polyatomic gas mixture," *J. Comp. Phys.*, Vol. 18, 1975, pp.405-420.

Borouchaki, H. and George, P., "Aspects of 2-D Delaunay Mesh Generation," *Int. Journal for Numerical Methods in Eng.*, Vol. 40, 1997, pp. 1957-1975.

Boyd, I. D., "Relaxation of discrete rotational energy distributions using a Monte Carlo Method," *Phys. Fluids A*, Vol. 5, No. 9, 1993, pp. 2278-2286.

Boyd, I. D., "Computation of Atmospheric Entry Flow About a Leonid Meteoroid," *Earth, Moon, and Planets*, Vol. 82, 2000, pp. 93-108.

Boyd, I.D., Penko, P. F., Meissner, D. L. and DeWitt, K. J., "Experimental and Numerical Investigations of Low-Density Nozzle and Plume Flows of Nitrogen," *AIAA Journal*, Vol. 30, 1992, pp. 2453-2461.

Boyd, I. D., Jafry, Y. and Banden Beukel, J., "Particle Simulations of Helium Microthruster Flows," *Journal of Spacecraft and Rockets*, Vol. 31, 1994, pp. 271-277.

Boyd, I. D. and Wang, W.-L., "Monte Carlo Computations of Hypersonic Interacting Flows," *Proc. 39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, January 2001.

Boynton, F. P., "Highly Underexpanded Jet Structure: Exact and Approximate Calculations," *AIAA Journal*,Vol. 5, 1967, pp. 1703-1704.

Campbell, D. H., Wysong, I. J., Weaver, D. P., and Muntz, E. P., "Flowfield Characteristics in Free Jets of Monatomic and Diatomic Gases," *Proceedings of the 18th Rarefied Gas Dynamics Symposium*, Vancouver, British Columbia, July 1992.

Cercignani, S. and Lampis, M., *Rarefied gas dynamics*, Academic Press, New York, 1974.

Chamberlin, R. and Gatsonis, N. "DSMC Simulation of Microjet Expansion and the Design of a Micro Pitot Probe," *Proc. of the 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conf.*, San Francisco, CA, June, 2006 (a).

Chamberlin, R. and Gatsonis, N. "Numerical Modeling of Gas Expansion from Microtubes," *Proc. of the 4th International Conf. on Nanochannels, Microchannels and Minichannels*, Limerick, Ireland, June, 2006 (b).

Chamberlin, R. and Gatsonis, N. "Numerical Modeling of Gaseous Expansion from Micro and Nano Nozzles," *Proc. of the 25th Int. Symp. on Rarefied Gas Dynamics*, Saint-Petersburg, Russia, July, 2006 (c)

Chamberlin, R. and Gatsonis, N. "DSMC Simulation of Gaseous Expansion from Microtubes," *To appear in The Journal of Nanoscale & Microscale Thermophysical Engineering*, 2007

Chapman, S. and Cowling, T., *The Mathematical Theory of Non-Uniform Gases*, Cambridge University Press, London, 1939.

Chen, G. and Boyd, I., "Statistical error analysis for the direct simulation Monte Carlo technique," *Journal of Computational Physics*, Vol. 126, 1996, pp. 434-448.

Chen, G. and Boyd, I. D., Roadman, S., and Engstrom, J. R., "Monte Carlo Simulation of a Hyperthermal Silicon Deposition Process," *Journal of Vacuum Science and Technology A*, Vol. 16, 1998, pp. 689-699.

Dettleff, G. and Plahn, K., "Experimental Investigation of Fully Expanding Free Jets and Plumes," *Proceedings of the 21st Rarefied Gas Dynamics Symposium*, Marseille, France, July, 1998.

Edwards, R. H., and Quan, V., "Hypersonic Adiabatic Impact Pressure of a Rarefied Gas," *Phys. Fluids*, Vol. 9, No. 4, 1966, pp.663-670.

Fan, C., and Robertson, S. J., "Monte Carlo Solutions of Mass, Momentum and Energy Transfer for Free Molecule and Near-Free Molecule Flow Through Circular Tubes," *Proc. of the 6th Int. Symp. on Rarefied Gas Dynamics*, Ed. L Trilling and H. Wachman, 1969, pp. 655-666.

Garcia, A. L., and Wagner, W., "Time step truncation error in direct simulation Monte Carlo," *Phys. Fluids*, Vol. 12, No. 10, 2000, pp.2621-2633.

Gatsonis, N. A., Maynard, E., and Erlandson, R. E., "Monte Carlo Modeling and Analysis of Pressure Sensor Measurements During Suborbital Flight," *Journal of Spacecraft and Rockets*, Vol. 34, No.1, 1997, pp.83-91.

Gombosi, T. I.**,** *Gaskinetic Theory*, Cambridge University Press, Cambridge, MA, 1994.

Haas, B. L., and Boyd, I. D., "Models for direct Monte Carlo simulation of coupled vibration-dissociation," *Phys. Fluids A*, Vol. 5, No. 2, 1993, pp.478-489.

Haas, B. L., Hash, D. B., Bird, G. A., Lumpkin, F. E., and Hassan, H. A., "Rates of thermal relaxation in direct simulation Monte Carlo methods," *Phys. Fluids A*, Vol. 6, No. 6, 1994, pp. 2191-2201.

Hadjiconstantinou, N. G., "Analysis of discretization in the direct simulation Monte Carlo," *Phys. Fluids*, Vol. 12, No. 10, 2000, pp. 2634-2638.

Hadjiconstantinou, N. G., Garcia, A. L., Bazant, M. Z., and He, G., "Statistical error in particle simulations of hydrodynamic phenomena," *Journal of Computational Physics.*, Vol. 187, 2003, pp. 274-297.

Hammel, J., "Development of Unstructured 3-D Direct Simulation Monte Carlo/Particle-In-Cell Code and the Simulation of Microthruster Flows," M.S. Thesis, Worcester Polytechnic Institute, 2002.

Hammel, J., Kovalev, K., and Gatsonis, N. A., "Unstructured Adaptive Monte Carlo Simulations of Flows in Micronozzles," *Proc. of the 35th AIAA Thermophysics Conference,* Anaheim, CA, 2001.

Hubbard, E. W., "Approximate Calculation of Highly Underexpanded Jets," *AIAA Journal*, Vol. 4, 1966, pp. 1877-1879.

Hughes, P. C. and de Leeuw, J. H., "Theory for the Free Molecule Impact Probe at an Angle of Attack," *Rarefied Gas Dynamics*, Vol.1, 1965.

Kalos, M. H., and P.A. Whitlock, *Monte Carlo Methods, Volume 1: Basics*, John Wiley & Sons, New York, 1986.

Kannenberg, K. C., and Boyd, I.D., "Monte Carlo Computation of Rarefied Supersonic Flow into a Pitot Probe," *AIAA Journal*, Vol. 34, No. 1, 1996, pp. 83-88.

Kannenberg, K. C., and Boyd, I.D., "Three Dimensional Monte Carlo Simulations of Plume Impingement," *Journal of Thermophysics and Heat Transfer*, Vol. 13, 1999, pp. 226-235.

Karipides, D. P., Boyd, I. D., and Caledonia, G. E., "Detailed Simulations of Surface Chemistry Leading to Spacecraft Glow," *Journal of Spacecraft and Rockets*, Vol. 36, 1999, pp. 566-572.

Karniadakis, G. E., Beskok, A., MICRO FLOWS Fundamentals and Simulation, Springer-Verlag, New York, 2002.

Kohler, J., Bejhed, J., Kratz, H., Bruhn, F., Lindberg, U., Hjort, K., and Stenmark, L., "A hybrid cold gas microthruster system for spacecraft," *Sensors and Actuators A*, 97-98, 2002, pp.587-598.

Kovalev, K., "Development of a 2-D/3-D Unstructured Adaptive Grid Generator for Direct Simulation Monte Carlo Computations," Master's Thesis, Worcester Polytechnic Institute, 2000.

Lewis, D. H., Janson, S. W., Cohern, R. B., and Antonsson, E. K., "Digital micropropulsion," *Sensors and Actuators*, 80, 2000, pp. 143-154.

Liou, W. and Fang, Y. "Implicit Boundary Conditions for Direct Simulation Monte Carlo Methods in MEMS Flow Predictions," *CMES*, Vol. 1, 2000, pp. 119-128.

Lohner, R. and Ambrosiano, J., "A Vectorized Particle Tracer for Unstructured Grids," *Journal of Computational Physics*, Vol. 91, 1990, pp. 2-31.

Lordi, J. A., and Mates, R. E., "Rotational relaxation in nonpolar diatomic molecules," *Phys. Fluids*, Vol. 13, 1970, pp. 291-308.

Marchetti, P. "Electric Propulsion and Controller Design for Drag-Free Spacecraft Operation in Low Earth Orbit" M.S. Thesis, Worcester Polytechnic Institute, 2006.

Maxwell, J. C., *Phil. Trans. Roy. Soc. 1*, Appendix, 1879.

Maynard, E. P., "Direct Simulation Monte Carlo Analysis of Rarefied Gas Flow in Tubes," M.S. Thesis, Worcester Polytechnic Institute, 1996.

Micci, M. and Ketsdever, A., *Micropropulsion for Small Spacecraft*, AIAA, Virginia, 2000.

Mo, H. L., Gadepalli, V. V., Skudarnov, P. V., and Lin, C. X., "Continuum Modeling of Gas Flows in Micronozzles," *Proc. of the 3rd International Conference on Microchannels and Minichannels*, Toronto, Canada, 2005.

Nance, R., Hash, D., and Hassan, H., "Role of Boundary Conditions in Monte Carlo Simulations of MEMS Devices," *Proc. of the 35th Aerospace Sciences Meeting and Exhibit*, Reno, NV, 1997.

Rjasanow, S. and Wagner, W., "On time counting procedures in the DSMC method for rarefied gases," *Mathematics and Computers in Simulation*, Vol. 48, 1998, pp. 151-176.

Rossi, C., Oriuex, S., Larangot, B., Do Conto, T., and Esteve, D., "Design, Fabrication and modeling of solid propellant microrocket – application to micropropulsion," *Sensors and Actuators A*, Vol. 99, 2002, pp. 125-133.

Rubinstein, R. Y., *Simulation and the Monte Carlo method*, John Wiley & Sons, New York, 1981.

Schaaf, S. and Chambre, P. , *Flow of Rarefied Gases*, Princeton Univ. Press, Princeton, 1961.

Sharipov, F., "Rarefied Gas Flow into Vacuum Through Thin Orifice: Influence of Boundary Conditions," *AIAA Journal*, Vol. 40, 2002, pp. 2006-2008.

Spirkin, A. M., "A Three-dimensional Particle-In-cell Methodology on Unstructured Voronoi Grids with Applications to Plasma Microdevices," Ph.D. Dissertation, Worcester Polytechnic Institute, 2006.

Wagner, W. "A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation," *Journal of Statistical Physics*, Vol. 66, 1992, pp. 1011-1044.

Wang, M. and Li, Z. "Simulations for gas flows in microgeometries using the direct simulation Monte Carlo method," *Int. Journal of Heat and Fluid Flow*, Vol. 25, 2004, pp. 975-985.

Watson, D., "Computing the Delaunay Tessellation with Application to Voronoi Prototypes," *The Computer Journal*, Vol. 24, 1981, pp. 167-172.

White, F., *Fluid Mechanics*, WCB/McGraw-Hill, Boston, 1999.

Whitfield, D. and Janus, J., "Three-Dimensional Unsteady Euler Equation Solutions Using Flux Vector Splitting," *AIAA*, Paper 84-1552, 1984.

Wu, J. S., and Lian, Y. Y., "Parallel three-dimensional direct simulation Monte Carlo method and its applications," *Computers & Fluids*, Vol. 32, 2003, pp. 1133-1160.

Wu, J. S. and Tseng, K. C., "Parallel DSMC method using dynamic domain decomposition," *International Journal for Numerical Methods in Engineering*, Vol. 63, 2005, pp. 37-76.

Wu, J. S., Tseng, K. C., and Kuo, C. H., "The direct simulation Monte Carlo method using unstructured adaptive mesh and its application," *International Journal for Numerical Methods in Fluids*, Vol. 38, 2002, pp. 351-375.

APPENDIX A. Sampling From Distribution Functions

Cumulative Distribution Function Method:

One method of sampling a given distribution function is the cumulative distribution function method (CDF method) (Rubenstein, 1981, Kalos and Whitlock,1986). The fundamental concept underlying the CDF method is a mapping of the distribution variable to a uniformly distributed set of numbers. For purposes of computer based simulations the set of uniformly distributed numbers is most often taken as a pseudorandom set distributed between 0 and 1, which are readily available by means of most standard random number generators. Throughout this section a given number from the uniform set of numbers generated by a random number generator will be written as R.

For a given distribution, $f(x)$, the cumulative distribution, $F(x)$, can be written as

$$F(x) = \frac{\int_a^x f(x')dx'}{\int_a^b f(x')dx'} \tag{A.1}$$

Note that by the definition above the cumulative distribution, $F(x)$, is normalized even if $f(x)$ is not. Accordingly $F(x)$ can be viewed as the required map of $f(x)$ to a uniformly distributed set of numbers ranging from 0 to 1. Although there are limitations to the above definition of the cumulative distribution, this form is sufficient for the distribution functions encountered most frequently in particle methods. For cases investigated in this work the mapping of $f(x)$ to a uniformly distributed set of random numbers allows for the variable of the distribution, to be written in terms of a random number R for all distributions where

$$F(x) = \frac{\int_a^x f(x')dx'}{\int_a^b f(x')dx'} = R \tag{A.2}$$

can be solved explicitly for $x(R)$. In cases where an analytical expression cannot be found numerical tabulation may also be used to define an x for a given R.

Box-Muller Method:

The Box-Muller method is in essence the application of the CDF method to a normal or Gaussian distribution by means of a clever mathematical trick. For a normal distribution the form of the distribution function is given by the following, where μ is the mean and σ^2 is the variance,

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x-\mu)^2}{2\sigma^2} \right). \tag{A.3}$$

Applying the CDF method to a distribution of this form with a mean of 0 and a variance of 1 would result in a cumulative distribution of

$$F(x) = \frac{\int_a^x \exp\left(\frac{-(x')^2}{2} \right) dx'}{\int_a^b \exp\left(\frac{-(x')^2}{2} \right) dx'}. \tag{A.4}$$

Upon evaluation of the integrals the final form of the cumulative distribution would be found to be

$$F(x) = \frac{erf(x/\sqrt{2}) - erf(a/\sqrt{2})}{erf(b/\sqrt{2}) - erf(a/\sqrt{2})} = R, \tag{A.5}$$

where *erf* is the error function. Unfortunately the above cumulative distribution cannot be explicitly solved for $x(R)$. However, if a product of two independent normal distributions are considered a coordinate transform can be carried out. For example take

$$f(x) f(y) = \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(x)^2}{2} \right) \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-(y)^2}{2} \right) = \left(\frac{1}{2\pi} \right) \exp\left(\frac{-(x^2+y^2)}{2} \right). \tag{A.6}$$

Applying the standard transform of Cartesian coordinates to polar coordinates, such that

$$\begin{aligned} x &= r\cos(\theta) \\ y &= r\sin(\theta) \end{aligned} \tag{A.7}$$

yields a transformation of the cumulative distribution integrals that result in

$$F(r,\theta)=\frac{\int_0^r \int_0^\theta \exp\left(\frac{-(r')^2}{2}\right) r'\,dr'\,d\theta'}{\int_0^\infty \int_0^{2\pi} \exp\left(\frac{-(r')^2}{2}\right) r'\,dr'\,d\theta'}=R. \tag{A.8}$$

Since the distribution is already normalized the denominator is unity and thus the expression reduces to

$$F(r,\theta)=\int_0^r \int_0^\theta \exp\left(\frac{-(r')^2}{2}\right) r'\,dr'\,d\theta'=R \tag{A.9}$$

Carrying out the integral above yields

$$F(r,\theta)=\left[1-\exp\left(-\frac{r^2}{2}\right)\right]\theta=R \tag{A.10}$$

Noting that the angle θ varies uniformly over the interval between 0 and 2π, and further noting that r and θ are independent, a value for θ can be generated by

$$\theta=2\pi R_1. \tag{A.11}$$

While a value for r can be calculated by

$$r=\sqrt{-2\log(1-R_2)}, \tag{A.12}$$

which for a random number distributed between 0 and 1 the relation can be simplified to

$$r=\sqrt{-2\log(R_2)} \tag{A.13}$$

Therefore, by the Box-Muller method a value of x from $f(x)$ can be generated by

$$x=r\cos(\theta)=\sqrt{-2\log(R_2)}\cos(2\pi R_1) \tag{A.14}$$

and likewise a value for y from $f(y)$ is generated from

$$y=r\sin(\theta)=\sqrt{-2\log(R_2)}\sin(2\pi R_1) \tag{A.15}$$

with x and y being completely independent of one another. Sampling a large number of x values using a quality random number generator will thus generate the distribution $f(x)$. It is

important to note that the clever trick of considering two independent normal distributions allows for a closed form relation between x and a pair of random numbers by following the CDF method, a result which was unattainable using a single normal distribution.

Acceptance-Rejection Method

In the event that the CDF method does not return an explicit expression for x as a function of R and a mathematic trick cannot be found to alleviate this difficulty an approach referred to as the acceptance-rejection method can be used (Bird, 1994). In order to relate the x value to a random fraction the distribution is normalized by its maximum value,

$$f'(x) = \frac{f(x)}{f_{MAX}} \tag{A.16}$$

Since the values of x themselves are uniformly distributed between the lower and upper limits, a and b, the distribution of x can be written as

$$f_x = \frac{1}{(b-a)} \tag{A.17}$$

If the CDF method is then applied to the above distribution function the cumulative distribution is found to be

$$F_x = \frac{\int_a^x f_x dx'}{\int_a^b f_x dx'} = \frac{(x-a)}{(b-a)} = R \tag{A.18}$$

From which the direct relation between x and R is calculated to be

$$x = a + R(b-a) \tag{A.19}$$

Using the above relation to generate an x for a given random fraction becomes rather straight forward. Upon calculating x, the normalized distribution, $f'(x)$, can be evaluated. A second random fraction is then generated. If the value of $f'(x)$ is less than the second R the value for x is accepted, if not the entire process is repeated until an acceptable value for x is generated.

The obvious draw back of this method is the lack of a one-to-one relation that would insure an accepted value of x for a given random number. The advantage of this method is its applicability to an extremely wide range of distributions, since it doesn't suffer from the same limitations of the CDF method.

Loading: Extension to a Maxwellian Distribution

Since the results presented in the previous sections introduced and contrasted three methods of sampling a distribution, more precisely a normal distribution with zero mean and unit variance, further extension of the methodologies is required for sampling from distributions common to particle-based simulations. One such distribution often encountered in particle methods is the Maxwellian Distribution of Velocity, which describes the state of the velocity distribution amongst molecules in a gas at equilibrium. As regular practice in the majority of gas-dynamic particle simulations the initial loading of the computational domain is carried out by means of sampling a Maxwellian Distribution. The form of the Maxwellian Distribution of Velocity used by Bird (1994) is

$$f_0 = \left(\frac{\beta^3}{\pi^{3/2}} \right) \exp\left(-\beta^2 \mathbf{c}'^2 \right) \tag{A.20}$$

with

$$\beta = \sqrt{\frac{m}{2k_B T}} \tag{A.21}$$

and

$$\mathbf{c}' = \mathbf{c} - \mathbf{c}_0 \tag{A.22}$$

where $\mathbf{c}'$ is the peculiar or thermal velocity, $\mathbf{c}_0$ is the macroscopic or drift velocity and $\mathbf{c}$ is the velocity of the molecule. For Cartesian coordinates, the distribution for a single velocity component can be written as

$$f_u = \left(\frac{\beta}{\sqrt{\pi}} \right) \exp\left(-\beta^2 (u' + u_0)^2 \right) \tag{A.23}$$

with

$$u = u' + u_0 \tag{A.24}$$

where u is the molecule's x-velocity component, u' is the x-component of the thermal velocity and u_0 is the x-component of the macroscopic velocity. A few points are worth noting here. First of all, there is nothing special about the x-component and the other two components will share the same form as above. Furthermore, each component is independent of the other. Also,

the thermal velocity is independent of the macroscopic velocity and therefore the expression above can be separated into to independent distributions, one for thermal velocity the other indicating the drifting effect.

In order to ease the work load of re-deriving the expressions used in the three sampling methods for a Maxwellian Distribution it is convenient to simply point out the similarities between the normal distribution and the Maxwellian Distribution. If we rewrite the single component velocity distribution in terms of the thermal velocity we see

$$f_{u'} = \left(\frac{\beta}{\sqrt{\pi}} \right) \exp\left(-\beta^2 (u - u_0)^2 \right). \tag{A.25}$$

If we compare the form of the above distribution with the normal distribution, rewritten below for ease of comparison, similar terms can be related.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-\left(x-\mu\right)^2}{2\sigma^2} \right) \tag{A.26}$$

The first term to find the counterpart of is β, for which we find

$$\beta \equiv \frac{1}{\sqrt{2}\sigma} \tag{A.27}$$

Likewise it is clear that the drift term is equivalent in effect to the mean, thus

$$u_0 \equiv \mu. \tag{A.28}$$

From these relations a simple modification to the results for a normal distribution will in turn give the matching results for a Maxwellian Distribution. Therefore the Box-Muller relations for the distribution function for thermal velocity is

$$u' = \frac{\sqrt{-\log(R_2)}}{\beta} \cos(2\pi R_1) \tag{A.29}$$

which is taken from the Box-Muller method applied to a normal distribution with non unity variance (derivation not shown). Since the macroscopic velocity is independent of the thermal velocity the two can be uncoupled and analyzed separately with the final result of the product

being no more than the linear sum of the two solutions. Therefore a given molecule's velocity can be computed from

$$u = u' + u_0 = u' = \frac{\sqrt{-\log(R_2)}}{\beta}\cos(2\pi R_1) + u_0 \tag{A.30}$$

by means of the Box-Muller method. For the other two approaches the same linearity applies, therefore the value for the drift velocity can simply be added to the final value (after acc.-reject. for example) of the thermal velocity to comprise the molecule's velocity. Likewise each component of the molecule's velocity can be calculated such that each component independently satisfies its respective distribution function and as a result the overall set of molecules will together fill the corresponding 3-dimensional Maxwellian Distribution of Velocity.

The current implementation for loading used in U-DSMC follows the Box-Muller method as described above for the determination of the loaded particle's velocity components.

Injection: Extension to Maxwellian Flux

Another key step in modeling gas flows by means of particle methods is the proper capturing of the flux of molecules at the boundaries of the domain. For many cases a Maxwellian Flux is assumed for others a drifting Maxwellian Flux is more appropriate. Following section 7.1 of Gombosi (1994) the flux of particles across a surface element can be given as a relation between the surface normal and the corresponding distribution functions. If a set of local coordinates are attached to the surface as shown in the figure below the macroscopic motion of the flow can be related to the local coordinates such that c_{01} is the component of the drift in the x_1 direction and likewise for the other two components. Thus we find:

$$\begin{aligned} c &= c' + c_0 \\ c_1 &= c'_1 + c_{01} \\ c_2 &= c'_2 + c_{02} \\ c_3 &= c'_3 + c_{03} \end{aligned} \tag{A.31}$$

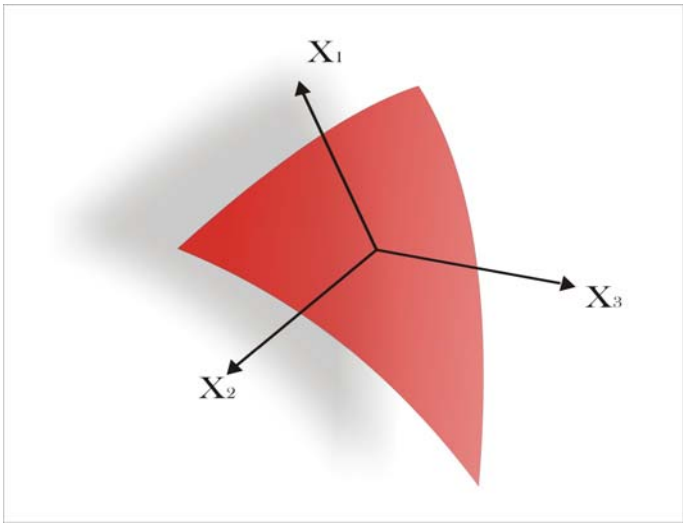


Figure 74. Local coordinate system on a surface element.

Using the localized values, the flux of a quantity, Q, through a surface which is perpendicular to the x_3 axis, is given by (adapted from Bird, 1994)

$$\dot{Q}_3 = n\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{-\infty} Qc_3 f_{c'_1} f_{c'_2} f_{c'_3} dc_3 dc_2 dc_1 \tag{A.32}$$

with,

$$f_{c'_i} = \frac{\beta}{\sqrt{\pi}}\exp\left(-\beta^2 c'^2_i\right) \tag{A.33}$$

A few points are worth noting about this formulation. First, the integral over c_1 is only over the positive values since the x_3 axis is in the positive normal direction (direction of flux) and since only particles with a c_1 component in the positive x_3 direction can cross the surface. Second, the distribution function for each component used above corresponds to the thermal distributions as used by Bird (1994) and Gombosi (1994). The number flux through a surface with the surface normal aligned in the x_3 direction is given as

$$\dot{N}_3 = n\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{0}^{-\infty} c_3 f_{c'_1} f_{c'_2} f_{c'_3} dc_3 dc_2 dc_1 . \tag{A.34}$$

Using a variable transform between c'_i and c_i yields

$$\dot{N}_3 = n\frac{\beta^3}{\pi^{3/2}}\int_{-\infty}^{\infty} f_{c'_1} dc'_1 \int_{-\infty}^{\infty} f_{c'_2} dc'_2 \int_{-c_{03}}^{-\infty} (c'_3 + c_{03}) f_{c'_3} dc'_3 \tag{A.35}$$

From the relation above we are reminded that each component is independent of the other two. The evaluation of each integral can be carried out to find

$$\dot{N}_3 = \frac{n}{2\sqrt{\pi}\beta}\left[\exp\left(-\beta^2 {c_{03}}^2\right) + \beta c_{03}\sqrt{\pi}\left\{1 + erf\left(\beta c_{03}\right)\right\}\right]. \tag{A.36}$$

In particle simulations a key value is the number of particles to be injected along a boundary surface with area A over the iteration time-step Δt. This value can be found from

$$N = \dot{N}_3 A \Delta t \tag{A.37}$$

which results in

$$N = \frac{nA\Delta t}{2\sqrt{\pi}\beta}\left[\exp\left(-\beta^2 {c_{03}}^2\right) + \beta c_{03}\sqrt{\pi}\left\{1 + erf\left(\beta c_{03}\right)\right\}\right]. \tag{A.38}$$

From the flux relation given above the corresponding velocity distributions of each thermal component can be extracted for the purpose of sampling from these distributions in order to set the injected particle's velocity components. The thermal velocity distributions that are retrieved are

$$\begin{aligned} f_{c'_1} &= \frac{\beta}{\sqrt{\pi}}\exp\left(-\beta^2 {c'_1}^2\right) \\ f_{c'_2} &= \frac{\beta}{\sqrt{\pi}}\exp\left(-\beta^2 {c'_2}^2\right) \\ cf_{c'_3} &= \frac{\beta}{\sqrt{\pi}}\left(c'_3 + c_{03}\right)\exp\left(-\beta^2 {c'_3}^2\right) \end{aligned} \tag{A.39}$$

The components parallel to the surface are found to correspond to the standard thermal distribution and as such may be sampled using the Box-Muller method, CDF method or Acceptance-Rejection as outlined in the previous section. The normal component however fits the form of a Maxwellian Flux and therefore must be evaluated using a new relation. If the CDF method is applied to the full drifting Maxwellian Flux shown in $cf_{c'_3}$ we find

$$F\left(c'_3\right)=\frac{\int_{-c_{03}}^{c'_3}\left(c''_3+c_{03}\right)f_{c''_3}dc''_3}{\int_{-c_{03}}^{\infty}\left(c''_3+c_{03}\right)f_{c''_3}dc''_3}$$
$$=\frac{\exp(-\beta^2{c_{03}}^2)-\exp(-\beta^2{c'_3}^2)+\beta c_{03}\sqrt{\pi}\left[erf\left(\beta c_{03}\right)+erf\left(\beta c'_3\right)\right]}{\exp(-\beta^2{c_{03}}^2)+\beta c_{03}\sqrt{\pi}\left[1+erf\left(\beta c_{03}\right)\right]} \tag{A.40}$$

It is helpful to note that if we take $c_{03}=0$ the distribution reduces to the stationary Maxwellian Flux and the cumulative distribution becomes

$$F_{0-drift}=1-\exp(-{c'_3}^2\,\beta^2) \tag{A.41}$$

If the methods of the last section are applied a closed form relation between c'_3 and a random fraction R would be

$$c'_3=\frac{\sqrt{-\ln\left(1-R\right)}}{\beta}=\frac{\sqrt{-\ln\left(R\right)}}{\beta} \tag{A.42}$$

which is equivalent to the relation given in section 12.1 of Bird (1994) {pg 259} for sampling a distribution function representing a stationary equilibrium gas crossing a surface. Unfortunately a closed form relation cannot be obtained for a drifting Maxwellian Flux and therefore the application of either the CDF or Acceptance-Rejection is required.

APPENDIX B. COMSOL Grid Generation Interface Programs

COMSOL to U-DSMC Direct: comTo3D.f:

The following program is used to covert a COMSOL grid file into a grid file formatted for use with U-DSMC. The program is written in Fortran and is compiled using Intel's Fortran complier. The program has been used extensively on Philippos. The input to the program is a COMSOL .MPHTXT file which is exported directly from COMSOL. The output of the program is a .3d file which is formatted for immediate use with the U-DSMC program.

```
!program that formats COMSOL mesh.txt output into grid.3d format

      program comTo3D
      implicit none

      real::h,xVir,yVir,zVir
      real::nx,ny,nz,x,y,z
      integer::INnodes,INFaces,INBFaces,INCells,fn1,fn2,fn3,old,i,j
      integer::iAt,nVir,in1,in2,in3,in4,ibf,ic,cfCount,in,cn1,cn2,cn3
      character(50)::InputFile,OutputFile,s1,s2,s3,s4,s5
        character(50)::s6,s7,s8,s9,s10
      character(100)::str,head1,head2,head3,head4,format
      integer,allocatable,dimension(:,:)::faces,cells
                                          !local arrays of faces,cells
      integer,allocatable,dimension(:,:)::fm_e,fm_t
      real,allocatable,dimension(:,:)::fm_p
             !incoming data arrays from fem.mesh e-bf, t-cells, p-nodes
      integer,dimension(4,3)::nIndex

!-------------------------------------------------
! [] specify input/output file names
!-------------------------------------------------
      InputFile='INPUTS/femMesh.mphtxt'
      OutputFile='OUTPUT/CD_dSphereF_5.3d'

!-------------------------------------------------
! [] specify rough node spacing:
!-------------------------------------------------
      h=1e-1

!-------------------------------------------------
! [] specify virus info. (if needed)
!from solver parser, it seems virus is not needed
!by the solver routine
!-------------------------------------------------
      nVir=0
      xVir=0.0
      yVir=0.0
      zVir=0.0

!---------- block 0: create node index arrays --------------
!                           ( cell node combos for faces)
```

```
 !group 1:
      nIndex(1,1)=1
      nIndex(1,2)=2
      nIndex(1,3)=3
 !group 2:
      nIndex(2,1)=2
      nIndex(2,2)=3
      nIndex(2,3)=4
 !group 3:
      nINDEX(3,1)=3
      nIndex(3,2)=4
      nIndex(3,3)=1
 !group 4:
      nIndex(4,1)=4
      nIndex(4,2)=1
      nIndex(4,3)=2


!---------- block 1: read in fem.mesh data from in-file--------
!open input file
      str=''
      write(str,*)"Input File:"
      str(14:)=InputFile(:)
      write(*,*) trim(str)
      open(15,file=trim(InputFile),STATUS='OLD')

!read in header:
      read(15,fmt=*)s1,s2,s3,s4,s5,s6,s7,s8,s9,s10
      write(*,*)'HEADER:'
      write(*,*)s1,s2,s3,s4,s5,s6,s7,s8,s9,s10

!loop over first block (data not needed)
      do i=1,8
        read(15,fmt=*)head1
      enddo

!first object block: nodes
      read(15,fmt=*)s1,s2,s3,s4,s5
      write(*,*)
   !skip next 4 lines
      do i=1,4
          read(15,fmt=*)head1
      enddo
!read in number of nodes
      read(15,fmt=*)INNodes,s1,s2,s3,s4,s5
      write(*,*)'# of Nodes: ',INNodes
      write(*,*)
   !skip next 2 lines
      do i=1,2
          read(15,fmt=*)s1
      enddo
!allocate array based on node number:
      allocate(fm_p(4,INNodes))            !node data array (fem.mesh)

!read in Nnodes (nx, ny, nz) and store in fm_p
  !loop over Nnodes, read in data, store needed info.
      write(*,*)'reading in node data'
```

```
      write(*,*)
      do i=1,INNodes
         read(15,fmt=*)x,y,z
         fm_p(1,i)=x
         fm_p(2,i)=y
         fm_p(3,i)=z
         fm_p(4,i)=h
      enddo

!skip over blocks of uneeded info:
      do i=1,4
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ # number of elements
      do i=1,in1+2
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ # number of parameters
      do i=1,in1+1
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ # number of domains
      do i=1,in1+1
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ # number of up/down pairs
      do i=1,in1+4
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !edges: _ # of elements
      do i=1,in1+2
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ number of parameters
      do i=1,in1+1
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ # number of domains
      do i=1,in1+1
        read(15,fmt=*)s1
      enddo
      read(15,fmt=*)in1      !_ # number of up/down pairs
      do i=1,in1+4
        read(15,fmt=*)s1
      enddo

!read in boundary faces and store in fm_e
      read(15,fmt=*)INBFaces             !#type 2: tri's, _# num. of elemets
      write(*,*)'# of boundary Faces: ',INBFaces
      write(*,*)
      write(*,*)'Reading in Boundary Face Data'
      write(*,*)
!allocate boundary face array:
      allocate(fm_e(4,INBFaces))        !boundary faces array (fem.mesh)
!skip label
      read(15,fmt=*)s1
!read in nodes and faceats for boundary faces
```

```
      write(*,*)
      do i=1,INBFaces
         read(15,fmt=*)in1,in2,in3
         fm_e(1,i)=in1+1
         fm_e(2,i)=in2+1                !femMesh.txt goes 0->n, need 1->n
         fm_e(3,i)=in3+1
      enddo
!skip line
      read(15,fmt=*)s1
      read(15,fmt=*)in1     !_ # number of parameters (don't need)
      do i=1,in1+1
        read(15,fmt=*)s1
      enddo
!get face attributes (Comsol boundary numbers)
      read(15,fmt=*)in1     !_ # number of domains  (faceAt's)
      read(15,fmt=*)s1
      do i=1,in1
        read(15,fmt=*)iAt
        fm_e(4,i)=iAt+1     !numbers are 1 off (0->n-1) instead of (1->n)
      enddo

!loop over up/down pairs (uneeded)
      read(15,fmt=*)in1     !_ # number of up/down pairs
      do i=1,in1+4
        read(15,fmt=*)s1
      enddo

!read in cell info and store in fm_t
      read(15,fmt=*)INCells        !#Type 3: Tet's: _ # number of elements
      write(*,*)'# of Cells: ',INCells
      write(*,*)
      read(15,fmt=*)s1
!allocate cell data array
      allocate(fm_t(4,INCells))            !cell data array (fem.mesh)
!read in cell node data
      do i=1,INCells
         read(15,fmt=*)in1,in2,in3,in4
         fm_t(1,i)=in1+1
         fm_t(2,i)=in2+1
         fm_t(3,i)=in3+1                !femMesh.txt goes 0->n, need 1->n
         fm_t(4,i)=in4+1
      enddo
      write(*,*)'DONE reading in data:'
      write(*,*)

!don't care about the rest of the data:
!close input file
      close(15)


!------------ Block 2: Create Boundary Faces -------------
      write(*,*)'Looping over Boundary Faces:'
      write(*,*)
!allocate needed arrays:
      in4=4*INCells
      allocate(cells(12,INCells))          !cell array - solver req.d info.
      allocate(faces(6,in4))               !face array - solver req.d info.
```

```
!initialize local arrays (-1 is a needed tag in the solver)
      do i=1,INCells
        do j=1,12
         cells(j,i)=-1
        enddo
      enddo
      do i=1,in4
        do j=1,6
         faces(j,i)=-1
        enddo
      enddo

      INFaces=0         !total # of faces in grid

!loop over all boundary faces
      do ibf=1,INBFaces
        INFaces=INFaces+1         !create new face (1-to-1 at this point)
          !set face's nodes and faceAt
        do i=1,4
          faces(i,INFaces)=fm_e(i,ibf)
        enddo
          !solver reserves faceAt=1 for interior so change 1's to 99's
        if(faces(4,INFaces).eq.1)faces(4,INFaces)=99
      enddo


!------- Block 3: Create Cell Array & Rest of Face Array --------------
      write(*,*)'Looping over all Cells: '
      write(*,*)
  !loop over all cells:
      do ic=1,INCells
        cfCount=4           !space counter for array index
           !loop over all node sets for the cell (4 PER CELL)
        do in=1,4
             !add 1 to cell face counter
          cfCount=cfCount+1
             !see if face with same four nodes already exsits
             !(temp variables used to shorten if statments)
          cn1=fm_t(nIndex(in,1),ic)
          cn2=fm_t(nIndex(in,2),ic)    !cell nodes making the current face
          cn3=fm_t(nIndex(in,3),ic)
            !add 1 to cell's node list (node 1 varies each time)
          cells(in,ic)=cn1
             !Loop over existing faces to see if current face is an OLD face
          old=-1
          j=0
          do while((j.lt.INFaces).and.(old.lt.0))
            j=j+1
            fn1=faces(1,j)
            fn2=faces(2,j)        !existing face nodes
            fn3=faces(3,j)
            if(((cn1.eq.fn1).or.(cn1.eq.fn2).or.(cn1.eq.fn3)).and.
     *              ((cn2.eq.fn1).or.(cn2.eq.fn2).or.(cn2.eq.fn3)).and.
     *               ((cn3.eq.fn1).or.(cn3.eq.fn2).or.(cn3.eq.fn3)))then
                !if true then its an old face, so don't add face, just
                !add cell # to face's cell list, also check if interior
                !boundary face:
```

```
                if(faces(6,j).eq.-1)then
                         !if ind:6 is -1, then Old interior
                   faces(6,j)=ic
                else
                         !otherwise it is an old interior boundary face
                   faces(5,j)=ic
                endif
                  !add face to cell list
                  cells(cfCount,ic)=j
                old=1
                  exit
              endif
            enddo        !end do while loop
              !if not an OLD face then create a NEW face
            if(old.lt.0)then
              !in this case, face is new, so create a new face
              INFaces=INFaces+1
                !set face's nodes
              faces(1,INFaces)=cn1
              faces(2,INFaces)=cn2
              faces(3,INFaces)=cn3
                !set face's first cell (second gets set when its old)
              faces(5,INFaces)=ic
                !if not a boundary face (which are done already) then it must
                    !be an interior face, so set faceAt to 1
              faces(4,INFaces)=1
                !now just add data to cell array
              cells(cfCount,ic)=INFaces
             endif
          enddo     !loop to in, node set loop
        enddo      !loop to ic, cell loop

        write(*,*)'Done looping over cells:'
        write(*,*)

!------------ Block 4: Tally neighbors --------------------
        write(*,*)'Tallying nearest neighbors:'
        write(*,*)
          !loop over all faces of all cells
        do ic=1,INCells
          do j=1,4
            !check if cell is in face's cell list position 1 or 2
            if(ic.eq.faces(5,cells(4+j,ic)))then
               !if cell is pos. 1 then cell in pos. 2 is a neighbor
              cells(8+j,ic)=faces(6,cells(4+j,ic))
            elseif(ic.eq.faces(6,cells(4+j,ic)))then
               !if in pos 2 then cell in pos 1 is a neighbor
               cells(8+j,ic)=faces(5,cells(4+j,ic))
            endif
          enddo   !loop over faces
        enddo   !loop over cells

        write(*,*)'ALL NEEDED DATA SET:'
        write(*,*)


!------ block 5: write out grid.3D file ------------------
```

```
!open output file
      str=''
      write(str,*)"Output File:"
      str(14:)=OutputFile(:)
      write(*,*)'Writing output file: ',trim(str)
      open(16,file=trim(OutputFile))

!write typical 3d header
      write(16,*)'< Nnodes >< Nfaces >< NCells >'
      write(16,*)INNodes, INFaces, INCells

!write Node data:
      write(16,*)' < Nx >< Ny >< Nz >< Nh >'
      do i =1,INNodes
        write(16,*)fm_p(1,i),fm_p(2,i),fm_p(3,i),fm_p(4,i)
      enddo

!write Face data:
      write(16,*)'< connectivity list >'
      do i = 1,INFaces
        write(16,*)faces(1,i),faces(2,i),faces(3,i),faces(4,i)
      enddo

!write Cells Data:
      write(16,*)'Cells - 4 nodes, 4 faces, 4 cells-neighbors'
      format='(1X,12I8)'
      do i=1,INCells
        write(16,format)cells(1:12,i)
      enddo

 !write virus stuff to end of file
      write(16,*)'<  nVir  >'
      write(16,*)nVir

!rc BE CAREFUL OF THIS BLOCK, coment out when nVir = 0
      if(nVir.gt.0)then
        write(16,*)'< XVir>< YVir >< ZVir >'
        write(16,*)xVir, yVir,zVir
      endif

      write(*,*)'--------DONE-------'

      end    !end program
```

COMSOL Surface Exporting: FemToSurf.m:

The following program is used to extract the surface triangulation from a COMSOL mesh structure. The program is written in Matlab and is run within the COMSOL/Matlab interface. The input to the program is a COMSOL .mesh data type which is exported directly from COMSOL. The output of the program is a surf.dat file which is formatted for immediate use with the fForm.f program.

```
%!m file to take in femlab mesh and create a surface for our meshGen
%!general values
numNodes=size(fem.mesh.p,2); %!number of nodes in FEM mesh
numFaces=size(fem.mesh.e,2); %!number of boundary faces in FEM mesh
h=-1.0;                  %!spacing number common to surf files
wArray1(1)=0;
wArray3(1:3)=0;
wArray4(1:4)=0;
%!mapping stuff
mapArray(1:numNodes)=0;
BNArray(1:numFaces)=0;
bnIndex=1;
for i=1:numFaces
   %!LOOP over nodes of face
   for j=1:3
      tempNum=fem.mesh.e(j,i);
      %!set mapping if new node
      if mapArray(tempNum)==0
         BNArray(bnIndex)=tempNum;
         mapArray(tempNum)=bnIndex;
         bnIndex=bnIndex+1;
      end
      faceNodes(j,i)=mapArray(tempNum);
   end
   %!make faceAt array, all 1's get changed to 99
   faceAt(i)=fem.mesh.e(10,i);
   if faceAt(i)==1
      faceAt(i)=99;
   end
end
%!start writing surfFile
%!leave a single zero for text input:
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
%!write <Nnodes>,<NBnodes>,<NFaces>
wArray3(1)=bnIndex-1;
wArray3(2)=bnIndex-1;
```

```
wArray3(3)=numFaces;
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray3','-ASCII','-append')
%!leave a single zero as a separator
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
for k=1:bnIndex-1
   %!write <Xn>,<Yn>,<Zn>,<h> for all BoundaryNodes
   wArray4(1)=fem.mesh.p(1,BNArray(k));
   wArray4(2)=fem.mesh.p(2,BNArray(k));
   wArray4(3)=fem.mesh.p(3,BNArray(k));
   wArray4(4)=h;
   save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray4','-ASCII','-append')
end
%!leave a single zero as a separator
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
for l=1:numFaces
   %!write <n1>,<n2>,<n3>,<at> for each face
   wArray4(1)=int16(faceNodes(1,l));
   wArray4(2)=int16(faceNodes(2,l));
   wArray4(3)=int16(faceNodes(3,l));
   wArray4(4)=int16(faceAt(l));
   save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray4','-ASCII','-append')
end
%!leave space for <Nvir>
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
%!leave space for number of viruses
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
% !leave space for <Xvir>,<Yvir>,<Zvir>
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
%!leave space for values
save('C:\Documents and Settings\ryanc\Desktop\newSurf.dat','wArray1','-ASCII','-append')
```

COMSOL Surface to U-GridGen: fForm.f:

The following program is used to reformat the surface triangulation from FemToSurf.m. The program is written in Fortran and is compiled using Intel's Fortran compiler. The input to the program is a surf.dat file which is the output of FemToSurf.f. The output of the program is a .srf file which is formatted for immediate use with the U-GridGen program.

```
!program that formats matlab output in surf.srf style
      program fileFormat

      real::a,b,Nnodes,NFaces,x,y,z,h
      real::n1,n2,n3,At
      integer::INnodes,INFaces,In1,In2,In3
      integer::IAt,nVir
      character(50)::InputFile,OutputFile
      character(100)::str

!-------------------------------------------------
!specify input/output file names
      InputFile='INPUTS/newSurf.dat'
      OutputFile='mmP_c1.srf'
!-------------------------------------------------

!careful when virus is need, must add location by hand at bottom
      nVir=0

!open input file
      str=''
      write(str,*)"Input File:"
      str(14:)=InputFile(:)
      write(*,*) trim(str)
      open(15,file=trim(InputFile),STATUS='OLD')
!open output file
      str=''
      write(str,*)"Output File:"
      str(14:)=OutputFile(:)
      write(*,*)trim(str)
      open(16,file=trim(OutputFile))
!read in header 0
      read(15,fmt=*)a
      write(*,*)'at header, a= ',a
     !write typical surf header
      write(16,*)'-surface triangulation-'
      write(16,*)'<Nnodes><Nbnodes><Nfaces>'
!read in Nnodes and Nfaces
      read(15,fmt=*)b,Nnodes,NFaces
      INnodes=int(Nnodes)
      INFaces=int(NFaces)
      write(*,*)'Nnodes: ',INnodes
      write(*,*)'NFaces: ',INFaces
     !write nodes, faces to file
      write(16,*)INnodes,INnodes,INFaces
!read in spacer 0
```

```
      read(15,fmt=*)a
      write(*,*)'at spacer, a= ',a
     !write header to file
      write(16,*)'< Xn >< Yn >< Zn >< h >Nodes'
!loop over Nnodes
      do i=1,INnodes
            read(15,fmt=*)x,y,z,h
            write(16,*)x,y,z,h
      enddo
!read in spacer 0
      read(15,fmt=*)a
      write(*,*)'at spacer, a= ',a
    !write next header to file
      write(16,*)'< N1 >< N2 >< N3 >< At > Faces'
!read in nodes and faceats for NFaces faces
      do i=1,INFaces
         read(15,fmt=*)n1,n2,n3,At
         In1=int(n1)
         In2=int(n2)
         In3=int(n3)
         IAt=int(At)
         write(16,*)In1,In2,In3,IAt
      enddo
!read in final spacer 0
      read(15,fmt=*)a
      write(*,*)'at end of file, a= ',a
    !write virus stuff to end of file
      write(16,*)'Nvir'
      write(16,*)nVir
      write(16,*)'XVir, YVir, ZVir'
!rc BE CAREFUL OF THIS BLOCK, coment out when nVir = 0
      if(nVir.gt.0)then
       write(16,*)'0.0 0.0 0.0'
      endif


      end
```