
[All ETDs from UAB](#)

[UAB Theses & Dissertations](#)

2008

Enabling the Direct Simulation (Monte Carlo) of Physical Vapor Deposition

Ramprasad Venkataraman
University of Alabama at Birmingham

Follow this and additional works at: <https://digitalcommons.library.uab.edu/etd-collection>

Recommended Citation

Venkataraman, Ramprasad, "Enabling the Direct Simulation (Monte Carlo) of Physical Vapor Deposition" (2008). *All ETDs from UAB*. 6650.
<https://digitalcommons.library.uab.edu/etd-collection/6650>

This content has been accepted for inclusion by an authorized administrator of the UAB Digital Commons, and is provided as a free open access item. All inquiries regarding this item or the UAB Digital Commons should be directed to the [UAB Libraries Office of Scholarly Communication](#).

ENABLING THE
DIRECT SIMULATION (MONTE CARLO)
OF
PHYSICAL VAPOR DEPOSITION

by

RAMPRASAD VENKATARAMAN

CHIH-HSIUNG CHENG, COMMITTEE CHAIR
RYOICHI KAWAI
ROY P. KOOMULLIL

A THESIS

Submitted to the Graduate Faculty of The University of Alabama at Birmingham,
in partial fulfillment of the requirements for the degree of
Master of Science

BIRMINGHAM, ALABAMA

2008

ENABLING THE
DIRECT SIMULATION (MONTE CARLO)
OF
PHYSICAL VAPOR DEPOSITION

RAMPRASAD VENKATARAMAN

MECHANICAL ENGINEERING

ABSTRACT

An existing, parallel, unstructured mesh-based Direct Simulation Monte Carlo (DSMC) code has been validated for scenarios akin to high-speed re-entry flows. Several test cases with experimental information from existing literature have been identified and used in the validation of the results. These include flows over a sphere and blunt cones at varying angles of attack, expanding flows through an orifice and flow over flat plates. The DSMC implementation has then been extended to verify its ability to handle other high Knudsen number regime flows. Work done, specifically targets enabling the simulation of vapor phase transport processes in Physical Vapor Deposition chambers by building a set of generic and extensible features into the implementation. Existing Physical Vapor Deposition simulation literature has been reviewed to understand the state of the art and to identify a prioritized list of new features that will benefit the current implementation. Enhancements revolve around the development of new (and more flexible) means of introducing and removing mass flux from the flow domains and include features like support for multiple inlets, arbitrary spatial variations in input flow properties, perfectly or partially sticking gas-surface interactions, the handling of backscatter fluxes in the simulation domain etc. This work has enabled this DSMC code to potentially handle vapor deposition flows of realistic complexity. Results from the simulations of a representative vapor deposition scenario are presented and compared against reference literature. The document also discusses and suggests avenues for further work in similar directions.

ACKNOWLEDGEMENTS

There is, of course, a huge bunch of people who've played a role in shaping my life and education, and I really intend these few words as my little way of acknowledging them all and saying thank you. A thank you for creating an environment that has nurtured and shaped my meagre knowledge, my not so meagre passions, and all my uninitiated interests. As must be common amongst the scores of other people who have attempted writing such acknowledgements sections, it is when I start that I realize the hopelessness of trying to condense all the thanks that are due into this small space. And hence, let me start with a sincere *'thank you'* to everyone whose lives have touched mine in selfless, beneficial ways.

I'd like to thank my parents for a zillion things small and big, for being the biggest single reason that I am who I am. For teaching me to question, to be curious and to be true to myself. My father, for being the reason that I am interested in science.

I'd like to thank all my teachers, every one of them my *gurus* in the whole sense of the word. More recently, my advisor Dr. Gary Cheng, from whom I have gained much more than just technical knowledge. I'd also like to thank him for his patience as I waded through the tedious phases of porting legacy implementations across different platforms.

There are also several people I'd like to thank in the context of this work. First, Yu-Young Lian, a visiting graduate student from the National Chiao-Tung University in Taiwan. I really appreciate his efforts in overcoming language barriers to help me get started with DSMC *and* PDSC. I'd also like to thank my wife Meenakshi, for helping me in numerous, not-so-little ways in the last weeks as I prepared my

document, and for really sharing in my ups and downs. My friend Prapanj, for all his time helping me get started with \LaTeX ; and Balaji, for proof-reading some chapters from my document.

TABLE OF CONTENTS

	<i>Page</i>
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xiii
LIST OF SYMBOLS	xv
1 Introduction	1
1.1 Statement of Objectives	1
1.2 Summary of Principal Contributions	1
1.2.1 Methodology and Code Capabilities	2
1.2.2 Implementation	2
1.2.3 Software "Good" Practices	3
1.3 Document Organization	3
I Background and Motivation	5
2 Physical Vapor Deposition	6
2.1 Vapor Deposition	6
2.2 Terminology	7
2.3 Applications	7
2.4 PVD Techniques	8
2.4.1 Evaporation	8
2.4.2 Sputtering	9
2.5 Process Variations in PVD	11
2.5.1 Ion beam assisted PVD	11
2.5.2 Plasma activation	11

	<i>Page</i>
2.5.3	Carrier gas assisted PVD 12
2.5.4	Reactive transport 12
2.5.5	Moving substrates 12
2.5.6	Collimation 12
2.6	The Need for PVD Simulations 13
2.7	Why Direct Simulation for PVD Flows 15
3	PVD Studies: Existing Literature 16
3.1	Simulation Methodologies 16
3.2	Reactor-scale Vapor Phase Studies 18
3.2.1	Vapor generation 19
3.2.2	Vapor collision properties 19
3.2.3	Sputter deposition 20
3.2.4	Directed Vapor Deposition 22
3.2.5	Ion Beam Assisted Deposition 23
3.2.6	Collimated vapor transport 23
3.3	Feature-scale Surface Evolution Studies 24
3.3.1	Influences on film morphology 24
3.3.2	Film porosity evolution 25
3.3.3	Continuum models for profile evolution 26
3.4	Integrated Multi-Scale Studies 27
3.5	A Note on Experimental Data 28
3.5.1	Chamber geometry 28
3.5.2	Target models 29
3.5.3	Metal vapor properties 29
3.5.4	In-situ flow property measurements 31
3.5.5	Deposition profiles 32
 II Methodology and Implementation 33	
4	Direct Simulation Monte Carlo: The Scheme 34
4.1	Approaches to describing fluids 34
4.2	The Need for Direct Simulation 35
4.3	Origin 36
4.4	Methodological Cornerstones 36
4.4.1	Dilute Gas 37
4.4.2	Motion-Collision decoupling 38
4.5	Phenomenological Collision Models 39
4.5.1	Elastic Collisions 40
4.5.2	Internal Energies and Inelastic Collisions 42
4.6	Probabilistic Particle Collisions 43

	<i>Page</i>
4.7 Gas - Surface Interactions	45
4.8 Sampling & Macroscopic Properties	46
4.9 Typical Algorithm	47
5 Parallel Direct Simulation Code: Highlights	50
5.1 Origins	50
5.2 Particle Models	50
5.3 Unstructured Grids	51
5.4 Multi-species Capability	51
5.5 Boundary Conditions	52
5.6 Variable Time Step (VTS) Scheme	54
5.7 Solution-Based Mesh Refinement	55
5.8 Parallel Procedure	56
5.9 Dynamic Load Balancing (DLB)	58
6 Enabling PVD Simulations: Contributions	60
6.1 Requirements	60
6.1.1 Capabilities of other codes in literature	60
6.1.2 Evaluating PDSC's abilities	62
6.1.3 Enhancements required	63
6.2 Multiple Inlet Conditions	63
6.3 Different Species Compositions at Inlets	67
6.4 Inlet Flow Profiles	70
6.5 Gas-Surface Interactions: Extended	76
6.6 Handling Backscatter: Constant mass flux inlets	79
6.6.1 Significance in Vapor Depositon Flows	79
6.6.2 Implementation	81
6.7 A Perspective on the Implemented Enhancements	84
III Results and Discussion	
	85
7 Validation for High Speed Flows	86
7.1 Corner FLOW over Perpendicular Flat Plates	86
7.2 Expanding Flow through an Orifice	90
7.3 Hypersonic Flow over a Sphere	93
7.4 Flow past a Blunt Cone	96
8 Some Results from PVD Simulations	105
8.1 Description	105
8.2 Simulation set up	108

	<i>Page</i>
8.3 Observations of the chamber flow	109
8.4 Wall and Substrate fluxes	113
8.5 Verifying the Implementation	115
8.6 Sensitivity Studies	117
8.6.1 Collision Cross-Section	118
8.6.2 Sticking Coefficient	120
9 Suggestions for Future Work	125
9.1 Methodology	125
9.2 Implementation and Validation	126
9.3 Computational and Numerical Considerations	127
LIST OF REFERENCES	129

LIST OF TABLES

<i>Table</i>		<i>Page</i>
7.1	A summary of some data for the test cases: A - Flow over perpendicular plates; B - Vacuum expansion through orifice; C - Flow over a sphere	87
7.2	A summary of relevant data for the test cases involving flow over blunt cone	97
7.3	Aerodynamic coefficients for flow past a blunt cone: Case 1	101
7.4	Aerodynamic coefficients for flow past a blunt cone: Case 2	101
7.5	Drag coefficients from the Parallel Direct Simulation Code (PDSC) compared against experiment and other simulations	102
8.1	Some relevant information from Balakrishnan et al [1]	107
8.2	Approximate computational size before and after mesh refinement	109
8.3	Refinement parameters	109
8.4	Predictions of absolute velocity compared against experiment	112
8.5	A comparison of Ti VHS model parameters used in the sensitivity study	118

LIST OF FIGURES

<i>Figure</i>	<i>Page</i>
2.1 An example of a Direct Evaporation PVD layout	9
2.2 Schematic layout of e-beam or laser beam PVD	10
4.1 Outline of typical steps in a DSMC procedure	48
5.1 Outline of the control flow in the PDSC - PAMR modules	57
6.1 Original inlet handling algorithm in the PDSC	64
6.2 Implementation of the multiple inlet conditions capability	66
6.3 Relevant portions of the algorithm handling inlet flow profiles	73
6.4 Overview of the logic encapsulated behind the flow profile handling interface	74
6.5 Contents of a profile definition file	76
6.6 Logic flow in handling gas-surface collisions	78
6.7 Modifications to handle sticking surfaces	78
7.1 Schematic of the perpendicular flat plates	88
7.2 Normalized density at $x/L = 0.4$	88
7.3 Normalized temperature at $x/L = 0.4$	89
7.4 Normalized density along the plane of symmetry at $x/L = 0.4$	89
7.5 Number density contours for expansion through an orifice	90
7.6 Normalized temperature contours for expansion through an orifice	91
7.7 Mach number contours for expansion through an orifice	91
7.8 Normalized density along the central axis of the orifice	92

<i>Figure</i>	<i>Page</i>
7.9 Schematic of the computational domain for flow over 10mm sphere	93
7.10 Number density contours for flow over 10mm sphere	94
7.11 Normalized temperature contours for flow over a 10mm sphere	94
7.12 Mach number contours for flow over a 10mm sphere	95
7.13 Stagnation line properties for flow over sphere	96
7.14 Blunt Cone (0° attack): Flow domain	97
7.15 Blunt Cone (Case 2, 0° attack): Normalized density contours	98
7.16 Blunt Cone (Case 2, 0° attack): Normalized temperature contours	98
7.17 Blunt Cone (Case 2, 10° attack): Normalized density contours	99
7.18 Blunt Cone (Case 2, 10° attack): Normalized temperature contours	99
7.19 Blunt Cone (Case 2, 20° attack): Normalized density contours	100
7.20 Blunt Cone (Case 2, 20° attack): Normalized temperature contours	100
7.21 Blunt Cone (Case 1, 0° attack): Surface heating rates	102
7.22 Blunt Cone (Case 2, 0° attack): Surface heating rates	102
7.23 Blunt Cone (Case 1, 10° attack): Surface heating rates	103
7.24 Blunt Cone (Case 2, 10° attack): Surface heating rates	103
7.25 Blunt Cone (Case 1, 20° attack): Surface heating rates	104
7.26 Blunt Cone (Case 2, 20° attack): Surface heating rates	104
8.1 Schematic of the Physical Vapor Deposition (PVD) chamber	106
8.2 Temperature and flux profiles for the vapor plume at the inlet	107
8.3 Initial grid used to start the DSMC simulations	108
8.4 Refined grid adapted based on the initial solution	108
8.5 Density contours	110
8.6 Temperature contours	111

<i>Figure</i>	<i>Page</i>
8.7 Absolute velocity contours	111
8.8 Normalized deposition profile	113
8.9 Fluxes to the side walls	114
8.10 Inlet flow profile generated by the PDSC preprocessor	115
8.11 Flux incident on the chamber surfaces	116
8.12 Flux reflected from the chamber surfaces	116
8.13 Sensitivity to VHS molecular diameter: Density contours. Left - $d = 4.0\text{\AA}$ (Balakrishnan et al); Right - $d = 5.844\text{\AA}$ (Fan et al)	119
8.14 Sensitivity to VHS molecular diameter: Normalized Backscatter Fluxes. Left - $d = 4.0\text{\AA}$ (Balakrishnan et al); Right - $d = 5.844\text{\AA}$ (Fan et al). Normalized against value at inlet center for base case ($d=4.0\text{\AA}$).	119
8.15 $D=4.0\text{\AA}$; Density contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9	121
8.16 $D=5.844\text{\AA}$; Density contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9	121
8.17 $D=4.0\text{\AA}$; Velocity contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9	122
8.18 $D=5.844\text{\AA}$; Velocity contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9	122
8.19 $D=4.0\text{\AA}$; Temperature contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9	123
8.20 $D=5.844\text{\AA}$; Temperature contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9	123

LIST OF ABBREVIATIONS

AR	Accept-Reject
BCT	Bi-atomic Collision Theory
CFD	Computational Fluid Dynamics
CVD	Chemical Vapor Deposition
DLB	Dynamic Load Balancing
DSMC	Direct Simulation Monte Carlo
EB-PVD	Electron-Beam Physical Vapor Deposition
IC	Integrated Circuit
KMC	Kinetic Monte Carlo
LJ	Lennard-Jones
MEMS	Micro-Electro Mechanical Systems
MPI	Message Passing Interface
MD	Molecular Dynamics
PAMR	Parallel Automatic Mesh Refinement
PDSC	Parallel Direct Simulation Code
PIC-MC	Particle-in-Cell Monte Carlo
PVD	Physical Vapor Deposition
PBS	Portable Batch System
SAR	Stop at Rise
SGE	Sun Grid Engine
SEM	Scanning Electron Microscopy
TBC	Thermal Barrier Coatings
TPMC	Test Particle Monte Carlo
VHS	Variable Hard Sphere

VSS	Variable Soft Sphere
VTs	Variable Time Step
YBCO	Yttrium Barium Copper Oxide

LIST OF SYMBOLS

Subscripts and Superscripts

For any property X, subscript and superscript notations have significance as follows:

X_{ref}	Reference value
X_{max}	Maximum value
X_{min}	Minimum value
X_0	Value in the macroscopic paradigm
\dot{X}	Value in the microscopic paradigm
X_T	Value at a temperature T
X_{tr}	Translational component
X_{rot}	Rotational component
X_v	Vibrational component
X'	Random component
\bar{X}	Average value
\vec{X}	X is a Vector
$X _f$	on a given face 'f' of the grid
$X _c$	in a given cell 'c' of the grid

Symbols

σ	Collision cross-section
c	Molecular speed
\bar{c}	Molecular velocity
d	Molecular diameter based on the cross-section

T	Temperature
m_r	Reduced mass
ω	Temperature index in the viscosity-temperature power law
$\Gamma()$	Gamma function
$erf()$	Error function
μ	Fluid viscosity
P	Pressure, a probability
P_{scaled}	A normalized probability
ρ	Density
u, v, w	Macroscopic velocity components
ε	Molecular energy
ζ	Degrees of freedom
F	Force
F_N	The number of real molecules represented by one <i>ambassador</i> , simulated particle
κ	Constant in the force function for the inverse power law model
r	Radius, Distance from centers of two molecules
η	Distance exponent in the force function for the inverse power law model
Δt	Time step
V	Volume
\dot{N}_i	Inward number flux
N	Number of molecules or <i>ambassador</i> particles
n	Number density
V_{mp}	Most probable molecular thermal speed
V_n	Velocity along normal to a surface or grid face
A	Area (usually of a face of the grid)
Kn	Knudsen number
λ	Mean free path

CHAPTER 1

INTRODUCTION

1.1 Statement of Objectives

The work done for this thesis has focused on inheriting, validating and extending an existing Direct Simulation Monte Carlo (DSMC) program - The Parallel Direct Simulation Code (PDSC). Consequently, the primary objectives involve:

- Verifying the capabilities of the code.
- Validating the results for high Mach and Knudsen number flow simulations akin to those found in upper-atmosphere / re-entry environments.
- Extending the program's feature-set to allow it to handle vapor deposition processes that approach real-world scenarios in complexity, and,
- Observing its performance in simulating the gas-phase phenomena in such processes.

1.2 Summary of Principal Contributions

Here is a brief list of the primary contributions of this thesis work. The list is not restricted to just methodological contributions but also mentions the outcomes of efforts directed at inheriting the code and ensuring that it has been setup satisfactorily in the local cluster environments.

1.2.1 Methodology and Code Capabilities

- **Multiple Inlet Conditions:** Implemented support for multiple inlets with different sets of flow properties.
- **Species Fractions at Inlets:** Modified the pre-processing and the core direct simulation components to make them accept species fractions as part of the inlet flow descriptions. Then, introduced the ability to handle different species fractions at each inlet.
- **Gas Composition for Domain Initialization:** Added capability to specify the gas composition for domain initialization independently from the compositions at the inlets. This feature prevents spurious vapor plume profiles and reduces the burn-in period.
- **Inlet property profiles:** Introduced support for multi-dimensional, arbitrarily oriented, free-form flow profile definitions that require no coding by the user.
- **Completely plastic gas-surface collisions:** Added ability to model solid surfaces as completely or partially sticking surfaces with a user-specified sticking coefficient. Set the stage for the adding support for multiple solid boundary conditions (akin to multiple inlet conditions).

1.2.2 Implementation

- **Re-usability:** Introduced reusable, encapsulated, object-based F90 implementations of some basic concepts (Cartesian vector, Cartesian plane, coordinate frame, grid point, macroscopic flow properties, flow profiles, interpolation algorithms) that can be used as the seed to modernize the rest of the code.
- **Flexibility:** Identified and removed instances of hard-coded parameters from both the PDSC and Parallel Automatic Mesh Refinement (PAMR).
- **User Input:** Streamlined the process of setting up program input to a set of intuitively-named files. Weeded out instances of duplicated user input.

- **Scripting:** Pruned driver script file that drove preprocessing, PDSC-PAMR interaction, successive refinement, post-processing and error logging.
- **Parallel Environment:** Converted the Portable Batch System (PBS) based PDSC-PAMR interfacing script to a Sun Grid Engine (SGE) based script. Weeded out parallel environment specific variables from the script.

1.2.3 Software "Good" Practices

- **Version Control:** Used Subversion to version control the code, enabling regression testing of each new feature in isolation.
- **Build Automation:** Automated build process by setting up GNU Makefiles.
- **Porting:** Resolved compiler & MPI library dependencies and portability issues on Coosa (a Linux cluster computing resource available on campus).
- **Documentation:** Initiated a code documentation exercise. Added comments to dozens of key subroutines.

1.3 Document Organization

The chapters in this thesis document have been divided into three parts to clarify their overarching objectives.

Part I, consisting of chapter 2 and chapter 3, provides the context for this work. chapter 2 briefly introduces PVD and discusses the relevant experimental setups and process phenomena. It also attempts to describe process variations popular in vapor deposition and wraps up by briefly making cases for the economic need for a simulation based approach to designing PVD apparatus, and, for the technical need for adopting a direct simulation approach to such studies. For someone acquainted with such thin film fabrication processes, the initial portions of this chapter may not provide new information. chapter 3 classifies and reviews available literature pertaining to PVD simulations. It also comments briefly on the availability of experimental data to

support studies akin to this work. Considerable effort has been invested in studying obtainable literature to prepare this review. Though the instances discussed are only a portion of what is available, they have been chosen with the belief that they best represent the developments and considerations that warrant some discussion.

Part II deals with the general simulation methodology, the specific implementation of this methodology, and the contributions of this work to that implementation. chapter 4 describes the DSMC methodology and attempts to present a very non-mathematical treatment of the assumptions and principles involved. For someone familiar with DSMC, the only novelty may be some of the phrases and quotations used. chapter 5 talks specifically about the particular implementation of the DSMC method that was used as the starting point for this work and attempts to explain some of the feature highlights of this code. chapter 6 discusses in some detail the principal contributions of this work. Again, an attempt has been made to minimize the use of equations in the explanation and to present only the extremely necessary parts of relevant algorithms.

The last portion of this document, Part III, presents results pertaining to the two major objectives of this work. chapter 7 presents results from an extensive validation exercise in which the ability of the code to handle high speed, high Knudsen number flows were studied. chapter 8 presents some of the results obtained from simulations of PVD scenarios using the enhanced version of the code. Finally, chapter 9 suggests possible future work in similar directions and concludes the discussions presented in this document.

PART I

BACKGROUND AND MOTIVATION

CHAPTER 2

PHYSICAL VAPOR DEPOSITION

2.1 Vapor Deposition

Vapor deposition, in general, is a fabrication technique that involves the condensation of element / compound vapor(s) to form solid deposits. The array of different vapor deposition techniques can be generally classified into two categories, viz. Physical Vapor Deposition (PVD) and Chemical Vapor Deposition (CVD). The significant difference between the two processes is the manner in which the binding energy is supplied. Binding energy, in this context, refers to the energy required to bind (deposit) the material onto the surface. In the case of PVD, film deposition on the required surface (substrate) is primarily a physical process like condensation. In contrast, CVD techniques depend on chemical reactions that occur on or near the substrate to bind the film material to the surface [2]. Another highlight of PVD is its dependence on a high-vacuum. As the transport processes involved in PVD are primarily physical, high-vacuum environments are a common requirement for most of the PVD methods [3]. This is not a stipulation for CVD methods, though.

The primary application of vapor deposition methods is in the manufacture of thin films. A common, alternative fabrication method is electroplating. While these methods are often viewed as complementary or competing techniques, each has its own niche. This is because of the range of possible film properties that can be obtained with each method [4, 5, 6].

2.2 Terminology

Before embarking on further details, it might be necessary to quickly explain a few relevant terms.

- **Target:** Refers to the source of material that is to be evaporated for later deposition. It can be either solid or liquid and can refer to more than one source (in the case of alloy deposition).
- **Substrate:** Refers to the surface on which the film is deposited. This may be planar or otherwise. From a deposition chamber design perspective, this is the region of maximum interest, as the prime concern will be to monitor the gas phase and surface phenomena at or near this surface.

2.3 Applications

PVD processes have now found a wide range of applications [3]. Some of these are listed here:

- the fabrication of high temperature superconducting films [7],
- coated fibers used in metal / ceramic matrix composites [8, 9],
- microelectronics and Integrated Circuit (IC) interconnects [10],
- coatings on medical implants like
 - stents [11, 12] and
 - dental prostheses [13, 14],
- thermal barrier coatings [15],
- optical coatings [3],
- magnetic & optical recording media [16],

- corrosion resistant coats on machinery like turbine blades [3]
- wear resistant coats on machine tools [17]
- inert coatings on high-pressure die casting machinery [18] etc.

2.4 PVD Techniques

PVD technology has advanced considerably over the years, and a wide variety of different configurations and techniques have been developed. This section will outline some of these major techniques, allowing me to place in context, the need for a simulation-based study of such deposition mechanisms. More detailed handling of PVD systems can be found in Holland [19] and, Kern and Vossen [20].

2.4.1 Evaporation

Evaporation deposition techniques effectively use heat to vaporize the atom and create a vapor phase that then travels from the target to the substrate. Different evaporation-deposition mechanisms differ mainly in the manner in which the target material is heated to produce the vapor phase.

2.4.1.1 *Direct heating*

This method is relatively the simplest, as it just involves vaporizing the component materials in a high vacuum to generate the vapor phase. The materials can be heated in crucibles or directly in the form of wires etc. The schematic in Figure 2.1 shows the approximate setup of such an apparatus. Disadvantages include finding the appropriate inert material for the containing crucibles that can withstand the high vaporization temperatures of the film material.

2.4.1.2 *Flash Evaporation*

The technique involves pre-heating the crucibles to a temperature higher than the boiling point of the materials and then gradually introducing the film material in

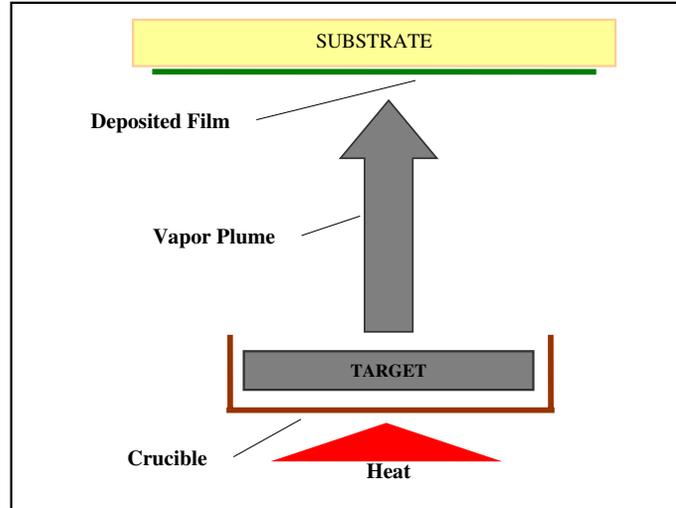


Figure 2.1 An example of a Direct Evaporation PVD layout

small amounts. The technique is a very crude method for depositing alloy materials and is not of much interest to precision film deposition industries.

2.4.1.3 Electron-beam evaporation

This approach involves shooting an electron beam at the target surface that then vaporizes the material in the immediate neighborhood owing to local heating. This generates a vapor phase without heating the crucible. Alloy films can be produced by rastering the electron beam across multiple sources so that the components mix in the vapor phase.

2.4.1.4 Pulsed Laser Deposition

A laser beam is used instead of an electron gun to vaporize the target material. The schematic in Figure 2.2 shows the general layout for both electron-beam and pulsed-laser depositions.

2.4.2 Sputtering

Sputtering techniques rely on bombarding the target surface with an ion stream to mechanically knock out the target atoms into the vapor phase. The sputtering phenomenon is because of the momentum exchange between the incident ions

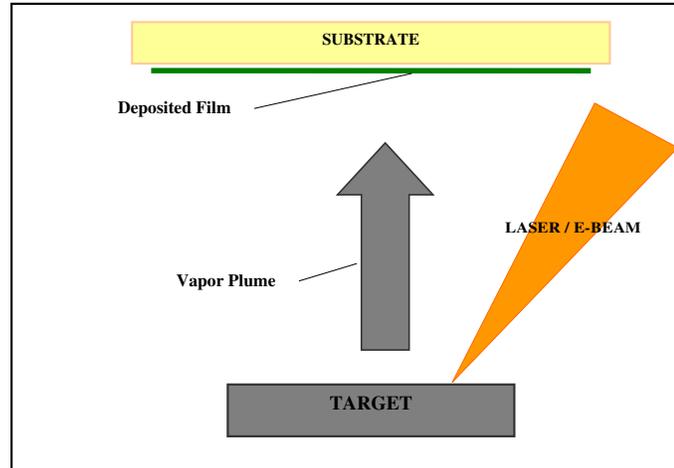


Figure 2.2 Schematic layout of e-beam or laser beam PVD

and the target atoms. Fundamental to the process is a source of ions that will sustain the process. Similar to evaporation techniques where the source of heat was the basis for classification, so too in sputtering, the source of ions gives rise to different methods. Some classifications for sputtering techniques follow.

2.4.2.1 Glow discharge sputtering processes

This utilizes the well known phenomenon of glow discharge that occurs due to the breakdown of a gas between two electrodes across which a potential is applied. Once ionized, the gas ions are accelerated towards the cathode. There, they impinge on the cathode to sputter out the cathode material. As a byproduct they also produce secondary electrons which are accelerated into the chamber ionizing more of the gas. Some of the types of glow discharge sputtering include

- Diode sputtering
- Bias sputtering
- Magnetron sputtering
- RF sputtering

2.4.2.2 Ion-beam sputtering processes

These are typically high-vacuum deposition methods where the ion source is removed from the system and the ion flux is introduced as a focused, collimated beam into the chamber. This beam then impinges on the target to sputter the material into the vapor phase. The deposition chamber can then be maintained at a higher vacuum without a loss of sputtering efficiency.

2.5 Process Variations in PVD

The different types of PVD are augmented by several variations in the process. These variations are mostly targeted at improving materials utilization or the properties of the deposited film, or film thickness & uniformity, substrate coverage etc. Often, these variations have significant impacts on the quality of the process and are prime necessitate detailed studies to improve an understanding of their impact.

2.5.1 Ion beam assisted PVD

Ion beam assisted deposition involves supplementing the vapors incident on the substrate with an ion flux. The ion flux is usually a focussed beam of chemically inert, energetic atoms (e.g. Ar^+) that strikes the substrate simultaneously with the vapor atoms. This technique is used to advantage in several scenarios to produce films with lesser residual stresses, a gradual composition gradient at the interface between the substrate and the film etc.

2.5.2 Plasma activation

Plasma activation refers to the process of activating (ionizing) the incident gas (vapor) atoms using a plasma in the vicinity of the substrate. This process affects the final film structure and properties by varying the activation levels of the incident vapor. A very similar technique is Ionized PVD [21] which uses a plasma to ionize

the vapor plume which is then accelerated towards the substrate by means of electric potentials.

2.5.3 Carrier gas assisted PVD

Carrier gas assisted vapor deposition, is a process where the vapor transport from the target to the substrate surface is aided by a gas jet that entrains the vapor atoms in its path and carries them to the substrate. Such an approach improves material utilization efficiency and also impacts mixing and the uniformity of the film composition. Published accounts use terms like Jet Vapor Deposition, Directed Vapor Deposition etc. [22, 23].

2.5.4 Reactive transport

A common approach to creating oxide coatings is to introduce oxygen into the reactor so that it can react with the vapor plume and deposit an oxide layer. This approach is commonly used to deposit superconducting films [24]. These kind of variations begin to blur the distinction between PVD and CVD. There are accounts of hybrid deposition methods in literature [25].

2.5.5 Moving substrates

A very common, industrial variation in the manufacturing process is to introduce mechanisms to translate or rotate the substrate [26]. This has applications in situations where the substrate is either a continuous surface that can be steadily fed into and removed from the deposition chamber (e.g. industrial, inline deposition equipment) [27] or in scenarios where coatings are desired on complex geometries (e.g. cylindrical fibers, biomedical implants etc.) [28].

2.5.6 Collimation

For certain applications, it is desired to introduce a collimator between the target and the substrate to control the angles of incidence of the vapor atoms on

the substrate. Collimation is usually achieved by means of a physical structure (e.g., a high aspect ratio honeycombed passage) that will filter out all vapor atoms with angles of incidence that are not near-normal. The atoms that pass through are ones which undergo near-collisionless transport from the target to the substrate. A typical application is for deposition on substrates with surface features like trenches (semiconductor wafers) which would otherwise experience void formations due to the formation of overhangs in the deposited film [29].

2.6 The Need for PVD Simulations

Vacuum deposition techniques have a niche for themselves in the core industrial fabrication processes. Over the last several years there has been a steady use of such methods. The applications typically demand tight process control to produce thin films meeting exacting standards. In this context, accurate design of deposition apparatus with the objective of achieving these standards is of prime importance. However, minimizing costs while retaining control over the process necessitates a good understanding of process variables that influence deposition thickness and uniformity. Such an understanding is key to reactor (deposition chamber) design.

section 2.4 and section 2.5 briefly talked about the different types of PVD, classified primarily by vapor phase generation mechanisms. The treatment has hopefully brought to attention the fact that a typical vapor deposition chamber plays host to several sensitive, interacting phenomena. Briefly discussed below are some influences worthy of attention.

- **High Vacuum:**Foremost among the complications in vapor deposition flows is the high vacuum environment. This one factor obfuscates the underlying flow phenomena and often produces flow features that defy intuitions based on continuum flow paradigms.

- **Metal vapors:** The majority of thin film applications involve the deposition of metal atoms. The presence of a metal vapor phase immediately entails high temperatures and steep temperature gradients, complicating interactions with the relatively cooler background gases.
- **Ion / Electron flows:** Vapor phase generation mechanisms that use electron / ion sputtering to produce the vapor phase also introduce additional challenges during the design phase. Designing chambers for such apparatus has to contend with the presence of charged particle flows and their interaction with the gas phase in the chamber. There have been several studies that focus on this aspect.
- **Chamber Geometry:** One major influence on the flow patterns in PVD is, obviously enough, the chamber geometry. Amongst the biggest aims of reactor design is to get this right to minimize the costs of operation and process control.

It must be quite obvious by now that the reactor design exercise calls for a better understanding and quantification of the extent to which each process variable can exert an influence. This in turn, requires a rigorous study of the chamber flow phenomena and deposition mechanisms. Minor variations in any of the process variables can cause significant changes in the flow patterns in the chamber, affecting the deposition rate and uniformity. More details on the complications involved in the design and study of such deposition chambers can be found in the exhaustive treatment of such technologies in Smith [30], Holland [19] and, Kern and Vossen [20]. Extensive experimental study is not viable, as generating high vacuums requires technology that is not cheap. Real-time flow measurements in these high vacuum, high temperature regimes also requires non-trivial tooling. A computational tool for the analysis and design of such equipment is, hence, of great value. Such a capability will provide a significantly cheaper and shorter design iteration cycle.

2.7 Why Direct Simulation for PVD Flows

The one unifying feature for most PVD techniques is their requirement of an evacuated (low-pressure) chamber [3]. This is because a low background pressure ensures that the sputtered (vaporized) atoms have high mean free paths, and thus form a molecular beam from the target to the substrate. This achieves higher deposition rates. High chamber pressures will result in more atomic collisions, a significantly higher role for diffusion processes, and hence, a drastic reduction in the incident vapor phase flux at the substrate. Apart from this, the heating effects on the target, due to evaporation or sputtering, generally cause steep temperature gradients. Additionally, in several real deposition environments, the ionic plasmas that participate in the vapor phase creation contribute heavily to thermal non-equilibrium effects.

Such factors result in the failure of continuum flow laws within the deposition chamber. The flows exhibit distinct non-continuum characteristics like thermal non-equilibrium, non-trivial vibrational energy components, low or counter-intuitive (from a continuum standpoint) diffusion behaviors etc. Knudsen number and breakdown parameter [31] calculations for deposition equipment will attest to this fact [24]. The numerical study of PVD flow phenomena, hence, requires non-continuum approaches like *direct simulation*. This should be more evident once the underlying assumptions and philosophies of the method are discussed in chapter 4.

CHAPTER 3

PVD STUDIES: EXISTING LITERATURE

3.1 Simulation Methodologies

The bulk of the numerical approaches used in vapor deposition simulations are particle based methodologies. As was mentioned in the previous chapter, this is because the applicability of continuum methods is severely limited. Listed below are a number of methodologies that have been used in non-continuum or micro-scale simulations.

- Direct Simulation Monte Carlo (DSMC) [32, 33]
- Test Particle Monte Carlo (TPMC) [34]
- Kinetic Monte Carlo (KMC) [35]
- Particle-in-Cell Monte Carlo (PIC-MC) [36]
- Molecular Dynamics (MD) [37]
- Bi-atomic Collision Theory (BCT)
- Lattice Boltzmann Method [38]

Several of these are based on the similar fundamental theories and differ in the computational approach or the algorithms involved in sampling the relevant results etc. Descriptions or comparisons of these methods for applications of interest will not aid or enhance any discussions in this document and, hence, have been left out. There exists literature that focuses on these topics [39, 40, 35]. However, it may be relevant to mention here that the advances in high performance computing technology and

resource availability have leveled the playing field and hence, have given an edge to the more complete or self-consistent simulation approaches.

The methods listed above trade varying degrees of fidelity and self-consistence against computational economy. Self-consistence here means the ability to incorporate different interacting components of the system to be able predict their collective influence on each other without having to resort to empirical models or fixed user-supplied conditions. For e.g, in a PVD system this might include the vapor phase, the background gas phase, the vapor generation mechanisms (ion beams etc.) and other such components of the PVD environment. A completely self-consistent simulation will incorporate each of these components into the simulation and account for their interactions. However as this may prove too complicated or computationally expensive, other approaches might decouple the interactions between these components and model them separately or include static empirical models for some of the components.

In the context of self-consistence, it may be pointed out that the truly self-consistent simulations will be those that span multiple length scales to capture all events of interest in the vapor deposition process. This will include *macro-scale* species transport, *micro-scale* film profile evolution, and *nano-scale* atomic interactions that provide first principles based estimates of model parameters to the macro and micro scales. It may be argued that a meso-scale module is necessary in bridging the gaps between the macroscopic length scales and micron-scale profile simulations. However, for the purposes of understanding, the length scales mentioned above host all the phenomena of interest in vapor deposition systems. Different approaches to PVD simulations adopt different levels of holism. A choice of simulation approach is also influenced by the objectives of the exercise. The methods, and available vapor deposition simulation literature using these methods can, hence, be classified, based on these criteria, into:

- Reactor scale gas phase transport and reaction simulations

- Feature scale surface evolution simulations
- Multi-scale models that combine both reactor-scale and feature-scale simulations

In the following sections is a review of some relevant literature. It discusses studies that are representative of the current status of vapor deposition simulation capabilities.

3.2 Reactor-scale Vapor Phase Studies

The primary goal in such analyses is to track the vapor phase flow (and reaction) mechanisms in the deposition chamber. Such a study offers deep insights into the transport mechanisms and the intricacies of the vapor phase phenomena that influence the deposition quantity and quality. Studies generally focus on the transport of the vapor atoms from the target to the substrate and also on the vapor flux at the substrate. Calculations usually predict the vapor flux, and the velocity & angular distributions of the vapor atoms arriving at the substrate. This information on its own provides some insight into the deposition rates and can be useful in the optimization of the process variables to achieve tighter control over deposition. There is also published work that has attempted to directly find empirical correlations (from simulation results), between the process variables and the film thickness at different points on the substrate. The hypotheses of such exercises seem validated by other studies that underline the influence of incident atom energies on film characteristics [41]. Apart from DSMC, several published works have used a Test Particle Monte Carlo (TPMC) scheme to obtain results. Specific instances of reviewed literature are given below.

3.2.1 Vapor generation

Chatain et al [42] have conducted experiments and used the TPMC method to study the properties of atomic beams that are produced by electron guns. They report experimental data of the mass flow rates, velocity and angular distributions of a Gadolinium beam and also estimate the same through simulation. The transport mechanism studied by them predicts the acceleration of the atoms owing to the conversion of the electronic energies into translational energy during inelastic collisions. To verify this model, they keep track of the electronic energies of the atoms and have also incorporated a method to account for this electronic-translational energy transfer in their simulations. In more recent work, they have repeated these calculations for a Copper-Gadolinium alloy [43].

3.2.2 Vapor collision properties

Fan et al [24] have studied the expansion of atomic plumes generated by Electron-Beam Physical Vapor Deposition (EB-PVD) and their deposition thickness on a planar substrate using DSMC. Studies include the single-species expansion of Yttrium and multi-species plume expansions (from multiple sources) and mixing of Yttrium, Barium and Copper. These elements are chosen because of their immediate applications in superconducting films. However, the oxygen required to deposit the Yttrium Barium Copper Oxide (YBCO) alloy is not considered in the simulation, and only non-reactive transport phenomena are studied.

The DSMC simulations in Fan et al [24] also account for the electronic energy of the atoms in the flow simulation. This is justified in an earlier paper by Balakrishnan et al [1], studying the ratio of translational to electronic energies for Ti atoms through a relevant temperature range. They find electronic energies to constitute a non-trivial portion of the particle energies and hence include them in their vapor deposition simulations. Being able to compute the atomic absorption spectra numerically also

provides them with a powerful means of validating their numerical studies against real-time gas-phase property observations in experiment.

The advantage with spectra measurements is that the observations are non-intrusive, and can be species-specific [44]. They also provide valuable glimpses at the gas-phase properties during the deposition process. In contrast, the bulk of the experimental studies report the deposition profiles as measured at the end of the deposition process. The availability of such in-process measurements of the gas phase properties will provide data that can be used to validate gas flow simulations in PVD chambers. This is a need that remains to be met by other experimental studies.

3.2.3 Sputter deposition

Vapor phase transport simulations of Ti atoms through a background gas (Argon) in two and three dimensional geometries have been performed by Kersch et al [45, 46]. The particle energy and angular distributions at the substrate have been found using both the DSMC method and the Test Multi-Particle Method (a variant of TPMC). Deposition rates have been calculated for background gas pressures in the 5 - 50 mTorr cm range. The authors have also compared multiple collision models and proposed a custom model that derives from the established Variable Hard Sphere (VHS) and Lennard-Jones (LJ) models.

The simulations by Kersch et al calculate the energy distributions of both the Ti and Ar atoms at the substrate. Chamber flow phenomena such as the “*sputtering wind effect*” are identified based on the simulations. Such phenomena stress the need for coupled background gas-vapor atom simulations like those performed using DSMC. In [45], Kersch et al also study the effect of a collimator on the deposition rates, and energy & incident angular distributions of the particles at the substrate. The collimator is modeled stochastically. Both of these simulation studies do not take into account the plasma used to sputter the target. Hence the energy and velocity distributions of the sputtered atoms at the target are modeled. A point of

interest in the work reported in [46] is the authors' choice of background gas models in the TPMC simulations. The background gas is modeled as a spatially varying Maxwellian distribution which is used to generate the molecular velocities whenever collisions between the background gas and the vapor atoms need to be performed. They observe that this choice results in deviations from the DSMC. This highlights the importance of recognizing the non-equilibrium flows dominant in the reactor and the costs of using approaches that sacrifice self-consistency.

The TPMC method, originally devised by Haviland & Lavin [34], is founded on the same assumptions and principles as the DSMC method. However, it is not a self-consistent scheme as it simulates only the vapor phase atoms, and is dependent on other sources for the ambient conditions in the deposition chamber. The underlying assumption in using the TPMC is that the vapor pressure of the Ti atoms is low enough to neglect the effects of Ti-Ti collisions. The influence of Ti-Ar collisions on the overall chamber flow phenomena is also assumed to be negligible. This is effectively an uncoupled approach to the simulation of the background gas and the vapor atoms. Once the background gas simulations are completed, a new simulation of just the Ti vapor atoms is started. This involves shooting a large number of 'test' vapor particles from the target to the substrate. The trajectories of each of the particles are influenced by collisions with the background gas atoms that are modeled stochastically, based on the results from the background gas simulations.

Minimizing statistical scatter in a simulation would require that a sufficient number of vapor atom trajectories be sampled. For low vapor phase concentrations of the film material this might sometimes result in an unmanageably large number of background gas atoms in the simulation. Unless other techniques are used to sidestep such issues associated with including trace species in the simulation, completely self-consistent simulations become expensive. One means of reducing simulation costs is at the expense of this self-consistency. The TPMC method goes down this road by assuming just a one-way influence between the background gas flow and the vapor

flow. Thus sufficient vapor atom trajectories can be sampled with the use of relatively lower computational resources.

3.2.4 Directed Vapor Deposition

Hass et al [28] have studied the Al atom transport through (on) a He carrier jet to estimate the coating uniformity on a thin, cylindrical substrate (fiber). Their experimental and computational studies are based on a novel technique known as Directed Vapor Deposition where a jet of gas is used to carry the metal vapor(s) onto the substrate [47]. This technique, a process that has been developed by Dr. Wadley and his group at the University of Virginia, has been shown to produce tightly controlled metal / metal alloy films. Literature from the group includes experimental studies [47, 48] and simulation results [49] that have been corroborated against these. Results published also include work done on multi-component alloy deposition and reactive deposition [50].

The simulation approach in the group's publications typically involve DSMC simulations of the carrier gas jet, followed by repeated sampling of vapor atom trajectories that use flow field properties generated in the first step. The random walk behavior that each individually modeled atom trajectory displays, collapses into a smoother, statistically averaged path after a sufficient number of samples. Again, such an approach decouples the vapor transport from the carrier gas transport, providing only the one-way influence of carrier gas flows on the vapor atoms. For low vapor phase concentrations, such approaches can be computationally beneficial.

Hass et al [28], however, uses a two-step DSMC simulation process. The first step considers only the carrier gas flow through the whole chamber flow domain. Generated flow field maps are used as boundary (inlet) conditions for a second, smaller and finer domain around the substrate. Vapor atoms are introduced into this second domain with the same flow velocities as the carrier gas. A two-species simulation is then conducted just in the vicinity of the substrate to obtain incident flow properties

of interest. The publication shows that the normalized vapor phase density compares well with experimental deposition profiles. The authors also repeat the experiments and simulations at different pressure ratios to observe the influence of the jet velocity (varied by altering the upstream pressure in the jet nozzle), on the deposition uniformity. Experimental and Computational Fluid Dynamics (CFD) studies for the same problem (PVD on fibers) have also been conducted by Hill [51].

3.2.5 Ion Beam Assisted Deposition

Youchison et al [52] have applied DSMC to model the plumes of Yttria-stabilized-Zirconia (YSZ) in an Ion-Beam Assisted Deposition apparatus to produce Thermal Barrier Coatings (TBC). They also investigate the effects of beam sharing and pivot rotation on the quality of the coat produced. The ion beam (Ar^+) and the oxygen (injected to maintain stoichiometry) are also included in the simulations.

3.2.6 Collimated vapor transport

Lin et al [53, 54] have used Monte Carlo simulations to study a collimated deposition process. They estimate the flux distributions of the vapor species at the collimator exit and also at the substrate surface. Both collision-less and collision-based models are used in a compare and contrast exercise. The dependence of the flux distribution at the collimator exit on the angles made with respect to the collimator axis is studied. The substrate flux distribution is also studied as a function of the position on the substrate. In Lin et al [53], The authors have also considered the effect of the different types of gas-surface interactions (specular and diffuse reflections, completely plastic collisions) on the flux through the collimator and the deposition profile. Based on all these studies, the authors attempt determine the influence of the parameters controlling the simulation model (mean free paths, surface interaction models, transport mechanisms) on the deposition profile.

3.3 Feature-scale Surface Evolution Studies

The primary objective of such simulations is to track the evolution of the profile of the deposited material on the substrate. Although tight process control might ensure that the vapor phase concentrations at the substrate surface are uniform throughout, this does not ensure a uniform coating all over the substrate. The final film profile depends on the microscopic topography of the substrate. For e.g., in the oft studied semiconductor wafer vapor deposition process [55, 56, 57], uniform vapor-phase concentration contours are not a guarantee of uniform deposition. Also, phenomena like diffusion or migration on the substrate surface might need to be accounted for. Going a step further, studies also target an understanding of the film morphology as a function of film profile evolution and hence, indirectly, as a function of the incident vapor properties.

The ultimate objective of the vapor deposition process is to produce thin films that meet the (usually exacting) requirements of the intended application. The ability to control the properties of the film by varying the process parameters is an important goal in vapor deposition innovation. This necessitates a quantified approach to analyzing the profile evolution on the substrate. Film growth simulations play a pivotal role in satisfying this need. These simulations normally yield the deposition profile on the substrate, which can then be directly compared with Scanning Electron Microscopy (SEM) data or other observations of the deposited profile. Their use in studying the fabrication of microscopic features is because of their inherent focus on surface phenomena at the required length scale.

3.3.1 Influences on film morphology

Hwang et al [58] have used Molecular Dynamics to study the film morphology in the Ionized PVD process.

The Ionized PVD process [21] is a variant of typical sputtering, where the vapor atoms get ionized while passing through a non-reactive, ionized plasma which occupies

the region between the target and the substrate. They can then be accelerated to different energies using electric potentials. Controlling the incidence energies of the particles influences film structure and morphology.

The authors in [58] have analyzed the influence of the composition of the incident ion beam on the trench filling mechanisms and the morphology of the film. The paper explains the models used to depict the trench, the incident source, the deposition process etc. and presents results of film profiles obtained by varying several process parameters like the incident ion energies, incident flux compositions etc. The simulations aid in a better understanding of the mechanisms involved in film deposition.

3.3.2 Film porosity evolution

Cho et al [59] use a KMC scheme to study TBC deposition on a rotating substrate. The objective of the work reported is to gain an insight into the evolution of porosity in the film during PVD.

Porosity plays an important role in determining the thermal conductivity and erosion resistance of deposited films, influencing one beneficially while the other, detrimentally. Typical influences on the porosity are the incident vapor properties, the energies of the incident (vapor / assisting gas) atoms, existing surface topography etc. Substrate rotation, a common fabrication technique to deposit films on complex geometries can also influence porosity by providing a “*sunrise / sunset*” [59] periodicity to vapor incidence.

The authors, in this study, take into account phenomena like surface diffusion and the shadowing effects of the surface topography while modeling the creation and evolution of pores on the surface. The dynamic relationship between the film topography and the angle of incidence of impinging particles is also accounted for. A detailed comparison of the film morphology from simulation and experiment is done. Terry [60], in related work, has also studied the influence of geometry on the film

microstructure while Schulz et al [61] have looked at the influence of vapor incidence patterns in more detail.

The authors in Cho et al [59] work run a KMC algorithm on a two dimensional hexagonal lattice structure that mimics the surface microstructure and observe the effects of a periodic incident flux on the profile evolution. The paper also describes the ballistic deposition model that has been employed to model the deposition process. The model basically allows the choice of different substrate geometries to begin the simulation and uses the lattice to specify possible sites that may be occupied by moving atoms. Incident atoms are modeled as impinging from a relatively great distance with varying angles of incidence. Vapor - Surface interactions are perfectly sticking, i.e. once an incident atom is captured onto the surface, it is restricted to move along the surface only. The model then simulates surface-bound evolution phenomena like particle diffusion, relocation etc. by imposing a few rules on the movement of atoms within the lattice (simulation domain). These rules are described in the paper.

3.3.3 Continuum models for profile evolution

O’Sullivan et al [57] have studied the development of the film topography using a hybrid Molecular Dynamics (MD) - Continuum (Level Set) method. Targeted applications are thin film fabrication for ICs. Their motivation stems from the objective of being able to predict minimum field and step coverage thicknesses using quick approximations. To this end, they have chosen the Level Set scheme which ignores microstructural information. The Level-Set method [62, 63] is a continuum formulation that is used in tracking the time evolution of a moving interface (front). A review of continuum models for film evolution by Cale et al Cale98, offers more information about such approaches. In [57], results are compared with experimental data for both collimated and un-collimated deposition.

3.4 Integrated Multi-Scale Studies

Multi-Scale models attempt to do in one integrated simulation process, what is done by the previous two types of simulations. They generally incorporate a simulation scheme that analyzes the vapor phase transport processes and uses the generated results as input for a feature-scale simulation model. The vapor phase model takes as inputs, the various process parameters that are relevant to the simulation. They then produce predictions of the vapor and background gas flowfields, usually including detailed profiles in the vicinity of the surfaces of interest. These results are then fed as an input to the feature scale model. This model uses the provided vapor phase fluxes as input values and simulates the evolution of the surface profile which is ultimately the result of interest. Thus the influence of the process variables on the film profiles can be predicted directly. This is a very powerful tool in the design optimization of reactors, as it defines a quantified relationship between macroscopic parameters and the microscopic details of interest.

It should be noted that there may exist a need for a meso-scale method to fill the gap between feature-scale and reactor-scale simulations. The following quote from Cale et al [64] should help clarify this.

“The two-scale approach (coupling reactor and feature scales directly) makes the tacit assumption that the reactor scale simulator is able to compute conditions above a particular feature, though its mesh is necessarily coarse compared to the micron length scale of a feature scale simulator.”

Coronell et al [65] have developed such an integrated model for Sputter deposition. The reactor scale model is based on the TPMC method which is used to generate the required flux distributions. This is then fed to the feature scale model, also based on a Monte Carlo scheme, that neglects the presence of the gas phase completely. This model then predicts the surface profile. Practical applications to reactor design are also demonstrated by applying the integrated model to a comparison of long-throw PVD and collimated PVD. Another sample problem that is attacked is

the identification of a target erosion profile that optimizes the cross-wafer deposition uniformity. Tuning the vapor generation mechanism to produce such profiles can result in greater film uniformity and spread.

Arunachalam [66] et al have extended this integrated model to Ionized PVD. They have also included MD simulations as another module that feeds data to the feature scale model. Ion-surface interaction data are estimated by this MD module. A similar study of ion-surface interactions has also been reported by Coronell et al [67].

3.5 A Note on Experimental Data

This chapter has focused on reviewing different simulation approaches documented in literature. This has been with the intention of evaluating the state of the art in PVD simulations and to identify a prioritized list of the capabilities of different simulation software that enable them to tackle the reported PVD simulations. The work done in this thesis is completely centered on the development of simulation capabilities to tackle realistic vapor deposition problems. However, experimental data can provide valuable insight and evidence for validating numerical models of a physical phenomena. Most of the literature reviewed in the previous sections also contains experimental data or references to sources of these data. However, this section collates such information and presents a brief note on the available experimental data that can support numerical model development. In an attempt to be concise, elaborate discussions are eschewed in favor of pithy comments on the different aspects of PVD studies with pointers to other material.

3.5.1 Chamber geometry

Chamber geometry is one of the primary macro-scale influences on the micro-scale phenomena that occur at the substrate. They directly and significantly affect the vapor fluxes and hence the deposition profiles at the substrate. Most of the

simulation literature provides descriptions of the chamber geometry that was used as the computational domain for the vapor phase simulations.

3.5.2 Target models

Sputtering and EB-PVD processes typically erode the target surface in a non-uniform fashion. This results in a non-planar target surface. This source surface topography might affect the angular distribution of vaporized atoms, hence, influencing the film uniformity and thickness. It might hence be beneficial to model this target topography. Balakrishnan et al [1] provide simulation evidence to the contrary after performing a sensitivity study to observe the effect of a non-planar target on the chamber flow profiles. Target profiles are not available in other reviewed literature.

It is more common to include an energy and angular distribution profile for the vaporized atoms in the simulation. This data has been reported in several reviewed studies of vapor transport [1, 46].

3.5.3 Metal vapor properties

Numerical modeling of the non-continuum transport processes in PVD requires the definition of particle (atom) interaction models that can accurately capture the diverse interactions occurring in the gas phase. Typical, industrial PVD processes involve multiple metal vapor species, background or carrier gas atoms and ionized plasmas in the gas phase between the target and the substrate. Modeling atomic interactions between all of the above presents considerable challenges and is critical to high fidelity predictions. Crucial to the definition and validation of such models is the availability of appropriate experimental data on which the parameters defining the particle models can be based.

Particle models will be discussed in greater detail in section 4.5. The most common particle model used in DSMC simulations is the VHS. There have been PVD studies that use the simpler Hard sphere model [68]. The VHS usually requires information that can allow it to reproduce the viscosity of the species being modeled at

the required temperature ranges. This will typically boil down to the specification of the collision cross-section at a reference temperature and the temperature dependence of the viscosity.

Such data for metal vapor atoms in the vaporization temperature ranges is not readily available in literature. This can be due to numerous reasons. Measuring metal vapor viscosities at the temperature ranges typical of PVD metal vapors is a challenging task. Also, experimental studies of PVD seem to focus more on the film profile and properties than on measuring the properties of the vapor plumes / atoms. In the course of this work, a need was felt for more comprehensive resources on high temperature metal vapor property data, which could be used to define appropriate model parameters.

A quick search of available literature shows that there exist numerous studies of alkali metal vapor viscosity and several reports of experiments to measure the same [69, 70, 71, 72, 73]. There have also been non-empirical approaches to computing the transport properties of alkali metal vapors [74, 75] and attempts to fine-tune existing atomic models (LJ) to match experimental data [76]. However, the very same searches also turn up a dearth of similar data for other metal vapors.

Fan et al [24] voice similar opinions and propose a method to calculate the collision cross-section for metal vapor atoms in the high temperature ranges that are associated with metal vaporization energies. The proposed approach starts by using coefficients of viscosity for alkali metal vapors obtained theoretically by Fialho et al [77]. These values are then plugged into Equation 3.1 to calculate the constants κ and η which are constants featuring in the inverse power law model [32].

$$\mu = \frac{5 \left(\frac{kmT}{\pi} \right)^{1/2} \left(\frac{2kT}{\kappa} \right)^{2/(\eta-1)}}{8A_2(\eta)\Gamma \left[4 - \frac{2}{(\eta-1)} \right]} \quad [3.1]$$

More on the inverse power law model can be found in section 4.5. The authors make the assumption that metals occupying the same row in the periodic table obey

the same power law in the vapor phase. Hence, the values of κ and η obtained for an alkali metal can be used to calculate the collision cross section and the temperature dependence of the viscosity for other metals in the same row. The calculations are according to the following equations:

$$d_{\text{ref}} = \left(\frac{15 \left(\frac{mkT_{\text{ref}}}{\pi} \right)^{0.5}}{2(5 - 2\omega)(7 - 2\omega)\mu_{\text{ref}}} \right)^{0.5} \quad [3.2]$$

$$\omega = \frac{1}{2} + \left(\frac{2}{\eta} - 1 \right) \quad [3.3]$$

Computed values for the coefficient of viscosity of Sodium and Caesium are compared with experimental data and seem to agree well. Such a study assumes significance in light of the facts discussed above. The calculations in [24] have been used to set up the collision model in [28] and will be used in this work too, where required.

3.5.4 In-situ flow property measurements

Similar to the dearth of adequate data to define accurate collision models, there is also a need for more literature on real time flow property measurements in vapor deposition chambers. Such data can provide a fillip to the process of validating numerical models for the gas phase transport mechanisms. On a lighter vein, it may be observed that it is the very intractability of such measurements that creates the need for and spurs the development of simulation methods.

The bulk of the literature reviewed in this chapter restrict themselves to the study of flow properties obtained from simulations to propose or confirm hypotheses about the flow mechanisms within the reactor. Balakrishnan et al [1] and Fan et al [24] have computed atomic absorption spectra for comparison with in-situ experimental observations. Chatain et al [42] also measure atomic spectra to corroborate their computations.

3.5.5 Deposition profiles

Deposition profiles are amongst the most commonly reported experimental observations from PVD studies as they are relatively easier to obtain. Almost all the simulation studies reviewed in the previous sections report deposition profiles or thicknesses on the substrate. Fan et al [24] report deposition thicknesses measured using a profilometer. Some scatter is seen in the measurements. Balakrishnan et al [1] report similar profile thickness measurements for Titanium deposition. In [45], the authors compare experimental deposition profiles with simulation results for different reactor geometries. Hass et al [28] also report deposition profile and compare it with relative (normalized) vapor fluxes very close to the substrate to demonstrate a good correlation between incident fluxes and profile thickness.

PART II

METHODOLOGY AND IMPLEMENTATION

CHAPTER 4

DIRECT SIMULATION MONTE CARLO: THE SCHEME

4.1 Approaches to describing fluids

Fluids can be described in two paradigms.

The more common of the two is the continuum formulation where the fluid is treated as a continuous medium and the flow description is in terms of the commonly encountered properties of density, temperature, velocity etc. A complete description of the fluid at any instant in time is given by specifying these flow properties at each point in the physical space of interest. This approach treats the fluid at a macroscopic level, which will become clearer when contrasted with the microscopic description.

The microscopic description of a fluid treats the fluid as a collection of molecules. The fluid system can then be quantified in terms of molecular properties like mass, momentum, energy etc. A complete description of the fluid at a given time, however, involves the specification of the positions and velocities of all the molecules in the physical space of interest. Fluid ‘*flow*’ phenomena result naturally from the evolution of these molecular properties through the interactions of the molecules with each other and with the system boundaries in the form of collisions. This is the foundation of Boltzmann’s equation.

When this individual molecular behavior is viewed collectively at length and time scales which are appropriately larger, then one begins to observe the collective system behavior amenable to description in the macroscopic framework. Mathematically, this is expressed by equating macroscopic flow descriptors to averages or summations of relevant microscopic properties of the system.

4.2 The Need for Direct Simulation

Computational Fluid Dynamics, primarily based on the Navier-Stokes equations, is a continuum formulation for describing fluids and is capable of handling a wide variety of fluid flow problems. The continuum assumption is valid when the *characteristic length*, which is the length scale of interest in the flow, is much larger than the *mean free path*, which is the average spatial separation between successive collisions undergone by a molecule in the flow. This stipulation is expressed in terms of the dimensionless Knudsen number ($Kn = \lambda/L$) by stating that the continuum assumption is valid only in the low Knudsen number regimes. When this condition is not met, the transport terms in the Navier-Stokes equations fail, leading to a *breakdown* of the formulation.

This scenario can occur in some extreme flow environments. Some of these extreme flow environments are now of considerable practical importance, as they are encountered in several areas. These include upper stage rocket engines, in-space propulsion systems, planetary entry, micro-flows like those involving Micro-Electro Mechanical Systems (MEMS), low pressure vapor deposition systems, etc. Accurate modeling of such flow phenomena is possible only by treating the fluid at a microscopic level as an aggregation of particles. Such approaches sidestep the assumption that the fluid is a continuous medium, and are based on the more fundamental laws of physics that are still valid in these flow regimes.

The mathematically rigorous approach to simulating such problems would be to solve the Boltzmann equation. However, this is a very computationally challenging task. DSMC is an approach that seeks to model a fluid system at the microscopic level, physically rather than mathematically. It is a more computationally feasible approach and is, today, the method of choice for analyzing such flows.

4.3 Origin

The DSMC methodology was propounded by Bird [78, 32] in the sixties as a computational method for studying rarefied gas flows. The primary applications intended were in the aerospace industry for the analysis of high speed, rarefied flows. The scheme adopts a physical, rather than mathematical, description of the fluid and is based on the fundamental laws of physics. Hence, the name ‘*Direct Simulation*’ as it directly simulates the physical system rather than modeling it mathematically. The scheme depends heavily on stochastic methods to conduct tractable numerical observations of fluid systems. The methodology extracts meaningful data from the simulations by ‘sampling’ a large number of repetitive trials which is the essence of a ‘*Monte Carlo*’ methodology.

“Just as gaming in the casinos of Monte Carlo rests upon the statistical properties of random events, MC-based methods are stochastic techniques. Simulations based on the MC method provide means of approximating solutions to mathematical problems through statistical sampling experiments on a computer.” (Battaile et al [35])

4.4 Methodological Cornerstones

The DSMC scheme captures the features of any flow by modeling a representative sample of the whole molecular population of the flow in a microscopic paradigm. This effectively means that each modeled particle is an ‘*ambassador*’ molecule that represents a (usually) large number of real molecules. Repetitive measurements on such a smaller sample, finally add up to reflect the properties of the actual population. Simulations are conducted directly in ‘*real*’ space and time without the need for numerical transformations of any kind. The DSMC method makes the same assumptions of molecular chaos and a dilute gas that are made by the Boltzmann equation [32]. In fact, it has been shown that the scheme converges to the Boltzmann formulation as the number of particles become large and the cell size and the time step

shrink to infinitesimal values [79, 80]. Some of the assumptions of the DSMC method are listed below to provide a context for the following text.

4.4.1 Dilute Gas

A dilute gas is one in which the total molecular volume of all the molecules in the system is insignificantly small when compared to the volume of the system in which these molecules exist. Such a condition implies that the mean molecular spacing which is of the order of $n^{-\frac{1}{3}}$ is much greater than the effective molecular diameter. The implications of such an assumption are

- The trajectory of a single molecule, for most part, lies outside the influence of the inter-molecular forces exerted by neighboring molecules.

Though quantum molecular models and popular collision models like the LJ (6-12) present an asymptotically decaying force function, it is practical and acceptable to include a cutoff distance outside of which, two molecules do not exert any influence on each other. Such an approach is adopted in both DSMC and MD approaches. In the case of DSMC, the dilute gas assumption enables the use of simpler collision models that represent only the short range inter-molecular repulsive forces. Molecular models will be discussed in greater detail in section 4.5.

- Binary collisions are the overwhelmingly probable intermolecular interaction event.

When the gaseous system is so sparse as to contain molecules in free motion for a significant portion of their trajectories, the probability of many-body collisions becomes vanishingly small. This implication of a dilute gas is the basis for another important assumption of the DSMC method that lets it handle large gaseous systems. This will be explained in the subsection 4.4.2. This is also another primary difference between DSMC and MD methods. The original DSMC

formulation neglects many-body interactions, whereas MD schemes inherently handle multi-body interactions.

Though the DSMC methodology was developed on top of this assumption, there are instances where the equivalent of many-body interactions (e.g. reactive collisions) can be incorporated into the simulations. Recent advances have also shown that the method is capable of simulating denser systems if the fraction of volume occupied by the molecules in the system is accounted for [81].

4.4.2 Motion-Collision decoupling

One of the fundamental assumptions of DSMC is that the motion of the particles and their interactions can be decoupled over time spans that are much less than the mean collision time. This assumption plays a very important role in making the scheme more computationally feasible for larger problems. This is another of the fundamental differences between the DSMC and MD approaches.

MD takes into account the detailed dynamics of particle collisions. It uses particle collision models in a classical collision mechanics framework to compute position and velocity changes directly due to collision events. This deterministic approach to handling collisions necessitates a much smaller time step in numerical simulations, currently making it viable for a different class of problems. The MD scheme also calculates collision properties through the evaluation of the inter-molecular interaction potentials. It is thus inherently capable of handling many-body interactions.

The class of problems (relatively denser systems) that are tackled by MD are also amenable to such a treatment. In the case of a dilute gas, inter-molecular interactions are relatively rarer events. If the simulation scheme were to perform deterministic collisions only when trajectories intersected, the simulation system would have to include so many molecules and be so expensive as to currently be only of theoretical interest.

DSMC, on the other hand, decouples the particle trajectories from the actual collision events. Collisions do not occur as a result of intersecting trajectories. Instead, collisions are actually performed by randomly selecting particle pairs that are in each other's vicinity and performing a *probabilistic collision*, where a probability (weight) is assigned to the collision based on the relative velocities of the particles involved in the collision. Post-collision molecular properties are computed from stochastic models. These considerations will be explained in section 4.5 and section 4.6.

4.5 Phenomenological Collision Models

“A model that reproduces the overall effects in a gas flow of some physical feature of real molecules, but does not explicitly incorporate that feature, is said to be a *phenomenological model*.” (Ch. 2 in Bird [32])

The success of the DSMC approach depends on the accurate modeling of the molecular interactions in a gas. The overall macroscopic properties describing the flow are determined by such interactions. Hence, particle collision models, which provide a computationally feasible approach to handling the molecular collision mechanics, play a large role in determining whether physically significant features in the flow are captured by the simulation.

A relevant initial source of such models was the treatment of classical collision mechanics in kinetic theory [82, 83]. Most collision models in use today can be traced back to these origins. The models in their original classical forms focused on the scattering of participant molecules due to the collision. This is evident in their treatment of post-collision properties based on the impact parameter. Effectively, this results in a faithful computation of the post-collision trajectories, when supplied with pre-collision velocities and trajectories. However, as DSMC decouples the collision process from the actual molecular motion to improve computational tractability, collision events cannot be provided with relative particle trajectories and impact parameters. Hence, collision models must provide a means of empirically estimating or

assigning scattering angles to each collision event in such a manner as to capture the overall flow physics accurately. Such models are known as phenomenological models.

“The phenomenological approach is to create the simplest possible mathematical model of a process at the molecular level, that reproduces the physically significant effects of that process on the gas flow.” (Bird [84])

Some of the particle models used in DSMC and other molecular descriptions of physical system are as listed below. They can primarily be classified base on whether the model describes just the repulsive forces that dominate short-range interactions between molecules, or if the relatively weaker medium-range attractions are also incorporated.

- Repulsive potential models
 - Hard sphere model [82]
 - Inverse power law model [82]
 - Variable Hard Sphere (VHS) model [85]
 - Variable Soft Sphere (VSS) model [86, 87]

- Attractive-repulsive potential models
 - Square well potential
 - Sutherland model [82]
 - Lennard-Jones (LJ) potential model [88, 89, 90]
 - Generalized Hard Sphere model [91, 92]

4.5.1 Elastic Collisions

The simplest of these is the hard sphere model that describes the particle as possessing a fixed diameter and scatters the particles uniformly in all directions. The interaction force between two hard spheres comes into play when the distance between

the centers is equal to half the sum of their diameters. A more generalized form of the simple hard sphere model is the inverse power law model that incorporates a continuous variation of the repulsive power based on the distance between the centers of the particles. The force function for the inverse power law model can be expressed as,

$$F = \frac{\kappa}{r^\eta} \quad [4.1]$$

The primary advantage of the inverse power law model is a scattering angle that is dependent on the impact parameter in the collision. This is a marked improvement over the hard sphere which produces isotropic scattering. However, the interaction potential extends to infinity and this poses complications in computing the collision cross-sections. The necessity to include cutoff distances in the potential and impact parameters in the model make computations more involved.

It has been shown that the scattering angles in molecular collisions is only the smaller of the two influences on the translation of collision events into flow property evolution [84], thereby justifying the continued use of relatively simple models for the generation of scattering angles. The other influence is the dependence of the collision cross-section (or effective molecular diameter) on the translational energy involved in the collision. Studies comparing the performance of different molecular models and the impact of the model parameters on macroscopic properties like the coefficient of viscosity demonstrated that simple models like the hard sphere do not capture the temperature dependence of the fluid viscosity accurately.

This led to the introduction of the Variable Hard Sphere model [85]. The VHS model extends the hard sphere model to include a variable collision cross-section that is dependant on the relative particle velocities. This dependence leads to a correct variation of the fluid viscosity with temperature. The VHS model hence improves the fidelity of simulations by modeling the cross-section dependence on temperature akin to the inverse power law model, while retaining the simple isotropic scattering model of the hard sphere. Relevant expressions are listed here for reference and can

be found explained in greater detail in Bird [85, 32].

$$\sigma_T = \sigma_{T,ref} \left(\frac{c_{r,ref}}{c_r} \right)^{\frac{4}{\eta-1}} \quad [4.2]$$

$$\frac{\sigma_T}{\pi d_{ref}^2} = \frac{(2kT_{ref}/m_r c_r^2)^{\omega-0.5}}{\Gamma(2-\omega)} \quad [4.3]$$

$$d_{ref} = \left(\frac{15(mkT_{ref}/\pi)^{0.5}}{2(5-2\omega)(7-2\omega)\mu_{ref}} \right)^{0.5} \quad [4.4]$$

One of the limitations of the VHS model observed in simulations was that it captured only the viscosity coefficients correctly but not the diffusion coefficients in flows involving gas mixtures. This need was addressed by the VSS model proposed by Koura and Matsumoto [86]. This model uses the same cross-section variation as the VHS but includes a non-isotropic scattering model.

Described briefly so far, were some particle models commonly used in simulations or analytical treatments. Models that include an attractive component in the interaction potential have simply been listed above and have not been discussed. This is because of their limited applicability to DSMC methods due to the assumptions made by the method. An alternative to particle models is to directly use experimental and theoretical data on the collision cross sections across the relevant temperature range. Bird [32] cites work published by Chatwani [93] in this context. However, information about continued work in this direction does not seem available.

4.5.2 Internal Energies and Inelastic Collisions

Discussed so far are particle models that describe elastic collision processes. Such models capture the flow physics of monatomic gases quite well. However, these models do not function as well for diatomic or polyatomic gases. This is because these models do not account for the participation of the internal energy modes in the energy exchange during inter-molecular interactions. Essentially, internal energy modes of polyatomic gases result in inelastic collisions where there is an exchange in energy between the interacting molecules as well as between the different energy modes of a molecule. Models proposed as extensions of the classical collisional models to account

for internal energy redistributions in collision processes were too involved to be used in simulations and generally caused prohibitive computational costs. This is because they attempted to create physical analogies of the internal energy states of a molecule (e.g. harmonic oscillators for modeling vibrational energies etc.) and this made them unsuitable for use in a DSMC context. The almost universally adopted solution to this issue, in the DSMC approach, is to model the internal energy redistribution during inelastic collisions phenomenologically.

Larsen and Borgnakke [94] proposed a scheme where a fraction of the collisions were treated as inelastic while the rest were handled as elastic collisions. For the inelastic events, they proposed that the post-collision internal energies be sampled from known equilibrium distributions that depended on a ‘*notional temperature*’ [84] that was associated with the collision energy. Though this approach is not based on any physical reality, it has worked well and is used widely in real gas flow simulations. Other models for handling rotational and vibrational energies are discussed in Bird [84] and notable among them is a scheme put forward by Lord [95] for inelastic collisions and Bergemann et al specifically for vibrational modes [96].

4.6 Probabilistic Particle Collisions

As explained earlier, the occurrence of inter-molecular interaction events is not predicted deterministically in DSMC. Instead collisions are handled probabilistically in order to reduce the size of the simulation system necessary for sampling the sufficient number of interaction events required for estimating macroscopic properties.

The probability that two molecules in a system collide, is proportional to their collision cross-section (σ_T) and their relative velocities (c_r). Within a single cell ‘ c ’ in the simulation volume, the probability of collision between a pair of simulated particles is given by Equation 4.5. The only terms in this expression that depend on the pair of participant particles are the cross section and the relative velocity. The maximum collision probability can hence be written as Equation 4.6. For reasons of

computational efficiency that are detailed in chapter 11 of Bird [32], Equation 4.5 is scaled against this expression for the maximum to yield the expression that is used to calculate the probability of collision between two simulated molecules within a cell. (Equation 4.7). In this equation, the subscripts p and q refer to different groups (species) of simulated particles. In other words, the collision probability for any two particles is scaled against the $(\sigma_{TCr})_{max}$ across all the particles in groups that the considered particles hail from.

$$P = F_N \sigma_{TCr} \Delta t / V|_c \quad [4.5]$$

$$P_{max} = \frac{F_N (\sigma_{TCr})_{max} \Delta T}{V|_c} \quad [4.6]$$

$$P_{scale} = \frac{(\sigma_{TCr})}{\{(\sigma_{TCr})_{max}\}_{pq}} \quad [4.7]$$

A derivation of the mean collision rate per unit volume in a system starting from considerations of the collision probability distribution over the relative velocity space can be found in Bird [32] and is given by Equation 4.8. This expression when adapted for a cell in the simulation system yields an expression for the number of collisions over a time δt as shown in Equation 4.9. The decision to represent n^2 as $N\bar{N}\dot{F}_N$ is also for computational reasons and is explained in Bird [32].

$$N_c = \frac{1}{2} n v = \frac{1}{2} n^2 \overline{\sigma_{TCr}} \quad [4.8]$$

$$\frac{1}{2} \frac{N_p \bar{N}_q F_N \{(\sigma_{TCr})_{max}\}_{pq} \Delta t}{V_C} \quad [4.9]$$

The process of conducting interaction events starts with the calculation of the collision rate in a cell of the simulation volume. Once the number of collisions is determined, particles from the same cell are randomly picked for possible collision. This ensures that only particles close to each other are selected. The probability (or weight) associated with the collision is computed based on the relative velocity of the two particles as expressed in Equation 4.7.

4.7 Gas - Surface Interactions

One of the assumptions of DSMC is that in gas-surface interactions, the gas is modeled at the molecular / atomic level whereas the surface is treated from a macroscopic perspective as a solid surface and not as a collection of bound atoms (as molecular dynamics does). This is because the length and time scales of interest in a DSMC simulation are of the order of the mean free path and the mean collision time respectively of the gas that is being studied. These are orders of magnitude higher than would be necessary to study the atomic level effects of collisions on the surface atoms.

Similar to inter-molecular collisions, gas-surface interaction events are also modeled phenomenologically. Boltzmann in his work proposed two types of solid surfaces; one that would allow the incident molecule to be reflected elastically with just a reversal of appropriate vectors, and another that would perform an inelastic collision. The elastic version is essentially identical to a mirror boundary and is not a good model of real surfaces. This version called *specular reflection* fails to account for the influence of collisions on the the surface properties. Diffuse reflection on the other hand allows for the collision energy to be redistributed between the surface and the molecule. The molecule is reflected off the surface with an energy that would cause the velocity distribution of the reflected particles to conform to a Maxwellian at some temperature (T_r) that is between the temperature of the incident gas (T_i) and the temperature of the surface (T_s). This process is known as thermal accommodation and diffuse reflection allows for varying degrees of thermal accommodation. Such a model is more realistic because one can view the surface as having a rough topography at the microscopic level causing incident particles to suffer multiple collisions on the surface increasing the residence time on the surface. An increased residence time will cause the particle to *relax* towards the surface temperature, and, when reflected will result in a reflected temperature as described above.

Surface interaction models that are currently in use, model this process of thermal accommodation by reflecting a fraction of the incident particles diffusely while the rest are reflected specularly. This approach, though not based on a physical analogue, captures the physical effects of thermal accommodation quite realistically. It should also be noted that for most real gases and most surfaces, complete thermal accommodation can be assumed. However, there are some scenarios when this assumption might have to be reexamined for validity. In Bird [32], there is a mention of some scenarios that may expose the insufficiency of existing surface interaction models (for diffuse reflection). They are listed here as these factors may have a bearing on the flows being considered as part of this work.

- High vacuum leading to outgassed solid surfaces
- High incident particle energies
- Highly polished surfaces

4.8 Sampling & Macroscopic Properties

After the particle motion and collision procedures are repeated multiple times, the particle properties are sampled. The macroscopic flow properties in each cell are assumed to be constant over the cell volume and are sampled from the microscopic properties of the particle within the cell. This is the purpose for the grid - to enable the assignment of particle properties to an immediate surrounding region of the physical domain. The macroscopic properties, including density, velocities and temperatures, are calculated according to the following equations [2.1].

$$\rho = nm \quad [4.10]$$

$$c_0 = \bar{c} = \bar{c}_0 + \bar{c}' \quad [4.11]$$

$$\frac{3}{2}kT_{tr} = \frac{1}{2}m(\bar{u}^2 + \bar{v}^2 + \bar{w}^2) \quad [4.12]$$

$$T_{rot} = \frac{2}{k} \frac{\varepsilon_{rot}}{\zeta_r} \quad [4.13]$$

$$T_v = \frac{2 \varepsilon_v}{k \zeta_v} \quad [4.14]$$

$$T_{tot} = \frac{3T_{tr} + \zeta_v T_v + \zeta_v T_v}{3 + \zeta_{rot} + \zeta_v} \quad [4.15]$$

n and m are the number density and mass of one molecule, respectively. \bar{c} , \bar{c}_o , and \bar{c}' are the total velocity, mean velocity, and random velocity, respectively. In addition, T_{tr} , T_{rot} , T_v and T_{tot} are translational, rotational, vibrational and total temperatures, respectively. ε_{rot} and ε_v are the rotational and vibrational energies, respectively. ζ_{rot} and ζ_v are the number of degrees of freedom of rotation and vibration respectively.

It may be beneficial to use two different cells for the selection of particles and the averaging of the results. A smaller sub-cell may be used to ensure that the selected particles are very near each other, whereas a larger cell may be used to average the properties over a larger number of particles than will be present in sub-cells.

4.9 Typical Algorithm

The general procedure of the DSMC method is described here and also depicted as a flowchart in Figure 4.1. The central procedure can be described in 4 major steps - Tracking, Indexing, Collision and Sampling of the particles in the domain.

- **Initialization:** A direct simulation exercise typically starts by initializing the domain with gas particles at a specified state. The number of particles in each cell is calculated based on initial density specifications and the cell volume. These initial particles are assigned velocities based on equilibrium Maxwellian distributions calculated from the macroscopic velocity and temperature.
- **Particle Tracking:** The particles in the domain are now moved based on their velocities. The task, now, is to locate each particle with respect to the grid. This is important as the sampling procedure calculates local macroscopic properties based on the particles present in each cell. Hence this stage primarily consists of

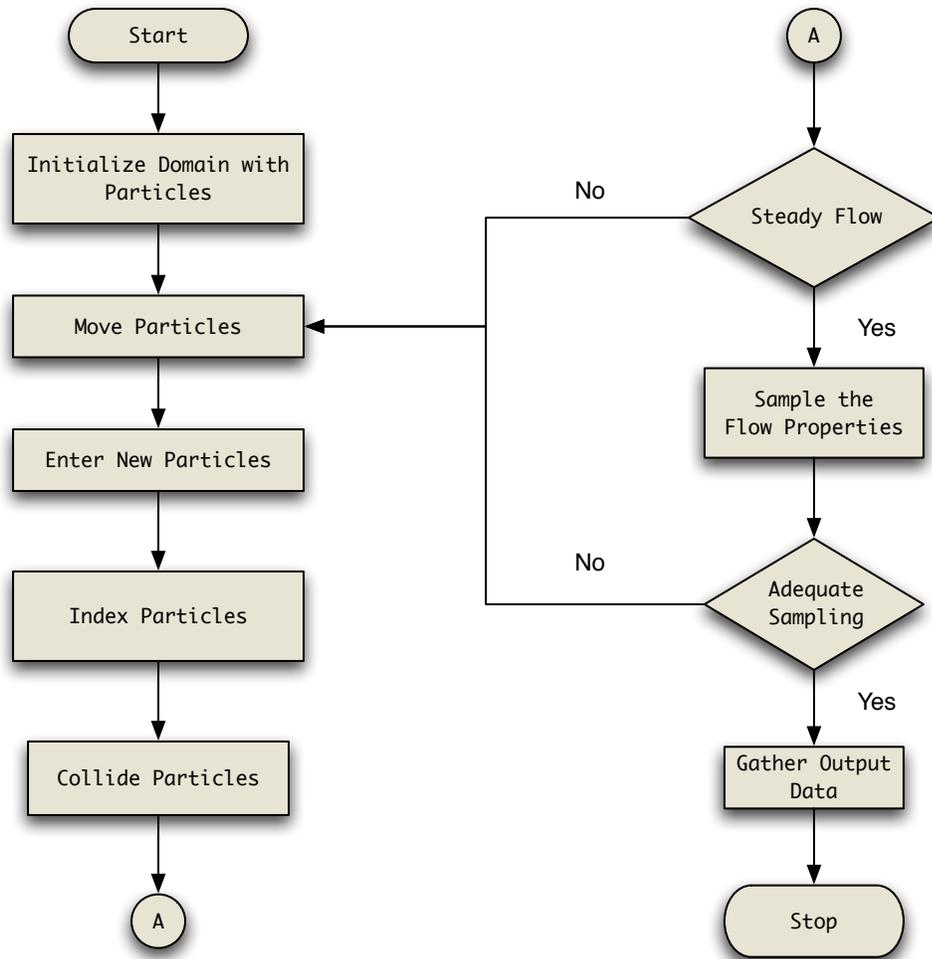


Figure 4.1 Outline of typical steps in a DSMC procedure

determining 'if', 'when' and 'where' a particle crosses the face of the cell under consideration. Particle interactions with solid surfaces are also handled at this time.

- **Indexing:** The location of the particles w.r.t the cell, after movement, is important information for particle collisions. The lists of particles and cells they belong to are reordered according to the location of each particle. Such a renumbering aids in the identification of in-cell collision partners during the collision stage.

- **Collisions:** Collision rates are calculated and the the appropriate number of collisions between randomly selected partners from the same cell are performed.
- **Sampling:** The macroscopic flow properties are obtained as summations or averages of the microscopic description that is being simulated.

CHAPTER 5

PARALLEL DIRECT SIMULATION CODE: HIGHLIGHTS

5.1 Origins

The Parallel Direct Simulation Code (PDSC) was developed at the Aerothermal and Plasma Physics Laboratory (formerly the Multi-scale Science and Technology Lab) at the National Chiao Tung University, Taiwan by Dr. J.S. Wu and his team [97]. It is a parallel, three-dimensional, unstructured grid, multi-species direct simulation code with several features that allow it to tackle non-equilibrium flow simulation problems of realistic complexity and size. Some highlights include a parallelized algorithm, a variable time step scheme, solution based mesh refinement and dynamic load balancing. This chapter describes in some detail these capabilities of the code.

5.2 Particle Models

The PDSC has implemented a VHS collision model for the molecules (particles) in a simulation. Particle translational and rotational energies are modeled. The particle model is defined by setting parameters specific to each species. These include the mass of a molecule, cross-section at a reference temperature, viscosity index, the number of rotational degrees of freedom and options to use a constant or polynomial rotational relaxation rate. The species can also be grouped into collision sampling groups but this becomes relevant only when multiple species are present and the species of interest are present only in trace quantities.

5.3 Unstructured Grids

The developed DSMC code uses an unstructured mesh to track particles in three dimensional physical space [98]. Though the structured grid topologies offers computational simplicity and lesser memory requirements, the unstructured grid system has been chosen because it can efficiently handle complex geometries and facilitates mesh adaptation. The increase in computational complexity and bookkeeping requirements are an acceptable trade-off for this advantage. The demonstration code in Bird [32] uses structured grids. However, there are several direct simulation codes in existence that have successfully used unstructured grids to analyze rarefied gas flows. Wilmoth et al [99] have compared structured and unstructured grid methodologies for DSMC computations and found the results comparable. The code can currently read grids exported in the '.lrg' format from Hypermesh, the grid generation tool available as part of the Altair Hyperworks Suite [100].

5.4 Multi-species Capability

The PDSC has been developed with support for multi-species problems built-in. Hence, the core algorithm and all supporting routines can handle multi-species flows. Setting up such a simulation involves specifying the collision model parameters for each species. Species fractions must also be specified and will be applied to the domain when the simulation is initialized and also at each inflow boundary in the simulation domain. This means that the gas mixture that is introduced into the domain in any manner is always of the same composition.

This capability was sufficient for modeling non-reacting, external, re-entry flows where it could be safely assumed that the flow composition was fixed. However, this rendered the code incapable of handling situations where the mixture composition is not the same at all sources. Typically, for any vapor deposition process there are at least two flux sources that introduce or remove molecules from the domain. One

is the *source* which introduces vapor atoms that are of interest and which are meant for deposition onto the target. The other is either the background gas inlet or outlet which maintains a near-vacuum background gas concentration and provides a neutral (non-reacting) environment for the sputtered molecules. The actual number and roles of the inlets can vary according to the reactor design. However, the presence of at least two such inlets with dissimilar functions is standard. Modeling such vapor deposition domains would thus require the capability to handle different species fractions at each inlet. This is a feature that has been added as part of this work and is described in greater detail in chapter 6.

5.5 Boundary Conditions

The PDSC provides the typical types of computational domain boundaries. The available boundary types are either provided to model physical flow boundaries in the simulation (like solid walls, inflow boundaries etc.) or to provide computational savings (like periodic boundaries, planes of symmetry). The PDSC is capable of handling

- Inflow-outflow boundaries
- Solid boundaries
- Vacuum boundaries
- Symmetry / Mirror boundaries
- Periodic boundaries

The computational domain can contain multiple surfaces of each boundary type. Users can specify boundary properties for each type of boundary (solid walls, flux boundaries etc.) and these are applied universally to all boundary surfaces that belong to that boundary type. The pre-processing code identifies each boundary surface using an ID associated with it during the grid generation phase. The user then

has to provide information mapping each boundary ID to a boundary type. The pre-processing code will then use this information to supply boundary properties for all the faces lying on the boundaries of the computational domain.

The code treats a solid boundary as one which participates in a collision process with incident particles.

Mirror boundaries (planes of symmetry) are modeled as solid boundaries on which incident particles undergo completely elastic collisions. Hence incident particles are merely reflected (mirrored) back into the domain. The intention is to mimic the presence of a replica of the computational domain on the other side of the face. Collision data from such mirror boundaries are not sampled or used in any further calculations. Real solid surfaces on the other hand are modeled as surfaces that undergo a mixture of specular and diffuse reflections.

Early versions of the PDSC required that the user write code that would be used in the pre-processing stage to identify the boundary that each cell face lay on and set boundary properties accordingly. Much of the results in chapter 7 were generated using such a version of the code. A limitation with this approach was that each boundary had to be represented as a set of equations that would uniquely associate a boundary ID with any point on the boundary of the domain. This usually turned out to be tedious. It was also a challenge in specifying arbitrarily oriented or irregularly shaped inlets.

A later improvement that was implemented by Dr. Wu's group allowed users to set the boundary ID in the mesh generation tool and then specify the boundary type using this ID. The preprocessing code contained automation that would identify the boundary ID for each cell face on a domain boundary specified in the input grid file and assign the face with a boundary type specified by the user. This was a very helpful addition as it greatly reduced the effort required by the user, especially when the simulation involved inlet surfaces that did not have simple orientations or shapes.

5.6 Variable Time Step (VTS) Scheme

The number of simulated particles per cell, $N|_c$, is given by Equation 5.1. The particle weight ($F_N|_c$) is the ratio of the number of real particles to the number of simulated particles in that cell. For a properly refined mesh the cell size (representative length) is inversely proportional to the number density, as mesh refinement will be triggered in regions where the number density is greater than a particular freestream parameter. With the time step and particle weights considered to be constant, we can obtain the proportionality given in Equation 5.2.

$$N|_c = \text{frac}nV|_c F_N|_c \quad [5.1]$$

$$N|_c \propto nV \propto n^{-2} \quad [5.2]$$

This leads to the situation where the number of simulated particles is higher in the low density regions and vice versa. This distribution of simulated particles is precisely the opposite of what is required, causing unnecessary sampling and computations for the low density regions, and inaccurate results in the regions of higher density. A VTS scheme [101, 102] has been implemented to overcome this difficulty, and to ensure a more uniform (simulated) particle distribution.

The underlying concept of the VTS scheme is to enforce flux conservation across the cell boundaries. The local time step size is scaled in accordance with the local cell size to ensure the particles have appropriate residence time within a cell for sampling. Flux is conserved by changing the particle weighting accordingly. Thus

$$\frac{F_N}{\Delta t}|_{c1} = \frac{F_N}{\Delta t}|_{c2} \quad [5.3]$$

where the subscripts c1 & c2 refer to different cells in the domain. The ratio of the particle weight to the time step is a constant for all the cells in the domain. The VTS procedure involves, identifying the cell with the minimum volume in the whole simulation domain, and calculating the reference time step size using this cell volume. The reference time step size is then used to determine the time steps for

the remaining cells. The particle weights are now calculated to conform to above mentioned equation. A more thorough handling of the VTS scheme can be found in Tseng [101].

5.7 Solution-Based Mesh Refinement

DSMC simulations may encounter steep gradients in the density and other flow properties within the domain-of-flow of interest. Typical examples include flows that involve shocks or rapid expansion to vacuums / near-vacuums. Increasing the grid resolution for the whole domain so that these features are captured satisfactorily will result in an unmanageably large grid. The optimal solution to this problem is to refine the grid only in the regions that encounter steep gradients needing greater resolution. The details of the flow gradients can be known only in hindsight after the simulation has been run to obtain at least an approximate solution. It is thus essential to have a mesh system that can adapt to the flow field to accurately capture the gradients, and thus be able to resolve the flow features with high fidelity. The latest version of the code has such a module called the PAMR that performs this process [103, 101].

The first step in the process is to identify the regions that have to be refined. Two parameters are employed to evaluate this. The first parameter employed is the cell-local Knudsen number, which can be defined as

$$Kn = \lambda/L|_c \tag{5.4}$$

If the local Knudsen number is well below 1.0, it indicates that the cell size is significantly larger than the mean free path, thus flagging it down for refinement. The local Knudsen number of the child cells is automatically higher, as the reference length $L|_c$ is smaller. This limits the number of levels of successive refinement that can be performed in a given region.

The second parameter that limits unnecessary refinement in the freestream region is a *freestream parameter*. The freestream parameter is a ratio of the local number density to the freestream number density. Refinement is carried out only if the number density in the cell is higher than a threshold value set by the freestream parameter. The choice of such a parameter depends on the nature of the flow and influences the tractability of the simulation. A conservative choice will limit the region which undergoes refinement whereas the other end of the spectrum presents scenarios where indiscriminate refinement causes an explosion in the number of cells.

Once the appropriate regions have been identified, the refinement module [104] starts up and refines the cells in the required regions. Details of the refinement procedure can be found in Tseng [101]. The refined mesh is then repartitioned and the direct simulation module is restarted with the new grid.

5.8 Parallel Procedure

Parallelization as a technique for making computationally demanding algorithms feasible needs no introduction. The earlier version of the direct simulation code, *DSMC-3D*, used a parallel algorithm only for the direct simulation component of the process [98]. This, though a very important feature on its own, could not be exploited fully as the adaptive mesh refinement was a serial module. Any simulation would thus have to be shuttled between parallel and serial platforms before a final solution could be obtained.

The latest version of the code, PDSC, is a completely parallelized code which handles both the central direct simulation computations as well as the supporting operations like domain repartitioning and adaptive mesh refinement [103] in parallel. This has eliminated manual handling of these procedures during the periodically occurring mesh refinement stages, thereby speeding up the whole simulation exercise considerably. A rough outline of the process can be found below.

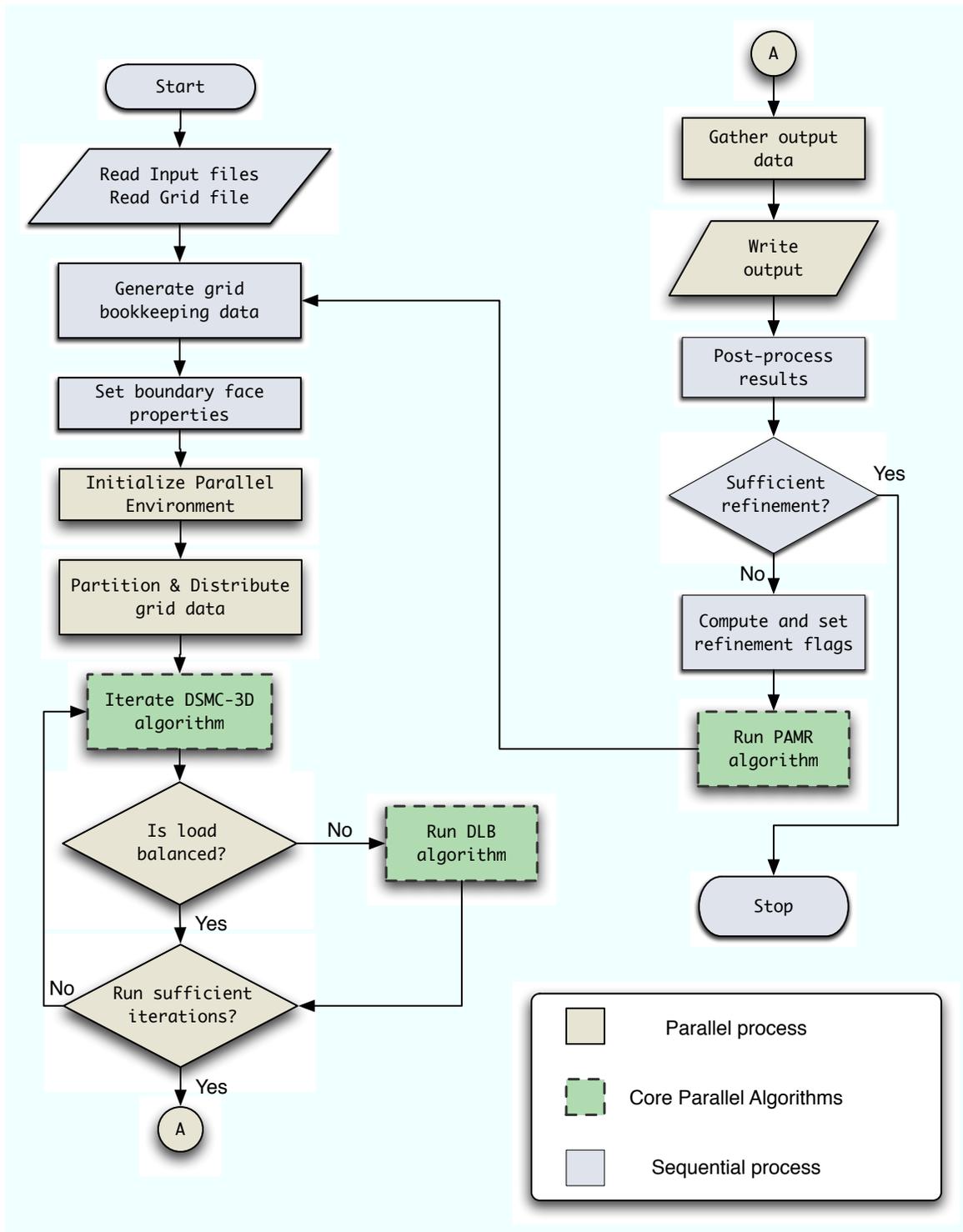


Figure 5.1 Outline of the control flow in the PDSC - PAMR modules

The main processes in the DSMC method consist of four steps: *tracking*, *indexing*, *collision* and *sampling* of the particles in the flow field. These processes are applied sequentially to all the particles in the domain. However, the flow domain can be directly decomposed into several portions to make this a data parallel application. This is achieved through a graph-partitioning algorithm [104] that slices up the computational grid and allocates different portions of the computational flow domain to the available CPUs. Each CPU then executes the main DSMC algorithm in serial over all its particles.

Inter-CPU communication primarily includes the exchange of data concerning the particles that cross the partition boundaries of the domain. This is achieved through the transfer of packed arrays that hold relevant information about the particles among all the CPUs. This process also serves as a synchronization pulse between each time step. The (computational) load on each CPU is monitored and significant degradation from the optimum triggers a load balancing algorithm that redistributes the workload on the fly. This is described in greater detail in section 5.9.

Once this procedure has been carried on for a sufficient length of time, the macroscopic averages are computed. The solution thus obtained is used as the basis for determining the need for mesh refinement. The PAMR [103], once started, refines the mesh locally based on the solutions obtained in that region. The refined mesh is then repartitioned and the direct simulation code is restarted with the new mesh. This process is optionally iterated multiple times to obtain an accurate solution to the problem at hand. A flowchart describing the parallel algorithm is presented in Figure 5.1.

5.9 Dynamic Load Balancing (DLB)

For parallel/distributed computing, the performance can be maximized only if the load on all the collaborating processes is balanced. The DSMC method being particle based, the workload on each CPU is directly proportional to the number of

particles it is handling. This depends on the flow gradients and the physical size of each partition. The task is further compounded because the number of particles is bound to change drastically during the initial burn-in stage of the simulation. This truly dynamic nature of the DSMC simulations makes it imperative to include algorithms for the dynamic balancing of the computing loads to improve performance. In the present DSMC code, the task of balancing the load is to re-divide the computational domain based on the number of particles being handled by each CPU.

The domain can be decomposed dynamically, by either invoking the algorithm at fixed time intervals, or by predetermining the optimal repartitioning interval, or by evaluating the need for a domain repartitioning. It has been shown that the first method leads to a poor realization of the advantages of parallelization [105]. The second method necessitates an analysis to predetermine the optimal time interval. Hence the final approach of evaluating the need for repartitioning in real time is adopted.

This approach uses a technique called Stop at Rise (SAR) to invoke the necessary algorithms. The method monitors the processor idle time, at the end of each time step. Assuming that the load is perfectly balanced, the processors should synchronize at the end of each time step. It then evaluates the deviation from this state of perfect balance using a degradation function, using it as the basis to call for a repartitioning. The procedure involved in dynamic balancing is detailed in Wu and Tseng [106]. An evaluation of the performance of the degradation function can be found in Nicol [107].

CHAPTER 6

ENABLING PVD SIMULATIONS: CONTRIBUTIONS

6.1 Requirements

6.1.1 Capabilities of other codes in literature

In chapter 3, published accounts of PVD simulations were reviewed. The focus there was to take stock of the approaches and achievements of such studies in order to gain an understanding of the typical objectives of vapor deposition simulations. Specifically, in section 3.2, we discussed literature that presented numerical and / or experimental observations of the gas phase phenomena in vapor deposition scenarios. In this section, the focus will be to recap the capabilities of the programs used in such studies in order to illuminate the motivation for the enhancements to the PDSC. that will enable it to handle PVD gas phase simulations. Relevant to the discussion here are publications by Balakrishnan et al [1], Fan et al [24], Kersch et al [46] and Hass et al [28].

Both [1] and [24] are publications that use the same direct simulation code (MONACO) [108]. Relevant to the problems tackled is the capability to handle multiple species entering the flow domain from multiple flux sources. The setup in [1] also necessitates the specification of a variation of flow properties across an inlet, corresponding to a target flux profile. The backscatter of sputtered vapor atoms back into the inflow surface is also accounted and corrected for, to satisfy a constant mass flux inlet condition. The deposition chamber walls are assumed perfectly sticking. The papers also report the incorporation of electronic energy in the particle interaction models.

Similarly, Kersch et al [46] consider pseudo one-dimensional and three dimensional representations of a vapor deposition chamber. They include the vapor species and the background gas species too in their simulation. Their simulations model the vapor atom inlet (the target) as a spatially varying flux profile. They also model a background gas inlet / exit that connects to a source of background gas at a reference temperature and pressure. Surfaces are assumed to be perfectly sticking.

Hass et al [28] use a DSMC code (Icarus) [109] to study jet assisted deposition. This by definition involves multiple species and multiple inlets for these species. They do not consider any chemical reactions in the gas phase and also do not require spatially varying inlet models. However, there is a significant gradient in the cell sizes which makes it necessary to simulate the chamber flows in two steps with different levels of resolution in each. Again, they also model the solid surfaces as perfectly sticking for incident vapor atoms.

Typical vapor deposition problems seem to require a multi-species, multiple inlet capability. This is because of the common need to study flows that involve both the (film material) vapor phase and the background gas phase. There are also applications for being able to include multiple vapor species in simulating alloy deposition. In either of these scenarios, chamber designs allow for different inlets for each of these species and this is most often the case. Apart from this, being able to model spatially varying inlet properties brings the simulation model one step closer to real-world vapor deposition chambers. This allows the simulation to mimic the influx temperature, velocity and angular distribution profiles at the vapor phase source (the target). Quite obviously, there is a need to model solid surfaces as perfectly sticking surfaces for the vapor atoms. This is a basic prerequisite to studying vapor deposition flows.

6.1.2 Evaluating PDSC's abilities

Armed with this understanding of the requirements, studying the capabilities of the PDSC yields a list of features that will be required to enable PVD simulations.

To begin with, the PDSC is inherently capable of handling multiple species in a flow simulation. It accepts as user input, the species fractions and the parameters that define the collision models for each species. The code then uses this information to introduce particles into the system with the specified composition and properties. However, when obtained from the original authors, the code could not handle multiple inlet conditions. This meant that though a flow domain could contain multiple flow inlets, all of them would be initialized with the same flow properties. The only exception to this was the ability to specify a vacuum boundary condition at an inlet. This mimicked an absolute vacuum at that boundary. In hindsight, such a design can be anticipated because the PDSC was initially intended for high speed reentry-like flows. Hence the code naturally supported the typical features of external flows, where all inlets or exits from the flow domain were assigned freestream conditions.

As an extension of these limitations, the PDSC recognized only one global flow composition specification. That is, the species fractions used at any inlet was the same and could not be controlled individually. This was also the same composition used to populate the domain when the simulation was initialized. The PDSC did not also permit any spatial variation of the flow properties along an inlet.

Gas - Surface interactions in the PDSC can vary between diffuse reflection or specular reflection. An accommodation coefficient permits a surface to exhibit both behaviors to different degrees. However, the surface interaction model does not include the capability for completely plastic collisions between the particle and the surface. Hence, the process of deposition whereby incident vapor atoms stick to chamber walls could not be modeled.

6.1.3 Enhancements required

From the previous discussion, a list of features required to enable the PDSC to simulate vapor deposition flows can be drawn up. Most of the items in the list below are essential requirements whose absence will hinder the tackling of moderately realistic vapor deposition flow scenarios akin to the ones available in literature.

- **Multiple inlet conditions** to model inlets with different inlet properties. This will permit the simulations to include sources for different species (e.g. metal vapor and background gas).
- **Enhanced multi-species capability** to permit the variation of species fractions at each inlet. Again, this will be required to model typical vapor deposition chambers.
- **Spatially varying inlet flow properties** to model the vapor source profiles. Several vapor generation mechanisms cause non-uniform heating of the target surface resulting in property gradients along the surface. Modeling this is quite critical to accurately estimating the gas phase flow contours.
- **Sticking surfaces** to model the deposition process. Such an interaction model with solid surfaces is an essential mechanism to mimic the removal of vapor atoms from the gas phase. It is also necessary to be able to select which species may stick to solid surfaces as we desire only vapor atom collisions to display this behavior.

6.2 Multiple Inlet Conditions

The PDSC code was already capable of accommodating multiple inlet boundaries in a flow domain. However, the biggest chink in its abilities was that users could not assign different flow conditions at each of these boundaries. Hence the first of the requirements was the implementation of a multiple inlet conditions capability.

As was just mentioned, approaches that would require modifications to the parallel modules were rejected. Fortunately, much of the inlet face processing is done in the sequential, pre-processing portion of the PDSC. Allowing multiple inlet conditions also fell in as a seamless extension of these pre-processing modules.

```

Input: Grid File
Input: Mapping between boundary number (ID) and boundary type (T)
Input: Inlet flow properties to be applied to all inlets
Read the grid file
Generate all node, cell and face book-keeping data
Identify all the cell faces on boundary surfaces
foreach cell face f on the domain boundaries do
| Identify the boundary number (ID) that the face f lies on
| Identify the boundary type (T) corresponding to this ID
| switch boundary type T of face f do
| | case Mirror boundary
| | | Process as necessary
| | | .
| | | .
| | case I/O boundary
| | | Assign global inlet flow properties to the cell that f belongs to
| | | Write the cell information out to an inlet cells file
| | case Vacuum boundary
| | | Assign vacuum conditions to the cell that f belongs to
| | | Write the cell information out to an inlet cells file
| | | .
| | | .
| | | .
| | case Solid boundary
| | | ...
| | case Other boundary types
| | | ...
|
Result: File listing all inlet faces with flow properties
Feed the parallel, direct simulation engine with this inlet cells file

```

Figure 6.1 Original inlet handling algorithm in the PDSC

Figure 6.1 provides a rough sketch of the relevant steps in the processing of inlet faces on the computational domain boundaries. As can be observed, the PDSC recognizes only one global set of flow conditions at all I/O boundaries. This set of macroscopic flow conditions are used to initialize the properties of all particles

that enter through any inlet. However, it should be observed that the core direct simulation modules do not have any knowledge of boundary surfaces, their IDs or flow conditions at these surfaces. The pre-processing modules feed the main engine with a list of inlet cells and flow properties at these cells. Hence, the implementation of multiple inlet conditions need not introduce changes into the direct simulation engine. Retaining the interaction between the pre-processing and main codes as it is, all the work required to handle multiple inlet conditions can be done in the sequential modules. Figure 6.2 describes an overview of the implementation.

Because, the PDSC is based on a direct simulation approach, the only occasions that it deals with macroscopic flow variables are at either ends of a simulation exercise: to convert the input flow properties into particle positions and velocities and to extract macroscopic flow descriptions out of statistically averaged particle motion and collision data. An effect of this approach is that the PDSC code does not define a data structure to track macroscopic flow variables for each cell / face in either the sequential or parallel phases. The only exceptions are at inlet boundaries, where the parallel direct simulation engine stores macroscopic flow properties for each face.

This one exception has proved very beneficial as it means that the parallel code can handle different flow properties at each face without any modifications. However, part of the task in implementing multiple inlet conditions has been to define such a data structure in the sequential pre-processing modules. One restriction to such an approach is that the size of this data structure (and hence, memory requirements) not scale with the grid size (number of cell / faces). This is desired so that the memory footprint of the sequential code be kept to a minimum, thereby allowing it to process larger grids for more complicated flow domains. These objectives have been achieved by storing the boundary conditions in a lookup table indexed by a *boundary condition number*. The size of this lookup table is independent of the number of cells / faces on the domain inlets. Multiple surfaces / cells can share a single boundary condition and hence there are no notable extra demands on memory.

Input: Grid File

Input: Mapping between boundary number (ID) and boundary condition number (BCnum)

Input: File listing all inlet flow conditions

Read the grid file

Generate all node, cell and face book-keeping data

Create a lookup table that stores the flow conditions corresponding to each boundary condition number Identify all the cell faces on boundary surfaces

foreach *cell face f on the domain boundaries* **do**

Identify the boundary number (ID) that the face f lies on

Read boundary condition number (BCnum) for this boundary ID

Determine the boundary type (T) corresponding to this ID

switch *boundary type T of face f* **do**

case *Mirror boundary*

. . .

.

case *I/O boundary*

Look up macroscopic flow conditions corresponding to this BCnum Assign these flow properties to the cell that f belongs to

Write the cell information out to an inlet cells file

.

.

.

case *Solid boundary*

. . .

case *Other boundary types*

. . .

Result: File listing all inlet faces with flow properties

Feed the parallel, direct simulation engine with this inlet cells file

Figure 6.2 Implementation of the multiple inlet conditions capability

As has been the objective, there are no changes in the algorithm in Figure 6.2 that affect the direct simulation code. It is still passed a file that retains the same format and lists all the cells adjoining domain boundaries along with their flow properties. However, prior to this stage, there are changes to the processing that are visible to the user. The user is now expected to supply a separate input file that is used to generate a lookup table storing a list of macroscopic flow conditions. The user

can then refer to the boundary condition number in this table while assigning boundary conditions to any inlet surface. The user also has to be aware of a numbering scheme that is used to specify the boundary type (solid, symmetry, inlet etc.) and a boundary condition number using a single number. One other change that has been introduced is the removal of the vacuum boundary condition. This has now become a redundant feature as any vacuum boundary can also be specified naturally like any other inlet condition using the current implementation. Finally, the freestream conditions that are specified in the PDSC-PAMR driver script are also rendered redundant as they are not used in the pre-processing code. However, there is still one other set of freestream flow parameters that are used to generate the velocities of the particles used to initialize the domain. A peripheral benefit of introducing multiple inlet conditions has been a reduction in the number of locations where the user has to duplicate input, thus reducing the setup time required for a simulation.

The net effect of this enhancement to the PDSC is that it is now able to handle simulations that consist of multiple flux sources. This has applications not only in vapor deosition simulations but also in other scenarios that demand more complicated geometries and flow setups

6.3 Different Species Compositions at Inlets

Ideally, the specification of the macroscopic flow properties at any point should also include the flow composition. Hence, if it is possible to specify different flow conditions at different boundaries, it should be implicit that the species fractions at each of these inlets can be controlled as well. A brief description of the PDSC's multi-species capabilities can be found in section 5.4. Support for multiple inlet conditions has extended its capabilities. Given these facts, it might be tempting to assume that the PDSC can now handle multiple flux sources, each introducing different species into the flow domain. However, because of implementation specifics, this is not so. Once support for multiple inlet conditions had been implemented, the next step in

the process was to enable the specification of flow composition as part of the inlet conditions, thereby permitting different flow compositions at different inlets.

As mentioned in section 6.2, the parallel, DSMC modules tracked macroscopic flow variables only at the cell faces that lay on domain boundaries. However, these properties did not include the flow composition which was treated separately as a global specification. Hence, introducing the ability to specify different compositions at each inlet meant that the species fractions also had to be handled in a manner similar to the rest of the flow properties. Some background on the primary role of inlet species composition specifications follows.

$$\dot{N}_i = \frac{1}{2\sqrt{\pi}} \cdot n \cdot V_{mp} \left[e^{-\left(\frac{V_n}{V_{mp}}\right)^2} + \sqrt{\pi} \left(\frac{V_n}{V_{mp}}\right) \left\{ 1 + \operatorname{erf} \left(\frac{V_n}{V_{mp}}\right) \right\} \right] \quad [6.1]$$

The inward number flux (\dot{N}_i) through a surface can be calculated as shown in Equation 6.1 (from eqn 4.22 in Bird [32]), where V_n is the inward velocity normal to the surface and V_{mp} is the most probable speed of the molecules in the flow (given by Equation 6.2). It is assumed that the inflow gas is in thermal equilibrium and hence the particle speeds are distributed according to the Maxwellian corresponding to the flow temperature.

$$V_{mp} = \sqrt{2kT/m} \quad [6.2]$$

Hence, the number of particles entering through a face ‘f’ (of cell ‘c’) lying on an inlet boundary can be expressed as

$$N_i|_f = \frac{\dot{N}_i}{F_N} \cdot \Delta t|_c \cdot A|_f \quad [6.3]$$

If the flow in the simulation consists of multiple species, then inward number flux for a species ‘s’ can be written as:

$$(\dot{N}_i)_s = \frac{1}{2\sqrt{\pi}} \cdot (n \cdot w_s) \cdot V_{mp} \left[e^{-\left(\frac{V_n}{V_{mp}}\right)^2} + \sqrt{\pi} \left(\frac{V_n}{V_{mp}}\right) \left\{ 1 + \operatorname{erf} \left(\frac{V_n}{V_{mp}}\right) \right\} \right] \quad [6.4]$$

The original implementation of the PDSC used a global species fraction for w_s in Equation 6.4. The essence of the changes to the core direct simulation modules can be summarized as the replacement of the global species fraction in this expression with an inlet-specific species fraction. This might become clearer if the species fraction is removed from Equation 6.4 and introduced into Equation 6.3 to yield Equation 6.5, where \dot{N}_i has the same meaning as in Equation 6.1 and $w_s|_i$ refers to the species fraction at the inlet 'i' on which the face 'f' lies.

$$(N_i)_s|_f = w_s|_i \cdot \frac{\dot{N}_i}{F_N} \cdot \Delta t|_c \cdot A|_f \quad [6.5]$$

Such an inlet-specific value of the species fraction ($w_s|_i$) is obtained from a sibling data structure of the original that holds the inlet face macroscopic properties. This new structure holds the species fractions at each inlet face and is created from input that is generated in the sequential modules. The species fractions have not been added to the existing data structure that holds the inlet face properties, to minimize the extent of the changes to the direct simulation modules. It should also be noted that with this change, the central DSMC code treats all inlet macroscopic properties at a fine-grained face level. This choice to create face-based data structures instead of simply inlet-surface based structures will get justified in section 6.4.

On the sequential, pre-processing side, the same requirement of minimizing memory usage steered the implementation in a direction similar to that for multiple inlet conditions. The objective was also to give the user the impression that specifying the flow conditions at any inlet would mean that the species fractions are also stated. Hence, the changes visible to the user now require the specification of the species fractions at each inlet if the simulation involves multiple species. This input is then converted into a full face-based specification that is fed to the data structures in the parallel modules using intermediate files.

This capability now gives the user complete control over the flow conditions at each inlet, finally paving the way for vapor transport simulations of some complexity.

Deposition chambers with multiple flux sources and background gases can now be modeled.

6.4 Inlet Flow Profiles

The first objective of this work was to enable the modeling of multiple vapor flux sources. Once support for the complete specification of the flow conditions at each inlet was achieved, the next objective was to improve the ability to model the vapor flux sources in a manner closer to reality.

It was pointed out near the beginning of this chapter that several accounts in literature have used spatially varying inlet conditions to model the vapor flux source. Such inlet property profiles have been measured / modeled and reported in literature, and have been shown to cause variations in vapor fluxes incident on the substrate. The flow property variations are caused by an uneven heating of the surface of the vapor sources (typically observed in electron / laser beam sputtering). Typically, electron beams raster across the target surface or are focused on one spot on the surface. For scenarios with multiple sources, the beam can also be switched rapidly between sources, causing pulsed and uneven heating effects. Modeling literature documents the use of one dimensional or axisymmetric flow property variations to describe the variations in influx properties. Hence, the minimum requirements of this task were to develop a capability to model axisymmetric flow profiles across inlets.

However, apart from the basic requirements, a few additional requirements / restraints to be placed on the implementation were identified. It was desired to use this opportunity to achieve a few other peripheral objectives that would yield fruit indirectly, in terms of facilitating future development.

The first requirement was to enhance the user experience by stipulating that the profile definition should not involve any programming by the user. It was desired that all flow profiles be described as the variation of flow properties across a collection of points. Typical experience with flow profiles in continuum fluid flows revolves

around boundary layer profiles, which can often be described analytically. Hence, this restriction seemed like a drawback as it would increase the effort required to specify such profiles. A collection of points and the flow properties at these points would have to be defined instead of simply coding an analytical function that yielded flow variables as a function of spatial location. However, it must be remembered that the flow profiles that may be encountered in non-continuum flows may not be amenable to such analytical description. In the specific case of vapor sources in PVD, the profile is usually somewhat arbitrary and not likely to be expressed as an analytical formulation. One general solution that could handle both arbitrary and regular spatial variations of the flow conditions was to describe the variation as the flow properties at a collection of points.

The second requirement was that the flow profile specification should be independent of the points defining the grid. In other words, the profile should not be defined by specifying the flow properties at a set of nodes / faces / cells that are part of the grid. In the PDSC's context, this can be justified quite easily. The PDSC workflow invokes the direct simulation modules multiple times in a complete simulation, interspersing these invocations with mesh refinement stages. Hence, during a single simulation, the grid points are likely to change during the mesh refinement stages. Defining a profile based on member features of the initial grid, might make this definition useless as the grid evolves to adapt to the flow predictions! Hence, the profile definition necessarily had to be specified in a manner that did not depend on specific nodes or faces or cells of the grid. It will be seen in the following discussion that such a formulation also permits numerous peripheral benefits, though it does necessitate a means of translating the profile definition to flow specifications along the inlet faces of the grid.

The first additional capability that was desired apart from the requirements was that the flow profile capability not be restricted to one dimensional or axisymmetric descriptions. It was desired that the implementation at least permit or

facilitate flow property variations along two dimensions. Though immediate uses for such a feature cannot be listed as a tangible set of arguments, it definitely seemed prudent to retain greater flexibility for minimal extra complications in the implementation.

An additional desire while designing this feature was that the concept of a macroscopic flow property profile should not be tied to or restricted to apply only to inlet boundaries. In other words, it should be possible to extend the code by permitting property variations independent of the feature it is applied to. This has some immediate and several long-term uses. The immediate application of such a capability would be to specify property (e.g. temperature) variations along other surfaces like solid boundaries. If the implementation facilitates the creation of flow profiles along more than one dimension, then several long-term applications emerge. Flow profiles could be extracted along different regions in the computational domain and written to disk to permit checkpointing during the simulations. Since these checkpoints store only macroscopic flow properties and not complete particle position and velocity information, they would be relatively cheap. These checkpoints could be used for restarting simulations by reading in these profiles and using them to recreate a similar flow scenario in the computational domain. Another application could be the definition of flow profiles for domain initialization thereby minimizing the burn-in time required before the sampling phase of the direct simulation algorithm commenced. Finally, such an algorithm-neutral means of expressing the flow properties along 2D / 3D regions in the flow domain could potentially be used to interface the direct simulation easily with another flow solver module.

The algorithm describing the implementation on the sequential side of the PDSC without going into specifics, can be found in Figure 6.3. Tasks that are actually separate routines in the implementation have been coalesced into bigger units to elucidate the logic rather than the actual implementation. Some of these subtasks are represented separately in Figure 6.4. In order to satisfy all the needs expressed

Input: Grid File
Input: Mapping between boundary number (ID) and boundary condition number (BCnum)
Input: File listing all inlet flow conditions
Input: Profile Definition Files

Read the grid file
Generate all node, cell and face book-keeping data
Read inlet flow conditions file

```

foreach Bcnum do
  | if BCnum is a uniform valued inlet condition then
  |   | Store the inlet conditions in a lookup table indexed by BCnum
  | if BCnum is a profile valued inlet condition then
  |   | // Instantiate a flow profile data structure
  |   | createFlowProfile(profile definition file name)

```

Identify all the cell faces on boundary surfaces

```

foreach cell face f on the domain boundaries do
  | Lookup the boundary number (ID) that the face f lies on
  | Identify boundary condition number (Bcnum) for this boundary ID
  | Determine the boundary type (T) corresponding to this ID
  | switch boundary type T of face f do
  |   | case Uniform valued I/O boundary
  |   |   | Look up macroscopic flow conditions corresponding to this
  |   |   | BCnum
  |   |   | Assign these flow properties to the cell that f belongs to
  |   |   | Write the cell information out to an inlet cells file
  |   | case Profile valued I/O boundary
  |   |   | Calculate the face center coordinates
  |   |   | // Interpolate face center properties from profile
  |   |   | definition
  |   |   | interpolateFromProfile(Profile Number, Face center
  |   |   | coordinates)
  |   |   | Assign these flow properties to the cell and write to the inlet
  |   |   | cells file
  |   | case Other boundary types
  |   |   | ...

```

Result: File listing all inlet faces with flow properties
Feed the parallel, direct simulation engine with this inlet cells file

Figure 6.3 Relevant portions of the algorithm handling inlet flow profiles

```

createFlowProfile(file name);
Input: file containing the profile definition
begin
  | Read the profile definition file;
  | Generate a local coordinate frame;
  | Store the points defining the profile and corresponding flow properties;
  | Run checks and ensure definition is valid;
end

interpolateFromProfile(profile number, point);
Input: query point
begin
  | Transform coordinates to local frame of reference;
  | Mask the appropriate coordinates;
  | Check if the masked point lies on profile;
  | Find bounding point using appropriate nearest neighbor algorithm;
  | Use default / user-specified interpolation algorithm to interpolate flow
  | properties;
end

```

Figure 6.4 Overview of the logic encapsulated behind the flow profile handling interface

above, the complete implementation has been written as a number of independent and generic modules. An object-based approach has been adopted by representing concepts as custom data types tightly associated with accompanying behavior that are hidden behind generic interfaces. Much of what has been developed has been with the intent of reuse and to act as a seed for modernizing the rest of the code. In the process, reusable routines are now available for:

- **Cartesian geometry:** Point, vector and plane definitions
- **Coordinate transformations:** Local frames of reference, global to local and reverse transformations
- **Interpolation schemes:** Generic interface for interpolation routines and implementations for two specific algorithms
- **Data structures** for grid points and macroscopic flow properties

- **Flow profile definitions** that are independent of any geometric or grid features

The flow profile concept has been generalized to permit 1D / 2D / 3D flow profiles. Also introduced is the process of *masking*, whereby simple profiles can be extruded, rotated etc. into additional dimensions. For e.g, a simple one dimensional profile definition could be used to specify the radial variation of properties along a circular surface. Since, the implementation supports higher dimensional flow profiles without any modifications, one can also create spherical, cylindrical, prismatic, pie-slice etc. shaped profile definitions with minimal effort. The potential benefits of being able to export or import 3-D profile definitions have already been discussed.

One of the most helpful features in the implementation is the freedom to define the flow property variations in any coordinate system that might be comfortable. With minimal input from the user the PDSC can now handle the translations and rotations required to convert points in a local coordinate frame to the global frame used throughout the solution grid (and vice versa). This means that the user can specify flow profiles on arbitrarily located and oriented surfaces using a local frame of reference. This greatly facilitates the modeling of complicated chamber geometries without thrusting the effort of computing inflow velocity vectors in global coordinates on the user.

Currently, the default interpolation algorithm is a simple linear interpolation between a set of identified bounding points. However, other interpolation schemes can be added later on with no modifications required to existing code. This is because all the interpolation schemes are hidden behind an interface and the user can potentially change the algorithm used. This might be of some interest when supplied input flux profiles have fewer data points than required for a smooth spatial variation and other interpolation schemes are desired.

From the user's perspective, the only additional task is the creation of a profile definition file for each inlet that requires spatially varying properties. A summary of the data required in a typical profile specification can be found in Figure 6.5.

```

[coordinate frame definition]
  Local origin
  Point on local X axis
  Point on local Y axis
[interpolation mode and masking specification]
  Interpolation mode
  Coordinate mask
[profile definition]
  Number of points in profile
  Number of species in flow descriptions
  List of points defining the profile

```

Figure 6.5 Contents of a profile definition file

6.5 Gas-Surface Interactions: Extended

chapter 4 describes the general principles of gas-surface collisions. Discussed there are the different mechanisms by which particles incident on a surface are reflected back into the gas phase. However, PVD requires the modeling of a different kind of surface behavior. As the name might suggest, PVD is basically a deposition process. This means that the simulation model has to include capabilities to mimic the capture of gas phase particles onto solid surfaces. Bird [32] does not talk at length about the aspects of modeling this phenomenon of surface capture. However, PVD simulation literature regularly talks about sticking surfaces. A sticking surface typically means a surface on which incident gas-phase particles are captured permanently onto the surface. In other words this models gas-surface interactions that mimic deposition processes.

Modeling completely plastic collisions between vapor atoms and solid surfaces is crucial to enabling PVD simulation studies. Such a model will provide an important means of removing mass from the simulation domain without which the only studies possible will be the vapor plume expansions into vacuum. Without sticking surfaces, all incident vapor atoms will get reflected off the substrate influencing the flow in the vicinity and disrupting estimates of the gas phase properties. Vapor atoms incident

on solid surfaces *have* to be modeled as sticking to the surface to obtain any accurate models of PVD.

One question that cannot be answered so unequivocally is whether all incident vapor atoms should be modeled as sticking to the surface. Most simulation studies assume perfectly sticking surfaces which means that all incident vapor atoms stick to the surface. Hass et al [28], Kersch et al [46], Fan et al [24], Balakrishnan et al [1] have all used this same assumption. Balakrishnan et al [1] study the influence of this assumption by varying the sticking coefficient slightly to observe its effect on gas phase phenomena. Their results lead to conclusions that the flow field properties are affected noticeably by altering the sticking coefficient. However they also observe that the deposited film thickness does not seem to be affected significantly by changes to the sticking coefficient in the vicinity of 1.0, and hence justify the use of perfectly sticking surfaces.

To summarize the objectives, the primary aim was to allow the PDSC to model the process of vapor atoms to stick to surfaces. It was also decided to incorporate a sticking coefficient so that sensitivity studies could be performed in the future. It is apparent that a PVD simulation model would only want the metal vapors to stick to the solid surfaces and not the lighter (and potentially lower energy) gas phase particles. Hence, the final requirement was to create a capability to handle different behaviors for the surface interactions of different species.

The original logic in handling gas surface interactions is summarized in Figure 6.6. The algorithm has been kept deliberately short in order to highlight only the steps pertinent to this discussion. It must be mentioned that the complete gas-surface interaction handling occurs during the *MOVE* stage of the DSMC algorithm chapter 4. The current implementation has introduced an additional clause if molecules are found to be incident on a solid surface. A user-specified fraction of the incident particles are now removed from the simulation once the collision data are sampled. It is the

```

foreach Particle p do
  Move the particle based on current position and velocities
  if p crosses any cell boundaries then
    Identify face crossed (f)
    if f lies on a solid boundary then
      Sample a random fraction  $r_1$  from uniform distribution  $U_1$ 
      if  $r_1 < r_{diff}$  then
        | Perform a diffuse reflection
      else
        | Perform a specular reflection
    else
      | Handle other types of boundaries (I/O, mirror, periodic, CPU
      | etc.)

```

Figure 6.6 Logic flow in handling gas-surface collisions

```

foreach Particle p do
  Move the particle based on current position and velocities
  if p crosses any cell boundaries then
    Identify face crossed (f)
    if f lies on a solid boundary then
      Sample a random fraction  $r_1$  from uniform distribution  $U_1$ 
      if  $r_1 < r_{sticking}$  then
        | Calculate the contribution of the incident particle to the
        | change in surface properties
        | Remove the particle from the simulation
      else
        | Sample a random fraction  $r_2$  from uniform distribution  $U_2$ 
        if  $r_2 < r_{diff}$  then
          | Perform a diffuse reflection
        else
          | Perform a specular reflection
    else
      | Handle other types of boundaries (I/O, mirror, periodic, CPU
      | etc.)

```

Figure 6.7 Modifications to handle sticking surfaces

remaining fraction of reflected particles that are then handed over to the original code for either diffuse or specular reflection. This is described in Figure 6.7.

The fraction of particles that stick to the surface is actually dependent on the incident gas species and also on the nature of the surface. Hence, ideally, the sticking coefficient should be specified as a function of the species and the boundary surface number in the computational domain. For example it might be interesting to model the substrate surface material as perfectly sticking, while the chamber walls are not so. However, such knowledge of the differences caused by the material or nature of the solid surface is not available in literature. In the absence of conclusive evidence to justify a need, the additional complications have been avoided and sticking coefficients have been associated with just the species of the incident gas. Changes visible to the user now require the specification of the sticking coefficient as an extra parameter for each species in the flow.

6.6 Handling Backscatter: Constant mass flux inlets

Backscatter is the phenomenon of molecules entering the flow domain being scattered back through the inlet face because of collisions with other molecules in the domain.

6.6.1 Significance in Vapor Deposition Flows

Backscatter is not something that is unnatural or specific to the type of flows being dealt with in this work. Even in a stationary gas at equilibrium, there are always molecules that will get scattered back across any reference surface that might be defined. However, what does make this phenomenon significant for vapor deposition flows is the lack of a significant macroscopic flow velocity at the inlet. The vapor flux that enters the deposition chamber is usually modeled as one with a zero macroscopic velocity. This means that the molecular velocities very close to the inlet face are of

the order of the mean thermal speed of the entrant species at the corresponding temperature. When this is the case, the probability that collisions with other molecules in the domain (which also possess similar thermal speeds) will reverse the direction of motion of an entrant molecule is relatively high. This is in contrast to scenarios where the flow has a definite macroscopic velocity. For e.g., for high speed rarefied flows, because the macroscopic flow velocities are of the order of the thermal speeds, backscatter phenomena are probably only of theoretical interest.

There is one other perspective from which this can be understood. In typical flow simulations, inlet faces are only imaginary surfaces that act as the boundary between the flow domain of interest and the flow without. Hence, in the immediate vicinity of an inlet face, on either side of it, there exists fluid with almost identical temperatures and velocities. In such a scenario, if the fluid has a zero macroscopic velocity, then the fraction of particles that are scattered back across the face in either direction will be equal. Hence this process will not influence the flow near the inlet significantly. However, in vapor deposition setups, the inlet face is an actual physical boundary that separates the gas phase from the solid surface or melt pool. Hence both vaporization and backscatter are processes that occur only on one side of the inlet surface - in the gas phase within the flow domain. Hence backscatter back into the melt pool is not counteracted by a similar process outside the domain of interest. This will influence the flow properties near the inlet and, as a result, elsewhere in the domain.

Backscattered particles in a vapor deposition process will impinge on the melt pool and recondense. This will result in the transfer of energy to the melt pool. Depending on the backscatter flux, this might be a non-trivial effect on the target surface and might cause the heating of the vapor source itself. An increase in the temperature of the melt pool will affect the number and velocities of the molecules that evaporate from the surface, which in turn affect the flow in the chamber and the backscatter processes in return! These effects will stabilize at some melt pool

temperature, but in essence will drive the vapor generation process away from the initial (intended) conditions. As gas phase simulations of the chamber usually model the melt pool as a fixed inlet condition, the backscatter processes will be decoupled from the numerical flux generation model. This might affect the flow in the chamber.

There is another point of concern in modeling exercises. Experimental studies of vapor deposition typically measure the feed rate of the source material into the chamber, This yields a mass rate of evaporation which is used to specify a mass flux at the inlet surface of the computational domain. Direct simulation of the chamber flows will now result automatically in backscatter processes causing some of the inlet vapor molecules to exit the simulation domain. Hence, the net mass flux that enters the domain at each iteration will be lesser than the intended value that has been set as the boundary condition. This will obviously affect the flow predictions throughout the chamber and can have quite a significant effect on the fluxes near the substrate.

6.6.2 Implementation

Though the process of backscatter is something that can be automatically captured by the direct simulation process, there may be a need to account for or adjust for the several effect of backscatter discussed above. Several possible approaches suggested themselves upon reflection:

- **Solid boundary inlets:** The most natural means of correcting for the backscatter flux *and* allowing the backscattered molecules to influence the vaporization surface's temperature is to model the vapor generation face (i.e. the target surface) as a solid boundary, but with a flux source capability. In essence, this would create a new, hybrid boundary condition that would exactly address the needs of the situation. Such a surface would participate in inlet face operations like flux entry etc. and also in solid face operations like gas-surface collisions etc. Hence, the surface would be able to introduce new particles into the domain and

also reflect incident particles back into the domain. It would also account for the energy transfer to the surface due to the impinging backscattered molecules, allowing the surface temperature distribution to evolve during the simulation as a function of the backscatter flux. This may be able to increase simulation fidelity significantly, as the influence of incident particle collisions on the target surface is also incorporated into the numerical model. There does not seem to be any available literature that has used this approach to model vapor source surfaces in film deposition simulations.

- **One-way inlets:** Simply adjusting for the backscatter flux to ensure constant mass flux inlets might improve the fidelity of the simulation considerably. This proposed approach modifies existing inlet conditions to create a scenario where particles that are incident on the inlet face inside the flow domain do not leave the domain. Instead they are reflected back into the domain. Particle reflection may either be diffuse or specular. However diffuse reflections are closer to physical reality as it mimics the removal of the backscattered particle and the introduction of a new vapor particle that is based on the surface conditions. The reflection process however, does not influence the inlet properties in any way as it is not treated like a surface collision. It is simply a modeling gimmick to ensure a constant mass flux at the vapor inlet. Modeling the inlet conditions statically, effectively decouples the influence of the gas phase on the target surface temperature. It achieves a constant mass flux by permitting only flux entry and preventing flux exit, thereby acting like a one-way inlet. An approach similar to the one proposed here has been used by Fan et al Fan00 in their simulations to correct for backscatter.
- **Adjusting inlet fluxes** to correct for backscatter is also another means of achieving a constant mass flux at the vapor source. Such an approach has been adopted by Balakrishnan et al [1]. They monitor the spatial distribution of the

backscatter flux across the inlet and adjust their inlet fluxes to account for this backscatter.

Though not identified as a necessary requirement in the initial stages of this work, preliminary results highlighted the need to account for the backscatter of the vapor phase atoms back into the melt pool so as to maintain a mass flux closer to intended values. Searches in existing literature also pointed to the influence of the backscatter phenomenon and the non-trivial fraction of molecules that recondense on the target surface [110, 1]. Without extensive investigation into the results, it was decided to implement the capability to handle at least the influence of the backscatter on the net mass influx at a vapor source.

The current implementation of the PDSC would have required extensive changes to both the pre-processing and direct simulation components to permit the introduction of a new boundary condition like solid-inlet boundaries. Though it would have been interesting to observe the change in predicted flow contours while incorporating an evolving target surface temperature, the focus of the current work is to enable the PDSC to handle PVD simulations in existing literature. Hence, it was decided to implement one-way inlets into the code to permit the maintenance of a constant mass flux at the vapor inlets. The changes involved simply adding clauses to the particle move algorithm in the DSMC model to diffusely reflect any particles that crossed inlet boundaries heading outside the simulation domain. The reflected particle is assumed to have completely accommodated to the inlet temperature at the crossed face and is hence given velocities sampled from the Maxwellian distribution at that temperature.

The implementation includes no user-visible changes. User options permitting the toggling of this feature has been implemented but has been held back in favor of the future introduction of inlet-specific switches to perform this function. The code is now capable of handling constant mass flux inlets for general applications.

6.7 A Perspective on the Implemented Enhancements

It is casually interesting to note that the one unifying factor for all the implemented enhancements to the PDSC focus on enhancing or implementing different flux entry / exit mechanisms from the simulated deposition chamber. The majority of the contributions of this work focus on improving the PDSC's deftness at handling realistic inlet conditions. Work has also been done to enable appropriate flux removal mechanisms (sticking surfaces) and required means for constant influx conditions (backscatter handling). In hindsight, this might seem expected as the PDSC, originally designed for external flows, would definitely require additional capabilities in handling internal flows. Also, attempts to study and evaluate the performance of the core particle or surface interaction models in vapor deposition scenarios would have to wait until the numerical model duplicated the flow chamber boundary conditions as closely as possible in the simulation. Hence, this work has focused on enhancing the PDSC in an attempt to move in this direction.

PART III

RESULTS AND DISCUSSION

CHAPTER 7

VALIDATION FOR HIGH SPEED FLOWS

The first step upon acquiring the PDSC code was to put it through a series of validation exercises. A set of high-speed flows intended to mimic atmospheric re-entry conditions have been used for this purpose. Publications used for reference provide experimental measurements of such flows and some also perform similar DSMC computations. This section details some of the key results from this validation exercise in an attempt to demonstrate the ability of the code to handle such flows. A summary of the relevant data for each of the cases can be found in Table 7.1. Some of these results are reported in Venkataraman et al [111] and Cheng et al [112].

7.1 Corner FLOW over Perpendicular Flat Plates

The first case considered is the supersonic flow over the corner formed by two perpendicular flat plates, verified with the data provided in Bird [32]. This problem is analyzed in detail by Bird and is used to compare the accuracy of his three dimensional code against his earlier 2-D demonstration code. The flow possesses characteristics very similar to the flow over a flat plate, but at the same time is a good starting test to verify the 3-D solver. As the whole domain is included in the computations, we can observe if the code is able to capture the inherent symmetry in the flow with reasonable accuracy.

The flow is a stream of Argon at a temperature of 300 K and a freestream Mach number of 6 flowing along the longer side of the plates. The leading edges of the plates are 0.05 m from the edge of the domain. The plates themselves are at

Flow Case	A	B	C
Gas	Argon	Nitrogen	Nitrogen
Inlet Number Density (m^{-3})	1.0×10^{20}	2.0×10^{22}	8.608×10^{21}
Inlet Temperature (K)	300	261.4	22.8
Inlet Mach Number	6.0	1.0	11.25
Inlet Velocity (m/s)	1773.917	329.642	1077.0
Solid surface temperature (K)	1000	296	600
Thermal Accommodation	Complete	Complete	Complete
Dimensions of flow domain	$0.3 \times 0.18 \times 0.18m^3$	$39.68 \times 39.68mm^2 \times 45^\circ$	$50 \times 30mm^2 \times 45^\circ$
Num. of cells	47000	32000	66000
Num. of particles	4.5×10^5	4.0×10^5	5.8×10^5
Num. of Timesteps	30000	60000	60000
References	Bird [32]	Weaver [113]	Dogra [114]

Table 7.1 A summary of some data for the test cases: A - Flow over perpendicular plates; B - Vacuum expansion through orifice; C - Flow over a sphere

right angles and are aligned with the y & z axes. The surface temperature of the plates is 1000 K and they are modeled as fully diffusive walls with complete thermal accommodation. The Knudsen number based on the free stream mean free path and the plate length is 0.043. A sketch of the plates and the computational domain is given in Figure 7.1.

The results predict the symmetric nature of the flow field quite well. The density and temperature contours, normalized with respect to the freestream values, at $x/L = 0.4$ (measured from the leading edge) are plotted in Figure 7.2 and Figure 7.3. These contours agree well with the results published in Bird [32]. Figure 7.4 compares the normalized density along the line of symmetry between the two plates at a distance of $x/L = 0.4$ from the leading edge with the results in Bird [32].

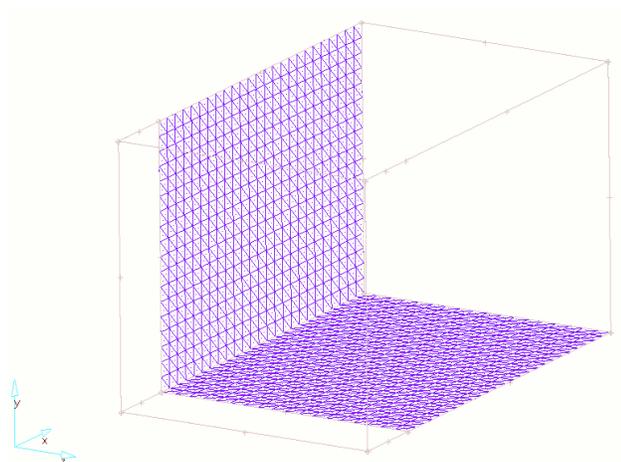


Figure 7.1 Schematic of the perpendicular flat plates

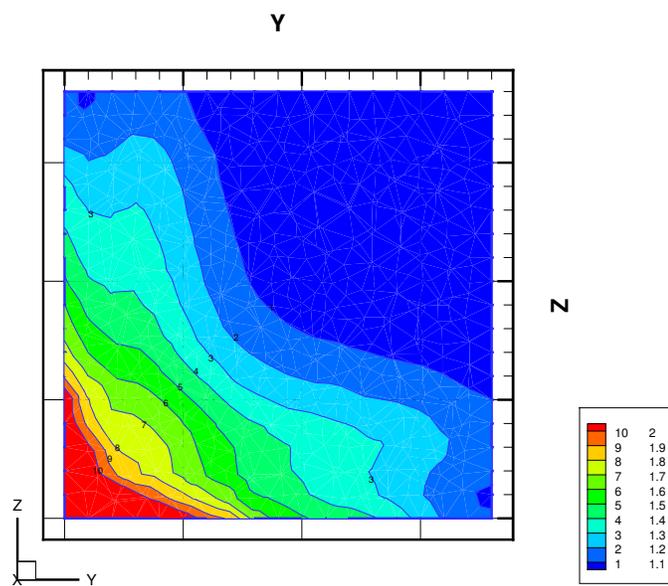


Figure 7.2 Normalized density at $x/L = 0.4$

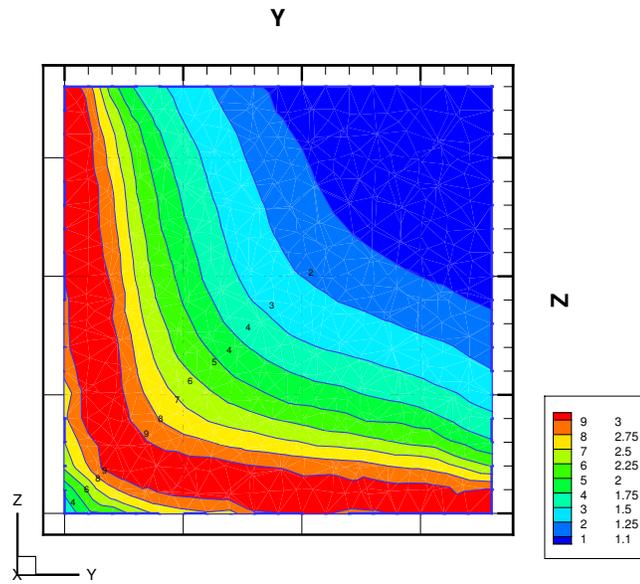


Figure 7.3 Normalized temperature at $x/L = 0.4$

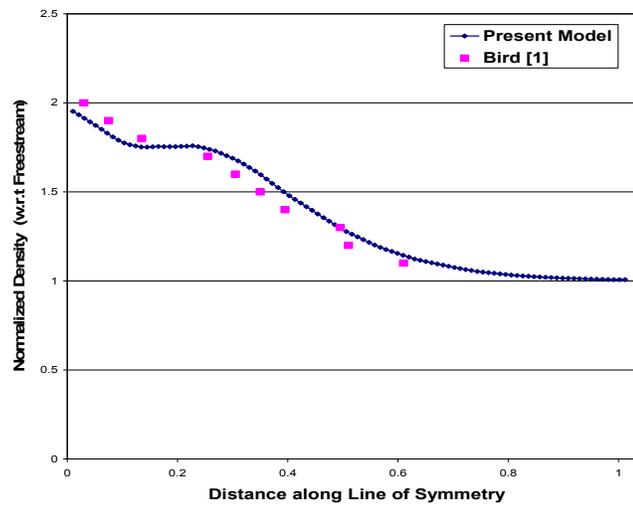


Figure 7.4 Normalized density along the plane of symmetry at $x/L = 0.4$

7.2 Expanding Flow through an Orifice

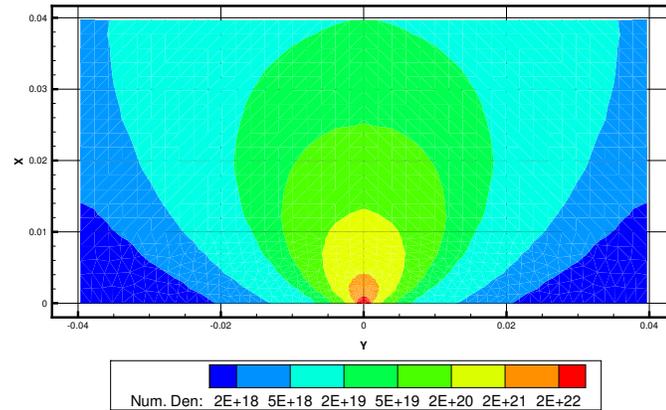


Figure 7.5 Number density contours for expansion through an orifice

The chosen problem mimics typical environments encountered by in-space propulsion systems, where nozzle plumes experience sudden expansion into vacuum. Such problems are of great interest in the design of satellites and space vehicles. One would typically expect to know the expansion profile of such a plume, regions in the neighboring structure which might be affected by backflow, the heating effects caused by plume impingement etc. The design of heat shields and placement of equipment on the vehicle hulls is strongly influenced by such data. As a first step it was attempted to predict the expansion into vacuum from a sonic orifice outlet. This problem uses the experimental data obtained by Weaver et al [113] and previous simulation results by Wu et al [97] for comparison.

The flow considered is that of a jet of nitrogen through an orifice of diameter 1.984 mm, followed by an expansion into vacuum. As the flow is axisymmetric, only one eighth of the actual flow domain is modeled in the simulation, resulting in appreciable savings in computational time. A point of consideration is that the slice

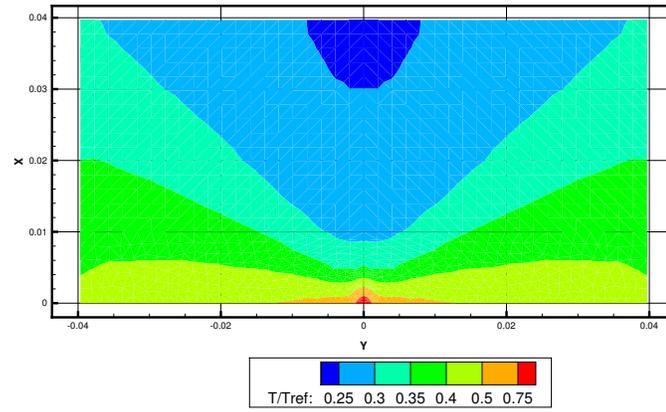


Figure 7.6 Normalized temperature contours for expansion through an orifice

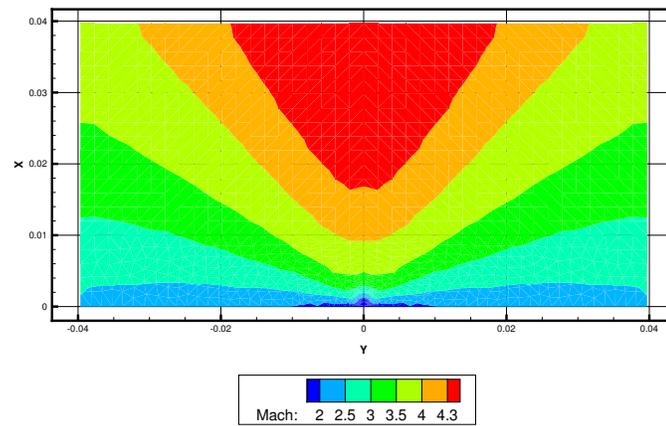


Figure 7.7 Mach number contours for expansion through an orifice

not be too narrow, as it results in skewed cells near the central axis, which affect the computed local Knudsen numbers. The side faces of the slice are treated as walls with specular reflection to model the symmetry boundary. Vacuum boundary conditions are used for the other faces. The Knudsen number based on the freestream mean free path and the orifice diameter is 0.0163.

The simulation initially used about 20,000 cells and approximately 2.3×10^5 particles. The results obtained from this first run were then used to perform mesh refinement that increased the number of cells to about 32000. The freestream parameter was chosen to be one hundredth of the inlet number density so that the mesh is refined around the orifice inlet.

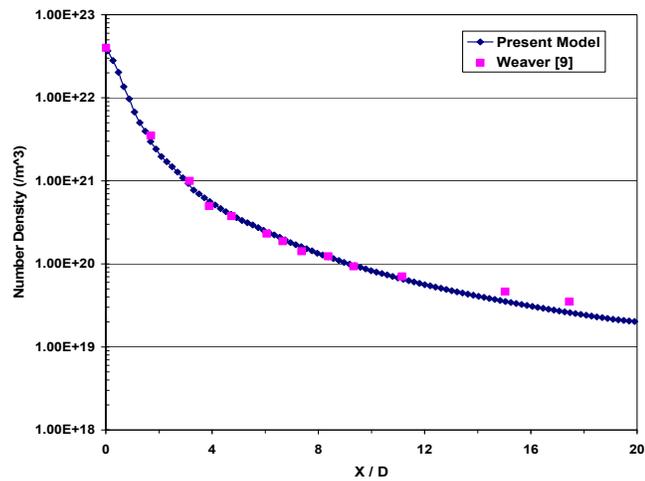


Figure 7.8 Normalized density along the central axis of the orifice

The results obtained are found to agree admirably with those presented in the mentioned references. Figure 7.5 and Figure 7.6 present the number density and normalized (w.r.t inlet value) temperature contours for the expansion plume through the orifice. Figure 7.7 presents the mach number contours for the flow. The contours match well with previous simulation results obtained with an earlier version of this code by Wu et al [97]. We observe that the code faithfully captures the large drop in

number density along the centerline. It matches with a drop of three orders of magnitude within a distance of 20 times the orifice diameter, recorded both experimentally by Weaver [113] and through simulation by Wu [97], as observed in Figure 7.8.

7.3 Hypersonic Flow over a Sphere

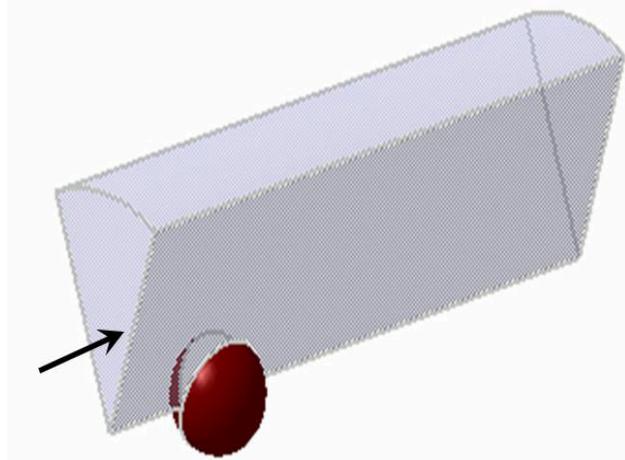


Figure 7.9 Schematic of the computational domain for flow over 10mm sphere

The third benchmark chosen is the hypersonic flow over a sphere of diameter 10 mm. The problem was modeled on, and results verified from, the data provided by Dogra et al [114]. The problem is of interest because of the steep gradients involved and the flow structure around the sphere. It is vital that the DSMC implementation be able to capture the flow structure around such objects at high Mach numbers under rarified conditions. The flow under consideration is at Mach 11.25. The freestream Knudsen number based on the diameter of the sphere is 0.078. Hence the problem represents a good test of the capabilities of the code. A sketch of the computational domain with a mock sphere in place for reference is shown in Figure 7.9.

The simulation results are plotted in Figure 7.10, Figure 7.11 and Figure 7.12. The plots indicate that the code has captured the bow shock that forms at some distance in front of the sphere. The normalized density (upper half) and temperature

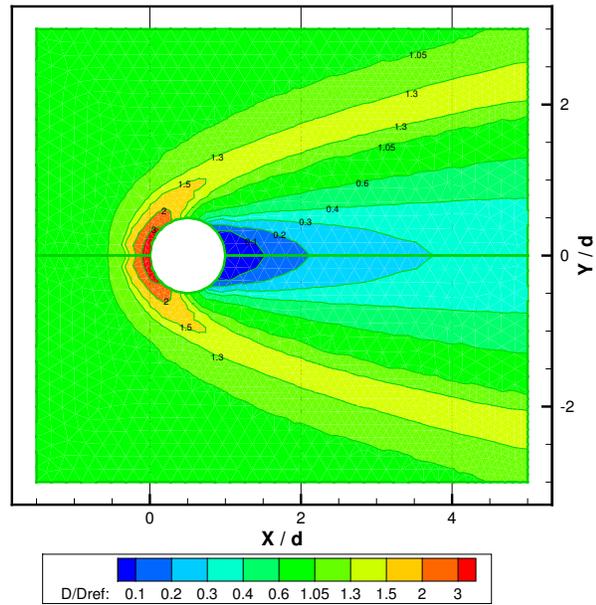


Figure 7.10 Number density contours for flow over 10mm sphere

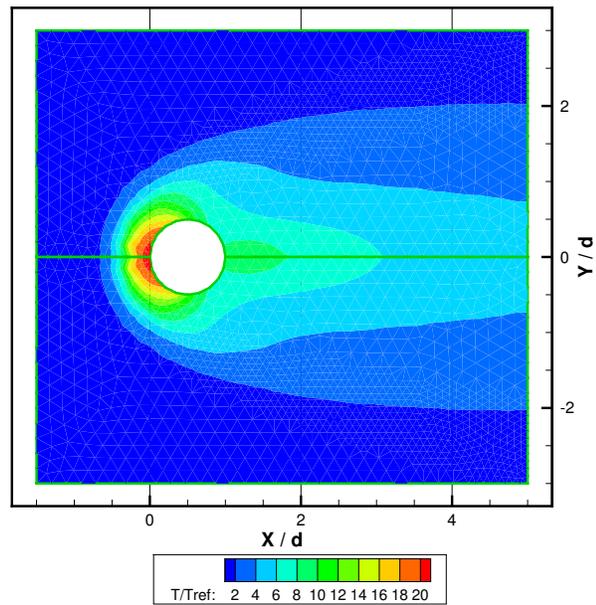


Figure 7.11 Normalized temperature contours for flow over a 10mm sphere

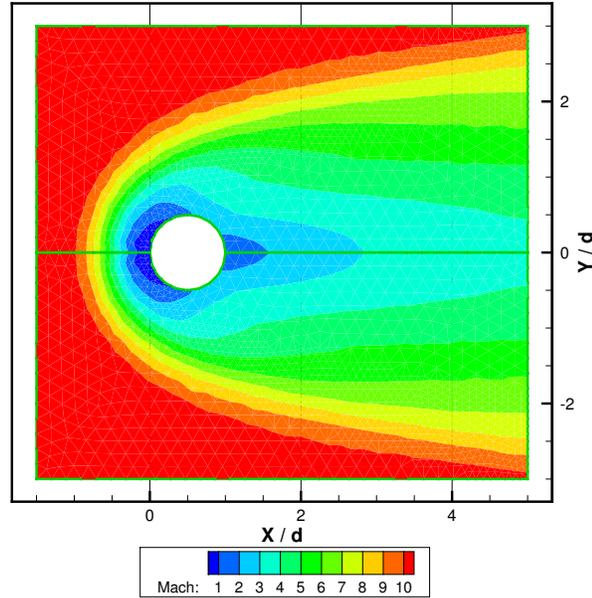


Figure 7.12 Mach number contours for flow over a 10mm sphere

(lower half) contours show that the code predicts the highly compressed shock in front of the sphere, as well as the rarefied wake region behind the sphere. A plot of the predicted flow density and temperatures along the stagnation line in front of the sphere is presented in Figure 7.13. The results agree well with those from Dogra et al [114], as well as with results obtained from earlier versions of the code by Wu et al [97]. The predicted rotational temperature, is however, lower than the predictions by Dogra. This may be due to an insufficient grid refinement immediately ahead of the sphere, where the number density is very high and would require a finer grid to resolve the gradient. The plot clearly shows that the flow is in thermal non-equilibrium in the shock region, as the translational and rotational temperatures are quite different. A more detailed explanation can be found in Dogra [114]. The good agreement of the results with previous data brings out the ability of the code to capture shocks faithfully, and to predict high flow field gradients.

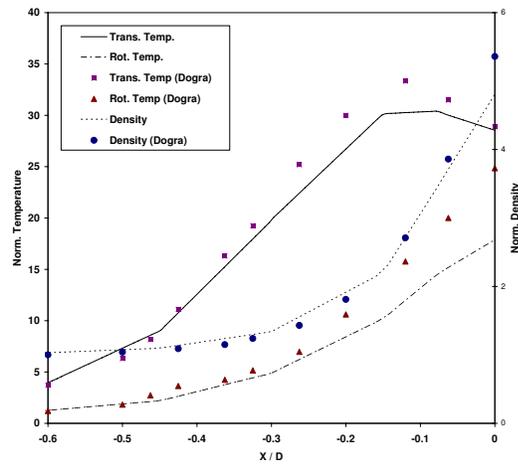


Figure 7.13 Stagnation line properties for flow over sphere

7.4 Flow past a Blunt Cone

The hypersonic flow field past blunt bodies is of special interest for re-entry flow simulations. The large gradients across the shock region and the rarefied wakes behind the body in such hypersonic flows are typical of re-entry situations. The problems chosen are Mach 20 flows that are definitely in the non-continuum regime. That the solver is capable of capturing flow features of interest like shocks has been demonstrated from the previous test case. However, this problem is of more real interest as it models the blunt nose cone that is found in most high-speed vehicles. The results obtained are hence very much of interest and were studied quite closely.

A set of 2 cases, each with 3 different angles of attack (0° , 10° & 20°), of a hypersonic nitrogen stream flowing past a 70° blunt cone are simulated. Details of the geometry can be obtained from Moss and Price [115]. The numerical results are compared with experimental data from the SR3 experiment available in Allegre et al [116, 117, 118], and with simulation results from Moss and Price [115]. The computational domain of the flow was either a 16th slice for the axisymmetric case or a semi-cylindrical domain for the non-zero attack angles. The flow conditions for the

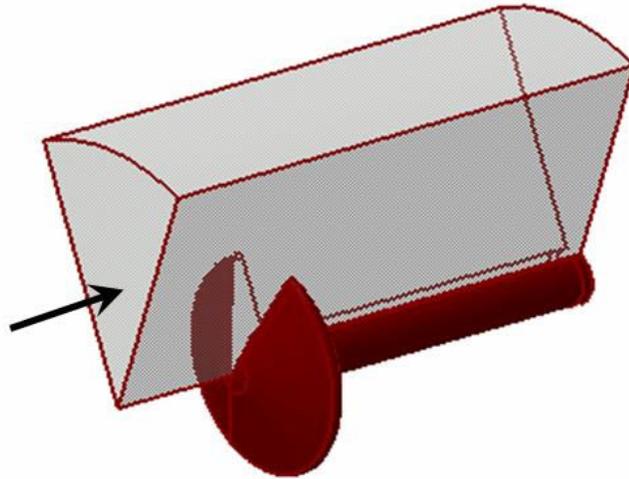


Figure 7.14 Blunt Cone (0° attack): Flow domain

	Case 1	Case 2
Mach Number	20.2	20.0
Velocity (m/s)	1502.0	1502.0
Density ($\times 10^{-5} \text{ Kg/m}^3$)	1.73	5.19
Temperature (K)	13.3	13.6
Mean free path (mm)	1.59	0.54
Gas species	N_2	N_2
Wall Temperature (K)	300	300

Table 7.2 A summary of relevant data for the test cases involving flow over blunt cone

six different cases considered are tabulated in Table 7.2. The computational domain for the axial flow is depicted in Figure 7.14.

All cases were run for a total of 60,000 time steps. Samples were not taken for the initial 6000 time steps, so that the flow in the simulation domain might become stable and any initial transience might die out. The number of simulated particles in any case was at least 5 times the total number of cells in the domain. The normalized density and temperature contours of the flow for Case 2 have been plotted for 0° , 10°

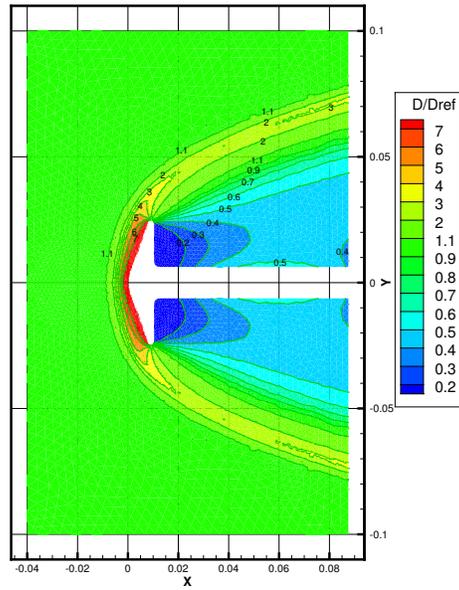


Figure 7.15 Blunt Cone (Case 2, 0° attack): Normalized density contours

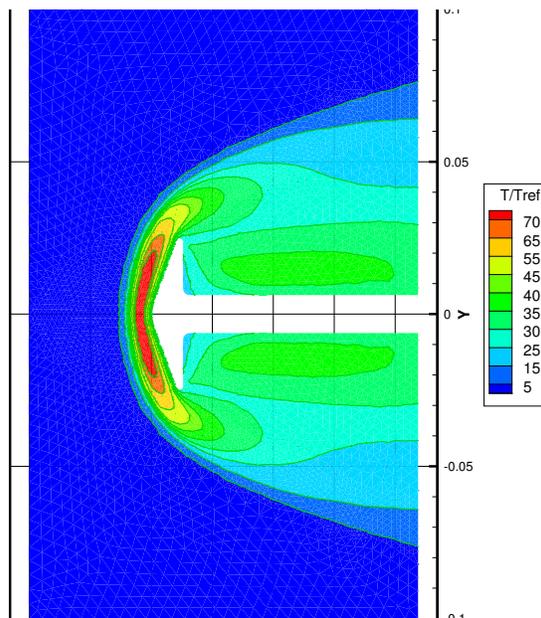


Figure 7.16 Blunt Cone (Case 2, 0° attack): Normalized temperature contours

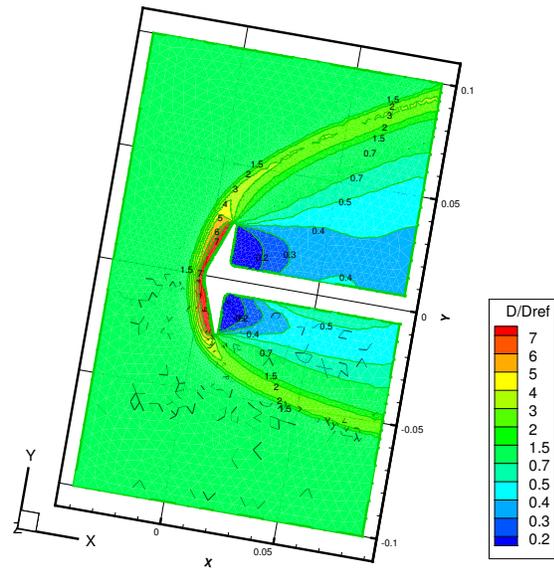


Figure 7.17 Blunt Cone (Case 2, 10° attack): Normalized density contours

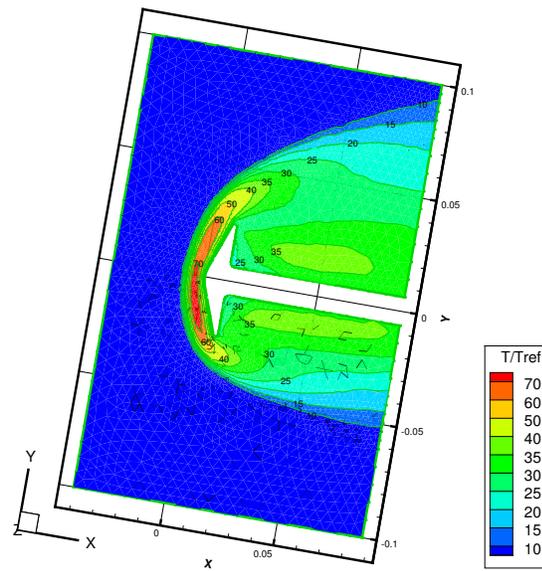


Figure 7.18 Blunt Cone (Case 2, 10° attack): Normalized temperature contours

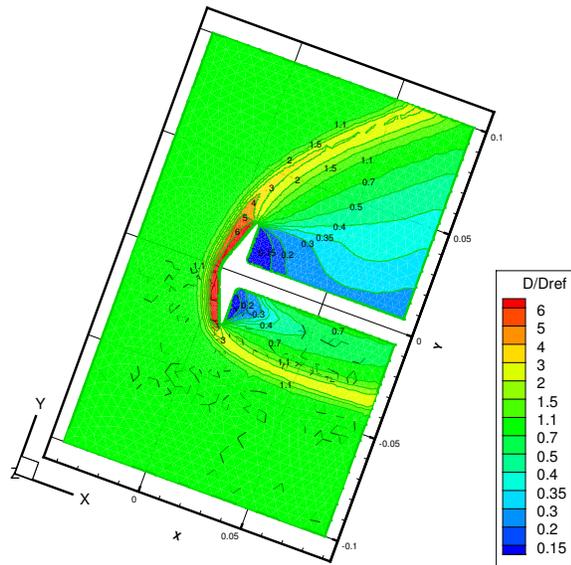


Figure 7.19 Blunt Cone (Case 2, 20° attack): Normalized density contours

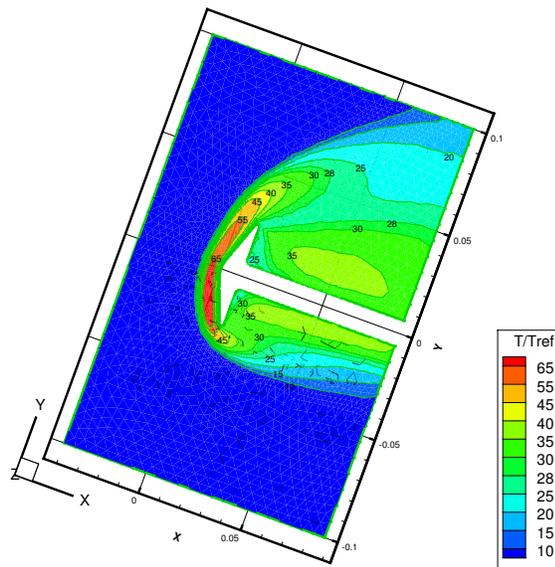


Figure 7.20 Blunt Cone (Case 2, 20° attack): Normalized temperature contours

and 20° angles of attack in Figure 7.15 - Figure 7.20. The density plot for 0° angle of attack in Case 2 is compared with the plot in Moss and Price [115], and is found to agree quite well.

Figure 7.21-Figure 7.26 plot the surface heating rates along the surface of the blunt cone. These are presented against the experimental results obtained from Allegre [118]. We can see that the results predicted match reasonably well with the experimental values. The plots also compare favorably against the heating rate plots in Moss and Price [115].

We also tabulate here, the aerodynamic coefficients for the 6 cases. Table 7.3 and Table 7.4 present the drag, lift and pitching moment coefficients for Case 1 and Case 2 respectively. These again compare well with the results from Allegre [117]. Finally we tabulate the drag coefficients for 0 attack obtained from the SR3 experiment [117], those obtained from simulation by Moss [115] and by the PDSC code in Table 7.5. We observe from the results that the values predicted by the PDSC match the experimental results more closely than the simulation results from Moss et al.

Attack Angle	C_d	C_l	C_m
0°	1.6522	0	0
10°	1.6104	-0.1624	-0.0620
20°	1.4960	-0.2931	-0.1323

Table 7.3 Aerodynamic coefficients for flow past a blunt cone: Case 1

Attack Angle	C_d	C_l	C_m
0°	1.5826	0	0
10°	1.5434	-0.1765	-0.0537
20°	1.4149	-0.3138	-0.1130

Table 7.4 Aerodynamic coefficients for flow past a blunt cone: Case 2

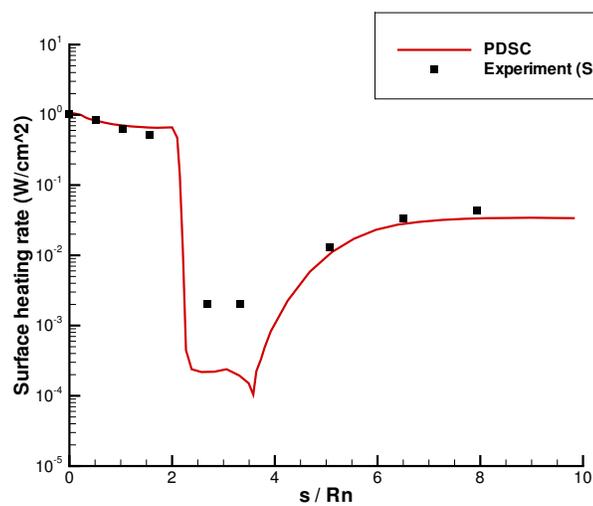


Figure 7.21 Blunt Cone (Case 1, 0° attack): Surface heating rates

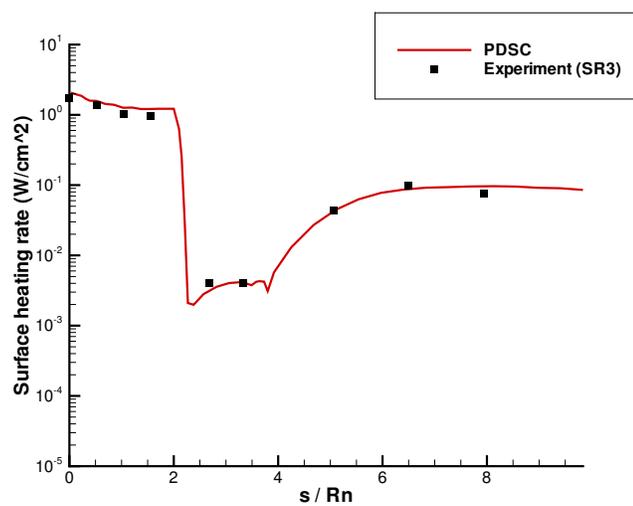


Figure 7.22 Blunt Cone (Case 2, 0° attack): Surface heating rates

Case	Moss et al	PDSC	SR3
1	1.69	1.65	1.66
2	1.62	1.58	1.52

Table 7.5 Drag coefficients from the PDSC compared against experiment and other simulations

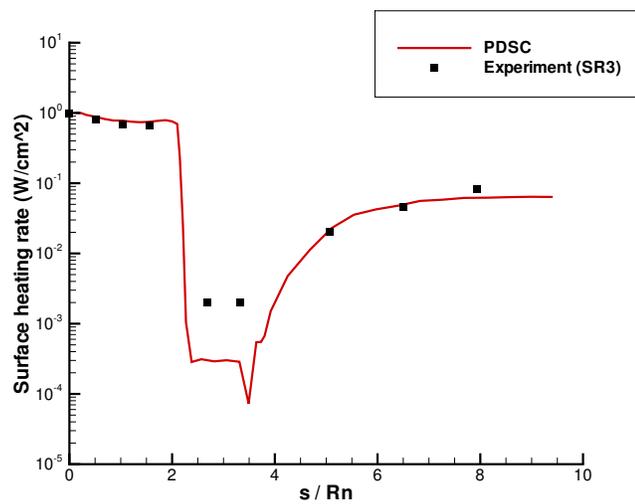


Figure 7.23 Blunt Cone (Case 1, 10° attack): Surface heating rates

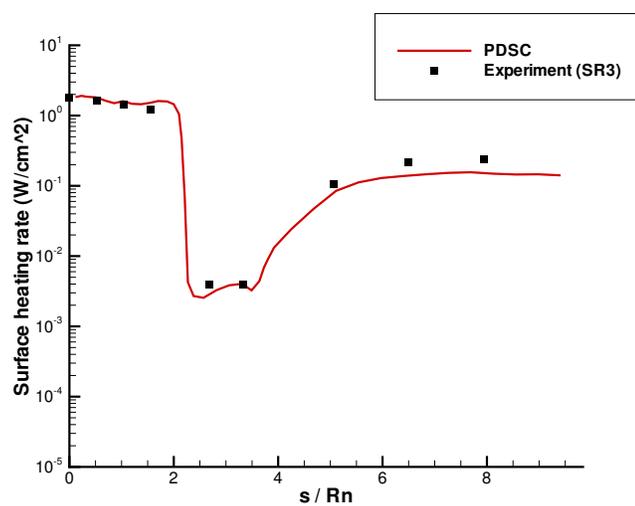


Figure 7.24 Blunt Cone (Case 2, 10° attack): Surface heating rates

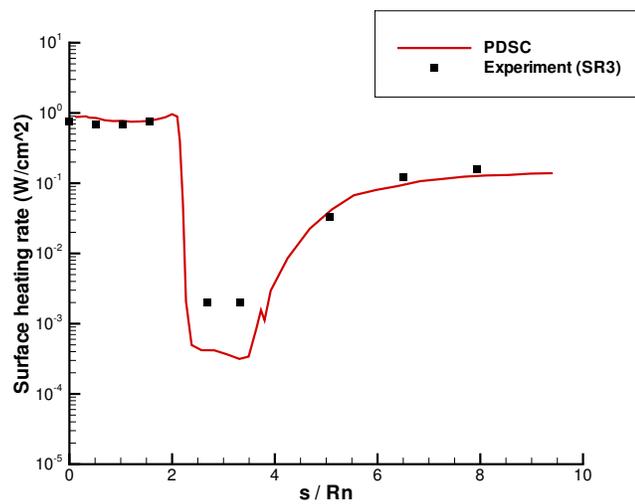


Figure 7.25 Blunt Cone (Case 1, 20° attack): Surface heating rates

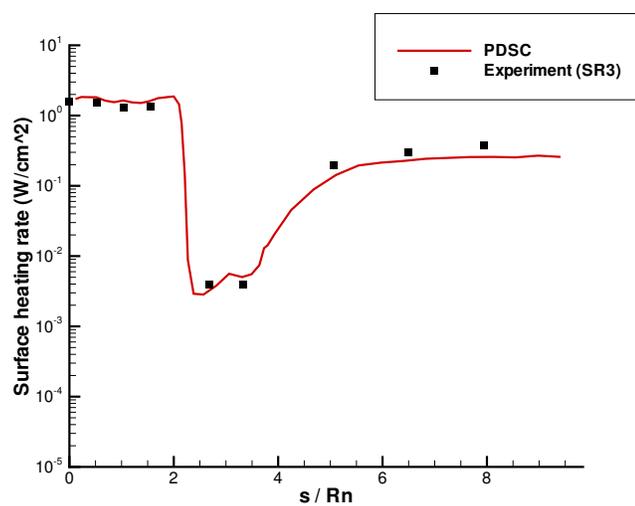


Figure 7.26 Blunt Cone (Case 2, 20° attack): Surface heating rates

CHAPTER 8

SOME RESULTS FROM PVD SIMULATIONS

8.1 Description

chapter 6 describes the features that have been built into PDSC as part of this work. The correctness testing of such features was done by putting the enhanced version of the PDSC through its paces on a series of simple, hypothetical flows. Test cases ranged from flow through multiple nozzles to expansions into a free stream. The results of these tests, though useful in verifying the implementation, will not be described here as they are not of interest in the larger objectives of this document. It should be sufficient to say that the implementations of the enhancements have been verified and the PDSC is now capable of tackling vapor deposition flows in realistic setups.

Described in this chapter are some results from a simulation study of a vapor deposition scenario described in Balakrishnan et al [1]. This flow scenario was intended as a validation exercise for the PDSC. The paper [1] describes an experimental setup for studying a Titanium vapor plume expansion in a near-vacuum, and its deposition on a substrate. The authors have also performed DSMC simulations of the same scenario. A few notable aspects of their study are discussed in some detail in chapter 3.

As mentioned before, it has been relatively difficult to find reports in literature that provide experimental measurements of the flow field properties in a PVD chamber *and* also describe the particle model chosen to study the same flows via simulation. As the publication by Balakrishnan et al provides pieces of both of these informations,

it was chosen as the starting point for validating the PDSC for PVD flows. Further simulations of other vapor deposition flows have been carried out, but have not been presented here as they are not currently in a form that can contribute to any additional inferences than the ones presented here.

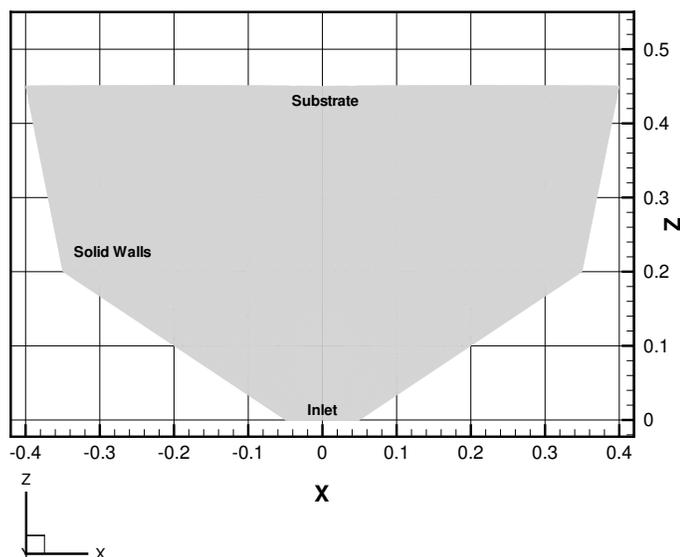


Figure 8.1 Schematic of the PVD chamber

Figure 8.1 presents a schematic of the cross-section of the PVD chamber studied by Balakrishnan et al. The vapor source is a Titanium ingot that is fed into the chamber through the bottom of the chamber. For simulation purposes, the bottom face of the domain can be treated as the inlet for the Ti vapor. The top wall of the chamber is treated as the substrate. The geometry presented here idealizes the actual setup by neglecting any outlets to vacuum pumps and the like. The schematic presented in the paper also describes the location of the laser beam equipment used for spectroscopic flow measurements.

The study measures the expansion and deposition profiles of a Ti vapor plume produced by bombarding the metal ingot with an electron beam. The electron beam causes an uneven heating of the ingot surface and results in a spatial variation of the vapor temperature and mass flux from the surface. This inlet profile is shown in

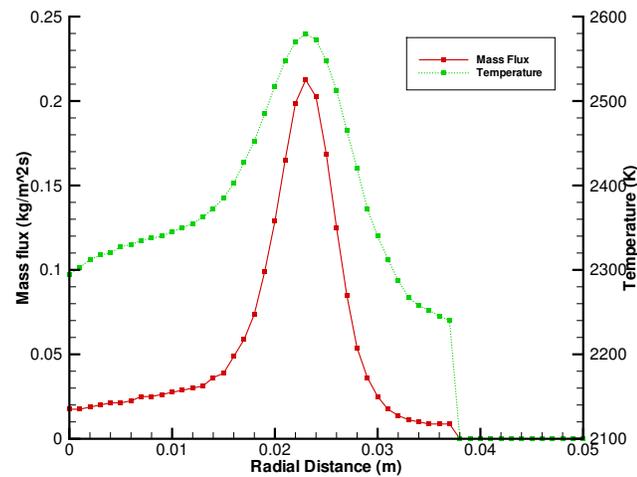


Figure 8.2 Temperature and flux profiles for the vapor plume at the inlet

Figure 8.2. The 2D DSMC simulation in Balakrishnan et al treats the flow as a single species plume expansion into a chamber, neglecting the presence of any background gas or the side effects of an electron beam in the chamber. They use a VHS collision model and also account for the electronic energies in their DSMC simulations. A summary of the relevant data from the paper is listed in Table 8.1.

Inlet diameter (m)	0.05
Target-Substrate distance (m)	0.45
Target Temperature (K)	2100-2600
Vapor mass flux ($\text{Kg}/\text{m}^2\text{s}$)	$\sim 0 - 0.23$
Number Density (near target)	$\text{O}(10^{22})$
Substrate sticking coefficient	1.0
Side wall sticking coefficient	1.0
VHS molecular diameter ($\times 10^{-26}$ m)	4.0

Table 8.1 Some relevant information from Balakrishnan et al [1]

8.2 Simulation set up

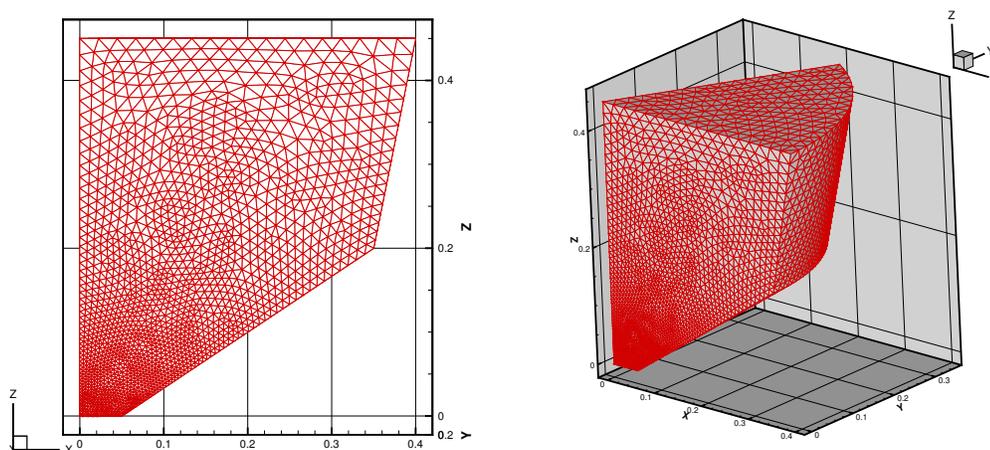


Figure 8.3 Initial grid used to start the DSMC simulations

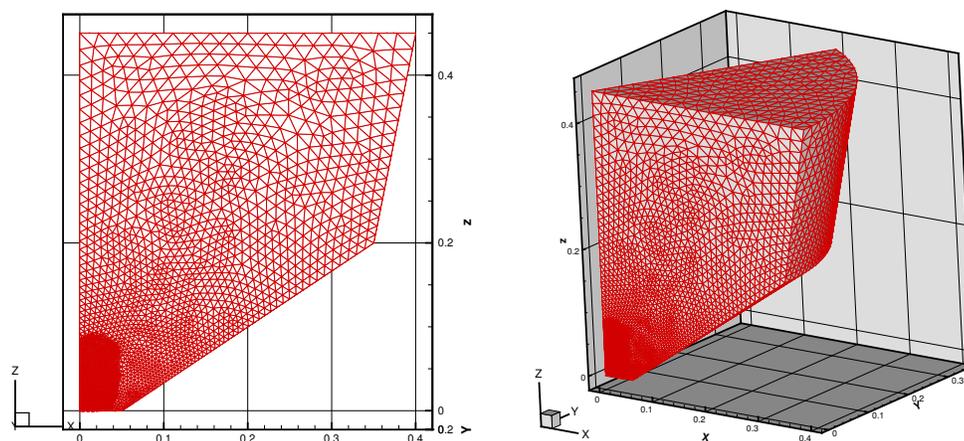


Figure 8.4 Refined grid adapted based on the initial solution

This PVD scenario was setup as a three dimensional flow domain for the PDSC. One sixth of the complete chamber was modeled in the simulation. Figure 8.3 shows the original grid used to start the simulation. It consists of about 57,000 cells and the node distribution along the different walls have been chosen to crudely mimic the mesh used in Balakrishnan et al. The included angle in the slice of the chamber in the simulation is 60° , chosen to minimize cell skew near the central axis. Refinement parameters (Table 8.3) fed to the PDSC, however, altered the mesh during

	Before Refinement	After refinement
Num. of cells	57×10^3	128×10^3
Num. of particles	2.5×10^6	5.8×10^6
Num. of time steps	180×10^3	
Num. of time steps for initial burn-in	35×10^3	
Num. of processors	64	
Execution (wall clock) time	~ 7 hours	~ 18 hours
Cluster architecture	Intel Xeon, 64 bit, 3.6 GHz	

Table 8.2 Approximate computational size before and after mesh refinement

Refinement parameters	
Threshold min. freestream density	0.1
Threshold max. cell-local Knudsen number	1.5

Table 8.3 Refinement parameters

the simulation based on an initial solution, and the refined mesh is shown in Figure 8.4. Table 8.2 compares relevant numbers for the direct simulation stages before and after mesh refinement.

8.3 Observations of the chamber flow

Figure 8.5 presents the density contours obtained from the PDSC. It should be noted that the density falls by more than three orders of magnitude as the plume expands through the chamber. This is reflective of the steep flow gradients involved and the challenges it poses to conventional flow solvers. Similar drops in density were observed in the scenario of flow expanding from an orifice into vacuum (section 7.2). Figure 8.6 and Figure 8.7 present flow contours for the translational temperature and the absolute velocity through the domain. Both of these figures conform the presence of steep flow gradients. The off-axis peak in the absolute velocities must be noted.

This can be explained as caused by the inlet flow property profiles. The number and temperature of the vapor atoms entering the domain peak off the central axis and this causes greater acceleration, which is reflected as greater flow velocities in these regions. The contours presented here agree well with the simulations by Balakrishnan et al [1].

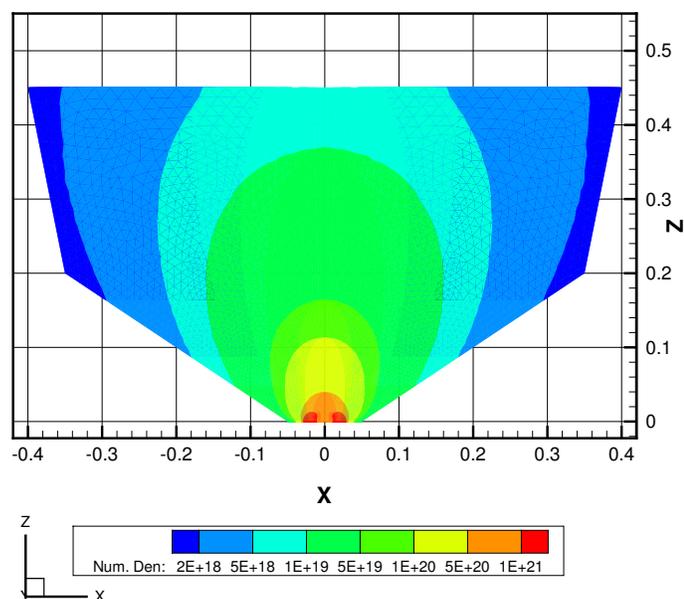


Figure 8.5 Density contours

Also of some interest is the observation that the vapor temperature drops drastically while the macroscopic flow speed increases from 0 m/s at the inlet to 1300+ m/s near the substrate. This can be explained as follows. The temperature of the vaporizing atoms represents the random thermal energy of the atoms as they vaporize off the inlet surface. When they vaporize, they do not have any macroscopic directional speed, i.e. the macroscopic velocity is 0m/s. However they have significant thermal speeds (which is the random component of the molecular velocities). As they undergo collisions the vapor plume expands into the chamber. Because there is a constant influx at the inlet, the collisions ensure that most atoms are accelerated into the chamber. Hence, farther into the chamber most of the molecular velocities are directed likewise and hence the major component of the molecular velocities are

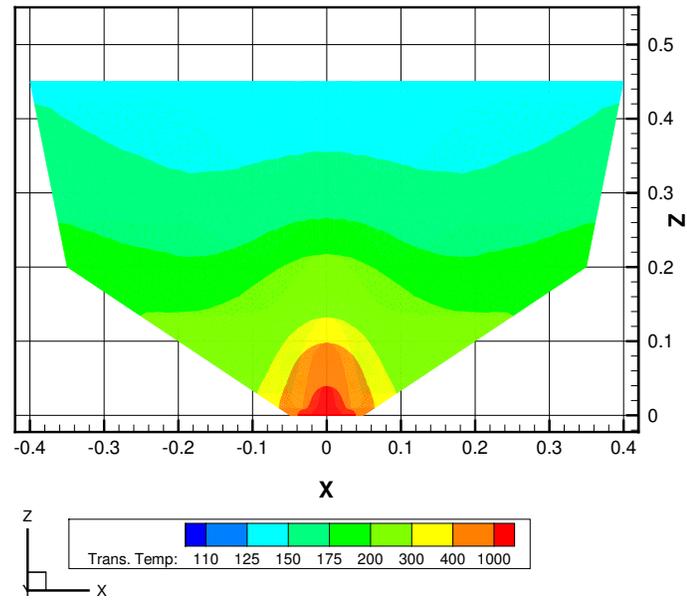


Figure 8.6 Temperature contours

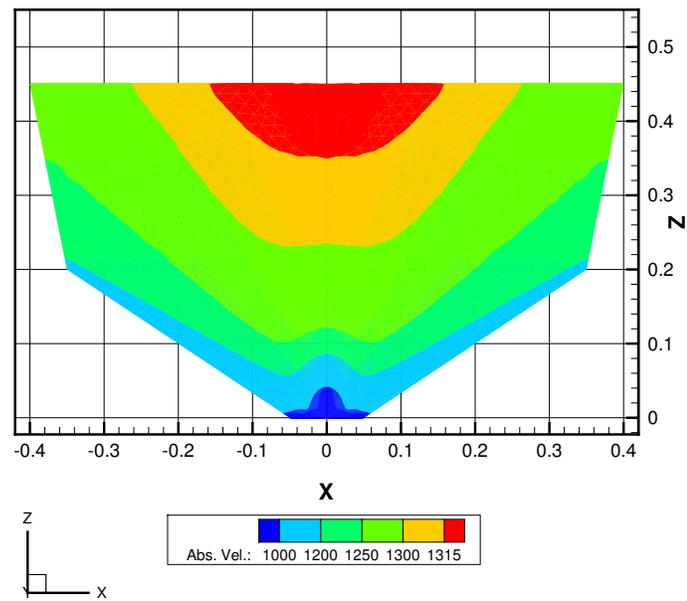


Figure 8.7 Absolute velocity contours

interpreted as the macroscopic flow velocity. Hence we observe a steep increase in the flow velocity and a sharp decrease in the flow temperature, though a major reason for this is simply a reinterpretation of the molecular energies into different macroscopic properties. There is no steep decrease in the translational energy in the system but just a redistribution of this energy between the collective component (flow speed) and the random component (thermal speed). Hence, temperature, which is an index of the random thermal energy, decreases; while flow velocity, which is an average of the individual molecular velocities, increases.

Source	V_{abs} (m/s)
Experimental	1400 ± 10
PDSC	1321.7
Balakrishnan et al (without electronic energy)	1326
Balakrishnan et al (with electronic energy)	1393

Table 8.4 Predictions of absolute velocity compared against experiment

Table 8.4 compares the absolute velocities predicted by the PDSC against experimental measurements and also against DSMC predictions by Balakrishnan et al [1]. The location of these measurements is evident from the experimental schematic that can be found in their paper. The authors compare DSMC predictions made with and without the inclusion of atomic electronic energies in the simulation. These numbers have been reproduced in the table for reference. It can be seen that the PDSC-predicted absolute velocities are about as close to the experimental measurements as those computed by Balakrishnan et al without the inclusion of electronic energies in the simulation. The PDSC's estimate differs from experimental values by about 5-6%. Apart from the fact that the PDSC does not account for energies in the atom electronic states, one reason for this difference could be that the values are obtained using collisional model parameters that may be different from those used by the authors of the reference paper. Complete information about the VHS model

parameters that were used have not been provided; hence the parameters that have not been reported have been obtained from a later publication by the same group [24]. This could result in a different collisional environment, affecting the rate of expansion (and hence, the velocities) of the plume.

8.4 Wall and Substrate fluxes

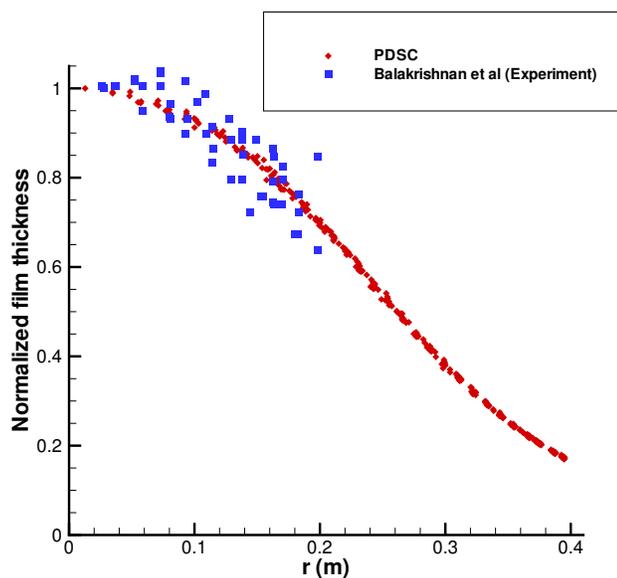


Figure 8.8 Normalized deposition profile

Figure 8.8 plots the deposition profile along the substrate from experiment and simulation, normalized against their respective axial deposition thicknesses. The predictions from the PDSC were in the form of incident fluxes on the substrate surface. As the surfaces were treated to be perfectly sticking, these values are directly proportional to the deposition thickness. Hence, when normalized against a reference value (the incident flux at the substrate center), they yield a curve identical to the one that may be obtained from a computed deposition profile. The values obtained from the PDSC are face-averaged properties plotted at the corresponding face centers. A

radial profile is obtained by simply collapsing these face-averaged properties onto the substrate radius.

It can be seen that there is a good agreement with experiment and with simulations by Balakrishnan et al. It is important that the PDSC's ability to make such predictions be verified as it is one of the ultimate objectives of such vapor-phase simulations. We can see that the profile predicted by the PDSC lies within the range of experimental measurements and also agrees with similar simulations without electronic energies [1].

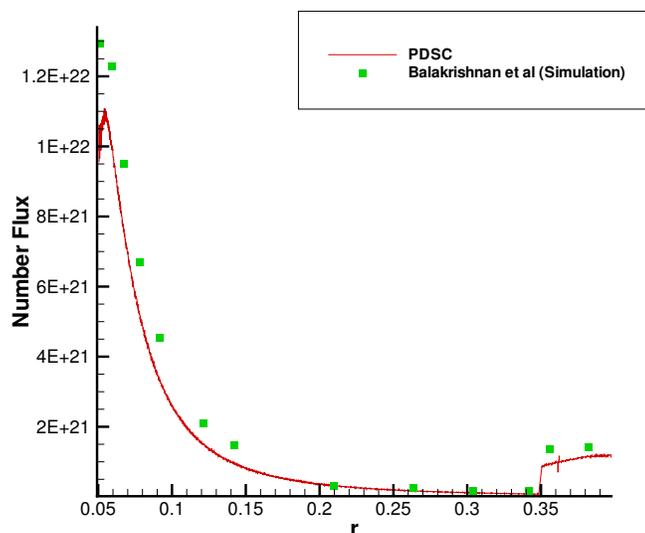


Figure 8.9 Fluxes to the side walls

Figure 8.9 compares the number fluxes at the side walls obtained from the PDSC and the simulations in the reference publication. Both the simulation attempts predict quite similar wall fluxes. There is some difference close to the inlet where the PDSC predicts lower fluxes. This can probably be explained by the differences in the particle model parameters. The results plotted are from a simulation where the PDSC used the VHS model parameters for Ti as proposed by Fan et al [24]. These parameters present an increased collision cross-section for Ti at the reference temperature. Assuming Balakrishnan et al used similar values for ω , this would

mean that the collision cross-sections in the PDSC were somewhat larger resulting in a greater collisional flow. If the assumption that the simulations differed in just the collision cross-section values is true, then the marginally lower wall fluxes predicted by the PDSC can be attributed to slightly greater plume focusing due to increased collisional transport. As the collisional processes are most dominant near the inlet, the focusing effect is also the greatest in this region. Hence, the difference between PDSC's predictions and those from the reference publication are the greatest here. There are several publications that talk about vapor-plume focusing effects due to collisional transport [68, 1].

8.5 Verifying the Implementation

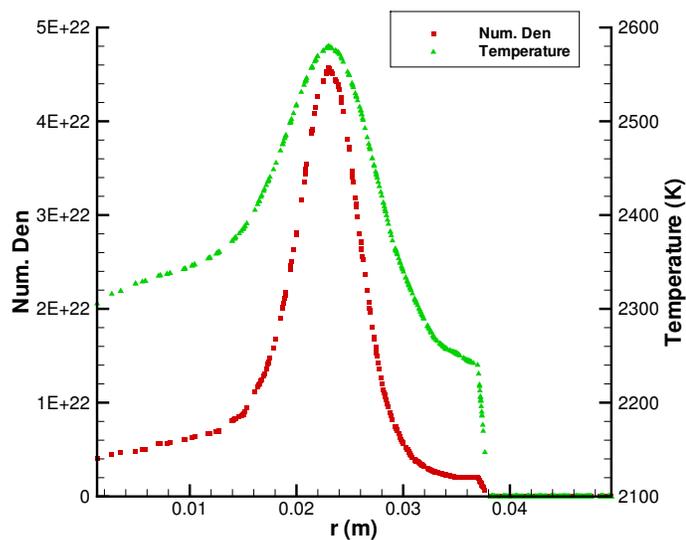


Figure 8.10 Inlet flow profile generated by the PDSC preprocessor

Figure 8.10 - Figure 8.12 provide a quick verification of the new features added to the PDSC. Figure 8.10 plots the number density and temperature profiles generated by the initial, sequential portion of the PDSC and fed to the parallel, direct

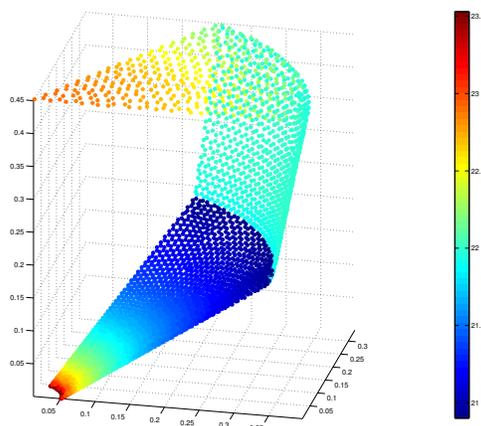


Figure 8.11 Flux incident on the chamber surfaces

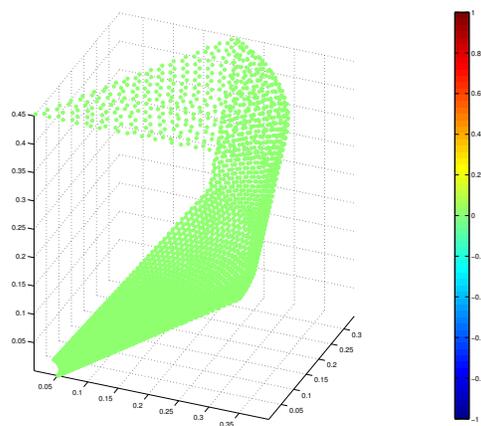


Figure 8.12 Flux reflected from the chamber surfaces

simulation module. As all inlet profile definitions are handled at this stage, the face-based data structure was written out as an intermediate result to verify if the profile definition was used correctly and the interpolations schemes performed well. The plots conform that the temperature profile is identical to the input and the number densities (directly proportional to the mass flux) retain the sharp off-axis peak. The plots presented here are only a glimpse of more rigorous checks that have been carried out.

Figure 8.11 and Figure 8.12 plot the variation of the incident and reflected fluxes along the substrate and the chamber walls. These plots are presented just as a quick check that the surfaces do behave as perfectly sticking surfaces. It can be seen that the incident fluxes (plotted on a logarithmic scale) vary as shown in the earlier graphs whereas the reflected fluxes are zero throughout the solid surfaces of the chamber. This indeed proves that the solid walls have been perfectly sticking for the vapor atoms.

8.6 Sensitivity Studies

It may be evident from the discussion so far that the results presented have been obtained under two major assumptions that affect the collision processes in the simulation. The first is the actual VHS model that has been used for Ti vapor. The model parameters supplied in the reference paper are not complete, and hence, two different sets of model parameters have been used in the simulations in this work. This will definitely influence the results and hence, a study is performed to compare the results obtained by comparing the results from the two models. The models differ primarily in the atomic diameter of Ti and, hence, the collision cross-section of the vapor atoms.

The second assumption is that the substrate and chamber walls are perfectly sticking surfaces. Though this does mimic the assumptions in the reference paper [1], it can also be reasoned that this provides an unphysical idealization of the actual scenario. Actual solid surfaces, may be very close to perfectly sticking, but the small fractional deviation from this perfect behavior might affect the flow field noticeably. A study has been done to verify this by altering the sticking coefficient of the surfaces and comparing the changes in the flow contours.

It should be noted that all the results presented in this section are obtained without correcting for backscatter. This will result in a lesser mass flux entering the simulation domain, but will not have any bearing on this comparative study.

8.6.1 Collision Cross-Section

Reference	Balakrishnan et al [1]	Fan et al [24]
Molecular diameter ($\times 10^{-26}$)	4.0	5.844
Reference temperature (K)	273.0	2000
Temperature exponent of viscosity	0.849	0.849
Surface sticking coefficient	1.0	1.0

Table 8.5 A comparison of Ti VHS model parameters used in the sensitivity study

The two sets of VHS model parameters used are listed in Table 8.5. The two models differ mainly in the atomic collision-cross section. The model parameters in Fan et al should result in increased collisional flow in the chamber. Figure 8.13 shows the density contours that are obtained using the two models. It can be seen from the density contours that an increased collision cross-section causes an increase in the flow densities near the central axis, while it causes lower densities farther from the axis (near the side walls). This means that the vapor plume is focused towards the central axis causing greater substrate fluxes near the central axis. A closer look at the figure will show that the plume contours for the case with the increased cross-section are indeed slightly flatter on the sides. Such a counter-intuitive focusing effect due to increased collisional transport has been observed and studied in other efforts documented in literature [68].

Proof of increased collisional transport can be seen in Figure 8.14 which plots the backscatter fluxes at the inlet surface normalized against, the backscatter flux at the inlet center for the base case (Ti VHS dia. 4.0\AA). It is immediately apparent that the increased cross-section greatly increases the backscatter fluxes which is a logical outcome of assuming increased collisional activity. Of further interest is the significant increase in backscatter. It must be noted that the VHS diameters used more than double the collision cross-section of the vapor.

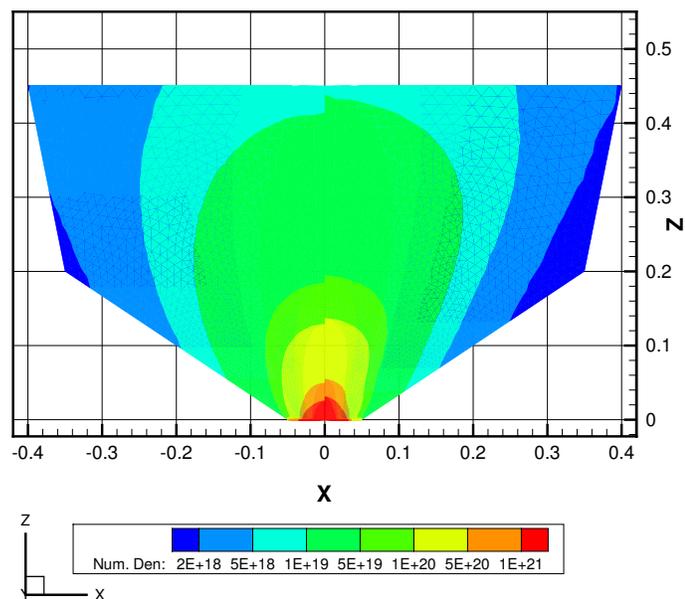


Figure 8.13 Sensitivity to VHS molecular diameter: Density contours. Left - $d = 4.0\text{\AA}$ (Balakrishnan et al); Right - $d = 5.844\text{\AA}$ (Fan et al)

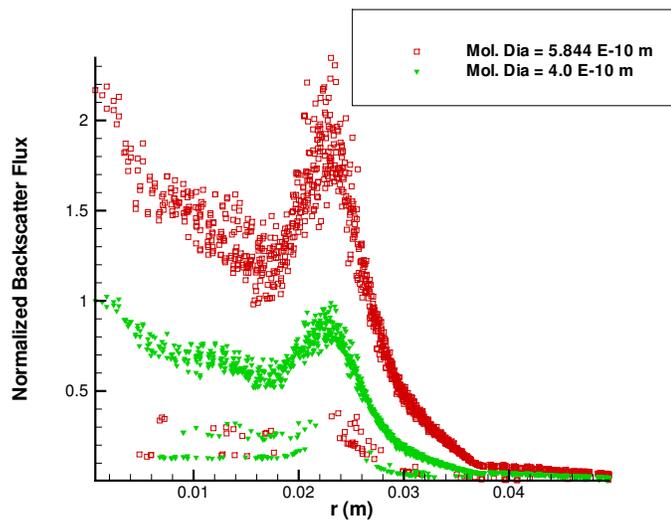


Figure 8.14 Sensitivity to VHS molecular diameter: Normalized Backscatter Fluxes. Left - $d = 4.0\text{\AA}$ (Balakrishnan et al); Right - $d = 5.844\text{\AA}$ (Fan et al).
Normalized against value at inlet center for base case ($d=4.0\text{\AA}$).

This comparison points to the fact that model parameters can affect the relative distribution of fluxes within the chamber. Accurate predictions of deposition profiles are therefore critically dependent on the particle collision models used and the choice of model parameters that best mimic actual collisional processes in real systems.

8.6.2 Sticking Coefficient

As has been mentioned earlier in this section and in section 6.5, perfectly sticking surfaces represent an unphysical but necessary approximation. The necessity arises because of the unavailability of literature that provides rigorous theoretical or experimental treatments of gas-surface interactions in the context of high temperature metal vapors. Thermal accommodation, though unrelated to sticking, also falls under the umbrella of gas-surface interactions, and has been shown to be complete for most gas-surface interactions of practical interest [119, 32]. However, similar conclusions for the sticking behavior of surfaces are not available. The results presented here attempt to highlight the sensitivity of the chamber flow phenomena to the gas-surface interactions occurring at the chamber walls. To somewhat emphasise the need for further exploration in this area, the two scenarios considered compare the flow contours for simulations that used sticking coefficients of 1.0 and 0.9. The decision to study a 10% variation in the sticking coefficient does not arise out of any qualitative / quantitative knowledge of surface behaviors, but rather out of the desire to study the effect of a non-trivial change in surface behavior on the chamber flows.

Figure 8.15, Figure 8.17 and Figure 8.19 present results for a change in the sticking coefficient for an atomic diameter of 4.0\AA [1], while Figure 8.16, Figure 8.18 and Figure 8.20 present the contours obtained using the VHS model parameters in Fan et al [24]. A point of note is that the mesh in the vicinity of the walls is relatively coarse and results in somewhat coarse contours. Finer meshes might yield better resolutions.

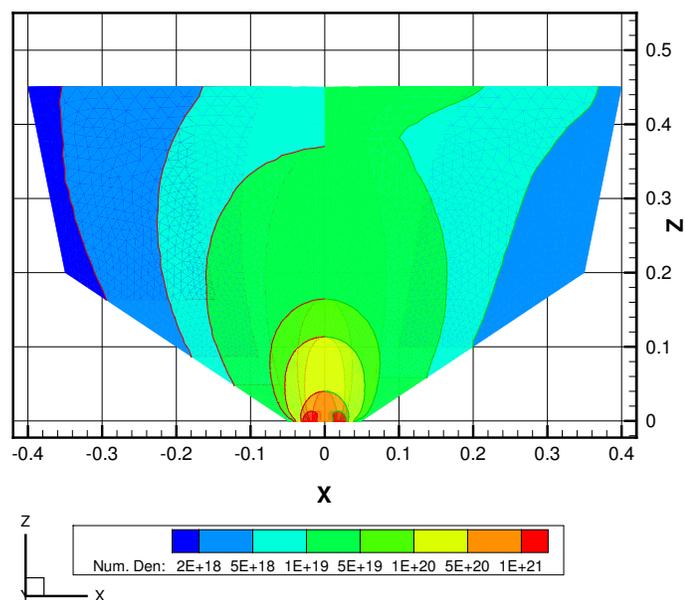


Figure 8.15 $D=4.0\text{\AA}$; Density contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9

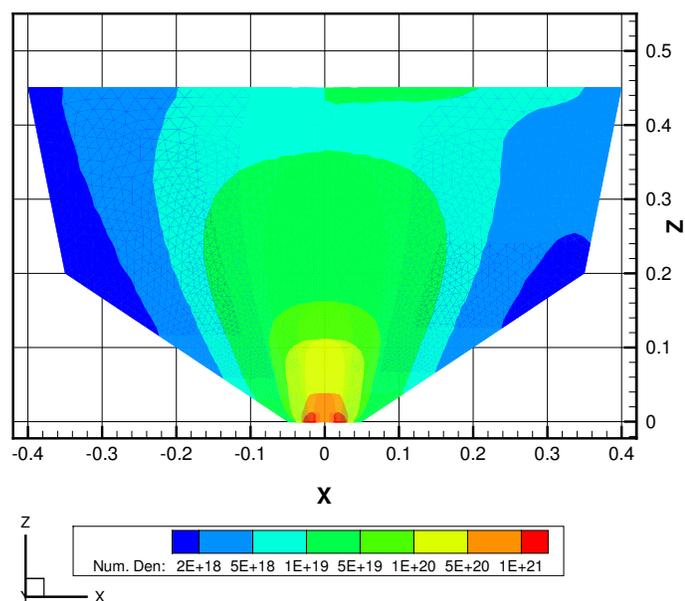


Figure 8.16 $D=5.844\text{\AA}$; Density contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9

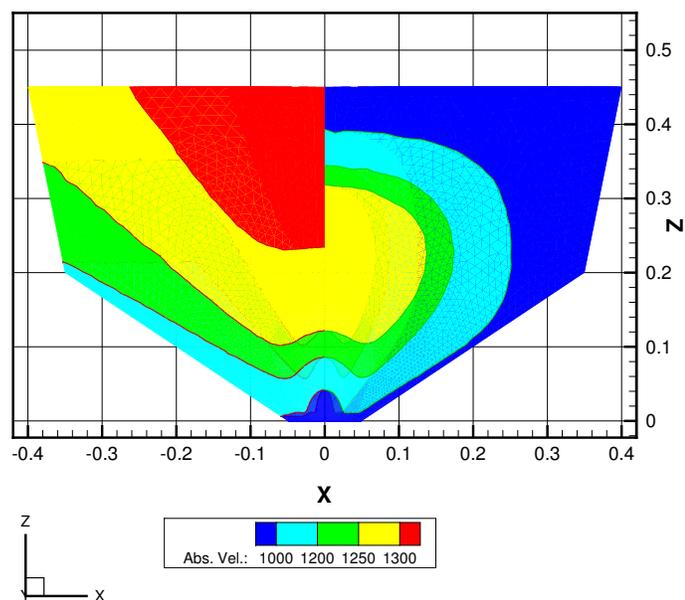


Figure 8.17 $D=4.0\text{\AA}$; Velocity contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9

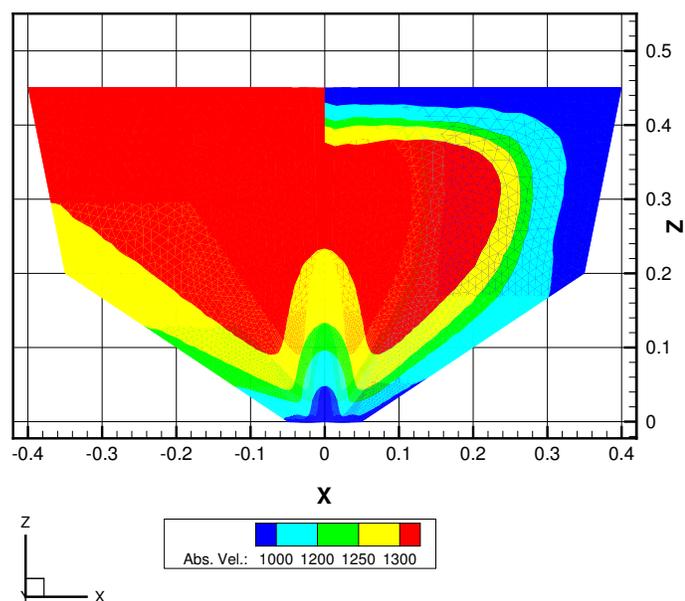


Figure 8.18 $D=5.844\text{\AA}$; Velocity contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9

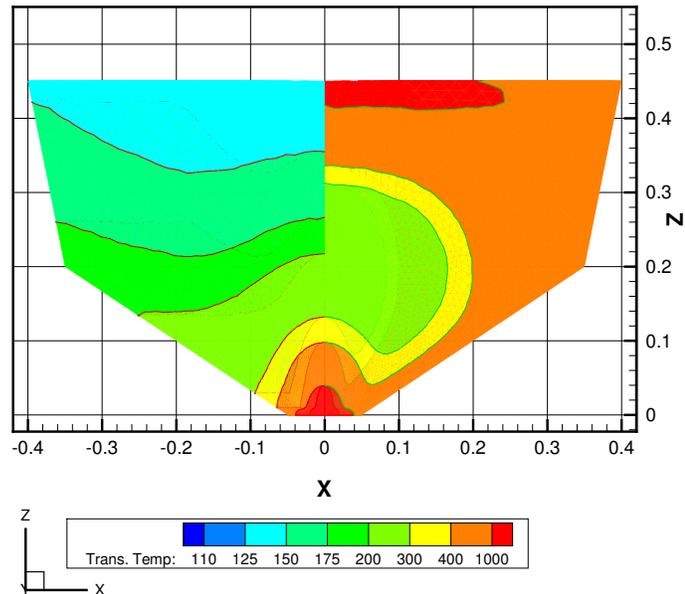


Figure 8.19 $D=4.0\text{\AA}$; Temperature contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9

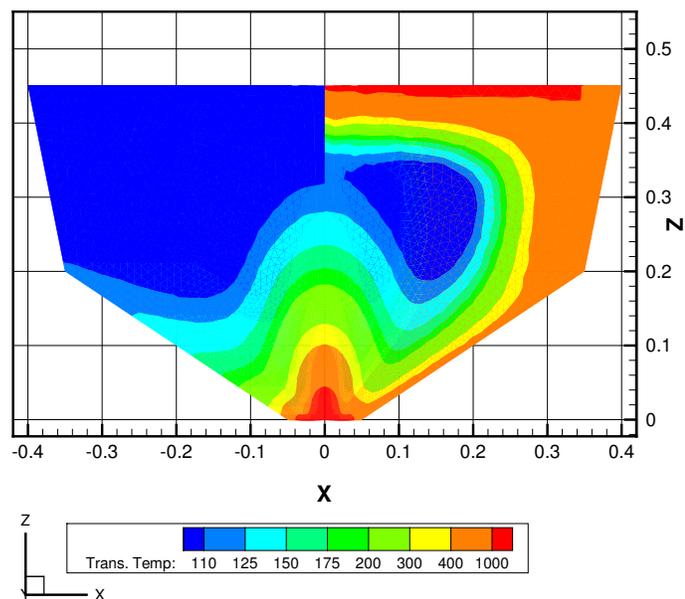


Figure 8.20 $D=5.844\text{\AA}$; Temperature contours: Left - Sticking coefficient 1.0; Right - Sticking Coefficient 0.9

One conclusion that is immediately apparent is that the sticking coefficient significantly affects the flow field in the vicinity of the chamber walls. This is quite understandable, as a perfectly sticking surface does not give any *feedback* to the gas phase and hence, its presence is effectively immaterial to the flow. The contours for perfectly sticking surfaces are identical to what will be obtained for a flow expanding into vacuum. A partially sticking coefficient will reflect some of the incident particles back into the gas phase. The reflected particles possess energies that are commodated towards the surface temperature. The gas phase hence gets some *feedback* from the surface. The flow contours are hence affected by the surface behavior. The density contours for the cases where sticking coefficient is 0.9 show an increase in the flow density near the chamber walls. As partially sticking surface reflect some of the particles back into the surface, this causes an increase in density near the walls. The reflected particles will also cause a net reduction in the flow velocity in the chamber. This is also evident from the flow velocity contours in Figure 8.17 and Figure 8.18. As the surface now emits some particles back into the gas phase, this leads to increased collisional flow near the surface and, hence an increase in flow temperatures evident from Figure 8.19 and Figure 8.20. Another approach to explaining this is the discussion of how the molecular velocities are distributed between the macroscopic flow descriptors - temperature and flow velocity. Reflected particles cause greater random thermal velocities in the vicinity of the substrate, hence, a greater temperature. The molecules near the solid walls now possess smaller likewise velocity components and hence the flow velocity is lower.

CHAPTER 9

SUGGESTIONS FOR FUTURE WORK

Just like in section 1.2, the avenues that warrant further exploration and work are listed below, classified primarily into methodology, implementation and computational aspects.

9.1 Methodology

- **Collision Models and Model Parameters:** As mentioned before, modeling vapor deposition processes requires non-continuum models (usually particle models) for the gas phase transport. In PVD processes, it is common to have multiple species interacting during the gas phase transport. Typically, one or more film component species, background or carrier gas atoms, ions, and reactive species like oxygen are present. It is of paramount importance that appropriate particle models be chosen to handle all these scenarios. The task of defining models that can account for interactions between particles of disparate masses and energies is still open to contributions. One of the biggest avenues for further work is in identifying adequate collision model parameters for describing the collisional transport phenomena involving high temperature metal vapors. Even for the widely used VHS model, sufficient data or literature is not available to make the choice of model parameters a trivial task. This work has used model parameters based on approximate quasi-theoretical derivations by Fan et al [24]. There exists a need for verifying this approach by applying it to further PVD test scenarios.

- **Gas-Surface Interactions:** Like inter-particle collisions, modeling surface-particle collisions are also critical to accurate simulations. Available PVD simulation literature uses perfectly sticking surfaces almost universally. No experimental or theoretical justifications for such an unphysical idealization have been encountered in literature. Even if there exist arguments that surfaces are almost perfectly sticking for metal vapor atoms, sticking coefficients that differ only slightly from unity may influence flow fields noticeably. This has been shown in sensitivity studies by Balakrishnan et al [1] and in this work subsection 8.6.2. Balakrishnan et al alter the sticking coefficient by about 5% from perfectly sticking and this work has studied the effects of a 10% reduction in the sticking coefficient. The results presented underline the need for more investigation into further work in this direction.

9.2 Implementation and Validation

- **Mass-flux based inlet conditions:** The bulk of the literature presenting results of experimental studies of vapor deposition processes presents data in terms of mass fluxes at the vapor source. Even in scenarios of uneven heating on the vapor source (ingot) surfaces, data available is usually in the form of a spatial variation of the mass fluxes along the surface. In this light, it will be advantageous to have the ability to specify inlet conditions in the PDSC that can directly specify the mass flux along an inlet. Current implementations require that an inlet number density be specified. This is obtained by computing the number density of an equilibrium gas that (if present at the inlet surface) would result in the experimentally observed mass fluxes. This process is something that can be easily built into the code as an additional boundary condition capability.

- **Background gases:** Background gases can play an important role in affecting the vapor plume by increasing collisional transport. Simulations are under way to observe PDSC's predictions of scenarios involving background gases. Such case may include simulations that This is work for the immediate future.
- **Solid wall inlets:** It will be interesting to implement a solid wall inlet boundary type that would act as a solid boundary for particles in the domain, but also act as a mass flux source. This would very closely mimic the surface of the ingots that provide metal vapors PVD processes. Such a representation will also cause the ingot surface temperatures to be influenced by the backscatter and other gas phase processes. It will be interesting to test whether such feedback can cause noticeable differences in flow field predictions.

9.3 Computational and Numerical Considerations

- **Random numbers:** Random number generation plays a very pivotal albeit underemphasized role in success of the DSMC approach. The current implementation samples from a single uniform pseudo-random number sequence for all its random number needs. The implementation has been extensively validated and used to simulate 'large' flow problems with good results. This does suggest that this choice of using a single random number sequence does not introduce any artificial correlation between distributions that are meant to be independent of each other. However, it will be interesting to study whether the pseudo-random sequence is '*recycled*' during the course of a long simulation, and if it is, whether it has any influence on the flow predictions.
- **Generating inlet fluxes:** The typical approach used by most DSMC implementations to introduce simulation particles into the domain at the inlets, is to use the Accept-Reject (AR) algorithm where a target distribution is simulated by sampling from uniform distributions and then rejecting a fraction of these

samples based on appropriate criteria. The rejection rate resulting from such an algorithm has not been studied. The dependence of this rejection rate on the nature of the flow problem is also not known. The chances that the fraction of rejected samples may depend on implementation and algorithm specifics is also quite high. In simulations where the number of particles required to enter the flow domain at each time step is considerable, this rejection rate during the AR algorithm might affect overall performance. A quick search of literature shows that attempts to investigate this question are not numerous. Lilley et al [120] have made attempts to address this with some improvement in computational efficiency for upstream boundaries. The possibility of investigating this question certainly has some merit but may not guarantee any significant effect on performance.

- **Scalability:** Computational efficiency is definitely a big concern for typical DSMC implementations. Parallelization can help tackle larger simulation problems but in the process, increases the challenges involved in developing and tuning efficient implementations. The current implementation (PDSC) has been shown capable of simulating large flow problems with domains involving $O(10^7)$ particles in meshes containing $O(10^5)$ cells. However crude performance analyses indicate significant room for improvement in performance. As a case in point, the simulations for the vapor deposition scenarios presented in chapter 8 typically spent about 25-33% of the execution time in inter-processor synchronization and data transfer. As the current implementation uses blocking communication patterns, a sizeable portion of the communication time is spent as idle processor time. Logging and real-time simulation tracking requirements also necessitate significant disk I/O. Optimization exercises should ideally involve rigorous profiling of the code to identify communication-bound and disk-bound bottlenecks. This is definitely an avenue for future work with a sufficiently reasonable guarantee of improved performance.

LIST OF REFERENCES

- [1] J. Balakrishnan, I. D. Boyd, D. G. Braun, Monte Carlo simulation of vapor transport in physical vapor deposition of titanium, *Journal of Vacuum Science & Technology, Part A: Vacuum, Surfaces and Films* 18 (3) (2000) 907 – 916.
- [2] C. F. Powell, J. H. Oxley, J. M. B. Jr (Eds.), *Vapor deposition*, Wiley (New York), 1966.
- [3] K. L. Chopra, I. Kaur, *Thin Film Device Applications*, Plenum Press (New York), 1983.
- [4] J. W. Dini, Properties of coatings: Comparisons of electroplated, physical vapor deposited, chemical vapor deposited, and plasma sprayed coatings, *Materials and Manufacturing Processes* 12 (3) (1997) 437 – 472.
- [5] Quinto, U. T. Dennis (Kennametal Inc. Greensburg, PA, G. J. Wolfe, P. C. Jindal, High temperature microhardness of hard coatings produced by physical and chemical vapor deposition, *Thin Solid Films* 153 (1987) 19 – 36.
- [6] J. Zhao, E. G. Garza, K. Lam, C. M. Jones, Comparison study of physical vapor-deposited and chemical vapor-deposited titanium nitride thin films using X-ray photoelectron spectroscopy, *Applied Surface Science* 158 (3) (2000) 246 – 251.
- [7] C. C. Chi, R. B. Dover (Eds.), *High-Tc Superconductivity: Thin Films and Applications*, SPIE, Washington, 1990.
- [8] T. Suzuki, H. Umehara, Pitch-based carbon fiber microstructure and texture and compatibility with aluminum coated using chemical vapor deposition, *Carbon* 37 (1999) 47 – 59.
- [9] K. A. Appiah, Z. L. Wang, W. J. Lackey, Characterization of interfaces in C fiber-reinforced laminated C-SiC matrix composites, *Carbon* 38 (2000) 831 – 838.
- [10] T. F. Russell, B. N. Baron, S. C. Jackson, R. E. Rocheleau, *Microelectronics Processing: Chemical Engineering Aspects*, No. 221 in *Advances in Chemistry Series*, American Chemical Society, 1989, Ch. Physical vapor deposition reactors, pp. 171 – 198.

- [11] M. Amon, A. Bolz, B. Heublein, M. Schaldach, Coating of Cardiovascular Stents with Amorphous Silicon Carbide to reduce Thrombogenicity, in: Proceedings of the 16th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 1994, pp. 838 – 839.
- [12] B. O'Brien, C. Chandrasekaran, Development of iridium oxide as a cardiovascular stent coating, *Advanced Materials and Processes* 162 (11) (2004) 82 – 84.
- [13] M. H. Fathi, M. Salehi, A. Saatchi, V. Mortazavi, S. B. Moosavi, In vitro behavior of bioceramic, metallic and bioceramic-metallic coated stainless steel dental implants, *Dental Materials* 19 (2003) 188 – 198.
- [14] M. H. Fathi, M. Salehi, A. Saatchi, V. Mortazavi, S. B. Moosavi, Novel double layer HydroxyApatite (HA)/Ti Coating for Biocompatibility improvement of metallic implants, *Surface Engineering* 17 (6) (2001) 459 – 464.
- [15] J. R. Nicholls, K. J. Lawson, D. S. Rickerby, P. Morrell, Advanced processing of TBCs for reduced thermal conductivity, in: Proceedings of the 85th Meeting of the AGARD Structures and Materials Panel, Aalborg, Denmark, 1997, pp. 61 – 69.
- [16] H. Mizuseki, Y. Jin, Y. Kawazoe, L. T. Wille, Growth processes of magnetic clusters studied by Direct Simulation Monte Carlo method, *Journal of Applied Physics* 87 (9) (2000) 6561 – 6563.
- [17] W. Koenig, R. Fritsch, D. Kammermeier, Physically vapor deposited coatings on tools. performance and wear phenomena, *Surface & Coatings Technology* 49 (1-3) (1991) 316 – 324.
- [18] S. Gulizia, M. Jahedi, E. Doyle, Performance evaluation of pvd coatings for high pressure die casting, *Surface and Coatings Technology* 140 (3) (2001) 200 – 205.
- [19] L. Holland, *Vacuum deposition of thin films*, London, Chapman and Hall, 1970.
- [20] W. Kern, J. L. Vossen, *Thin film processes*, Academic Press (New York), 1978.
- [21] S. M. Rossnagel, J. Hopwood, Metal ion deposition from ionized magnetron sputtering discharge, *Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures* 12 (1) (1994) 449 – 453.
- [22] H. Han, J.-Z. Zhang, B. Halpern, J. Schmitt, J. del Cueto, Jet vapor deposition of transparent conductive zno:al thin films for photovoltaic applications, *Materials Research Society Symposium - Proceedings* 426 (1996) 491 – 496.
- [23] H. Wadley, L. Hsiung, R. Lankey, Artificially layered nanocomposites fabricated by jet vapor deposition, *Composites Engineering* 5 (7) (1995) 935 – 950.
- [24] J. Fan, I. D. Boyd, C. Shelton, Monte Carlo modeling of electron beam physical vapor deposition of yttrium, *Journal of Vacuum Science & Technology, Part A: Vacuum, Surfaces and Films* 18 (6) (2000) 2937 – 2945.

- [25] X. Xi, A. Pogrebnyakov, S. Xu, K. Chen, Y. Cui, E. Maertz, C. Zhuang, Q. Li, D. Lamborn, J. Redwing, Z. Liu, A. Soukiassian, D. Schlom, X. Weng, E. Dickey, Y. Chen, W. Tian, X. Pan, S. Cybart, R. Dynes, MgB₂ thin films by hybrid physical-chemical vapor deposition, *Superconductivity and its Applications* 456 (1-2) (2007) 22 – 37.
- [26] B. Rother, G. Ebersbach, H. Gabriel, Substrate-rotation systems and productivity of industrial pvd processes, *Surface and Coatings Technology* 116-119 (1999) 694 – 698.
- [27] A. Pflug, M. Siemers, B. Szyszka, Parallel dsmc gasflow simulation of an in-line coater for reactive sputtering, in: *Lecture Notes in Computer Science (Proceedings of the 13th European PVM/MPI User's Group Meeting)*, Vol. 4192 LNCS, 2006, pp. 383 – 390.
- [28] D. D. Hass, Y. Marciano, H. N. G. Wadley, Physical vapor deposition on cylindrical substrates, *Surface and Coatings Technology* 185 (2004) 283 – 291.
- [29] S. M. Rossnagel, Directional and preferential sputtering-based physical vapor deposition, *Thin Solid Films* 263 (1) (1995) 1 – 12.
- [30] D. L. Smith, *Thin Film Deposition: Principles and Practice*, McGraw-Hill (New York), 1995.
- [31] W. L. Wang, I. D. Boyd, Predicting continuum breakdown in hypersonic viscous flows, *Physics of Fluids* 15 (1) (2003) 91 – 100.
- [32] G. A. Bird, *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford, UK: Clarendon Press, 1994.
- [33] G. A. Bird, Monte carlo simulation of gas flows, *Annual Review of Fluid Mechanics* 10 (1) (1978) 11–31.
- [34] J. K. Haviland, M. L. Lavin, Application of the Monte Carlo Method to Heat Transfer in a Rarefied Gas, *The Physics of Fluids* 5 (11) (1962) 1399 – 1405.
- [35] C. C. Battaile, D. J. Srolovitz, Kinetic monte carlo simulation of chemical vapor deposition, *Annual Review of Materials Research* 32 (1) (2002) 297–319.
- [36] C. K. Birdsall, Particle-in-cell chargedparticle simulations, plus Monte Carlo collision with neutral atoms, PIC-MCC, *IEEE Transactions on Plasma Sciences* 19 (1991) 65 – 85.
- [37] J. M. Haile, *Molecular Dynamics Simulation*, John Wiley, 1992.
- [38] S. Succi, *The lattice Boltzmann Equation: for Fluid Dynamics and Beyond*, Oxford University Press, 2001.
- [39] W. W. Wood, J. J. Erpenbeck, Molecular dynamics and monte carlo calculations in statistical mechanics, *Annual Review of Physical Chemistry* 27 (1) (1976) 319–348.

- [40] E. Oran, C. Oh, B. Cybyk, Direct simulation monte carlo: Recent advances and applications, *Annual Review of Fluid Mechanics* 30 (1) (1998) 403–441.
- [41] K. H. Guenther, Vacuum deposition processes and the properties of thin films, *Plating and Surface Finishing* 81 (4) (1994) 68 – 71.
- [42] S. Chatain, C. Gonella, C. Roblin, Experimental results and Monte Carlo simulations for a gadolinium atomic beam produced with a focused electron gun, *Journal of Physics D: Applied Physics* 30 (3) (1997) 360 – 367.
- [43] S. Chatain, C. Gonella, P. Roblin, Experimental and computational study of copper-gadolinium alloy evaporation, *Journal of Applied Physics* 83 (6) (1998) 2934.
- [44] W. Wang, R. Hammond, M. Fejer, M. Beasley, Diode-laser-based atomic absorption monitors for physical vapor deposition process control, in: *Materials Research Society Symposium - Proceedings*, Vol. 502, 1998, pp. 193 – 201.
- [45] A. Kersch, W. Morokoff, C. Werner, D. Restaino, B. Vollmer, Modeling of a Sputter Reactor Using the Direct Simulation Monte Carlo Method, *International Electron Devices Meeting (Tech. Digest)* (1992) 181 – 184.
- [46] A. Kersch, W. Morokoff, C. Werner, Self-consistent Sputter deposition with the Monte Carlo Method, *J. Appl. Phys.* 75 (4) (1994) 2278 – 2285.
- [47] J. F. Groves, G. Mattausch, H. Morgner, D. D. Hass, H. N. G. Wadley, Directed vapour deposition, *Surface Engineering* 16 (6) (2000) 461 – 464.
- [48] Z. Y. D. of Materials Science, U. o. V. Engineering, D. D. Hass, H. N. G. Wadley, NiAl bond coats made by a directed vapor deposition approach, *Materials Science and Engineering A* 394 (1-2) (2005) 43 – 52.
- [49] J. F. Groves, H. N. G. Wadley, Monte Carlo modeling of atom transport during directed vapor deposition, in: *Thin Films - Structure and Morphology*, Vol. 441, *Materials Research Society*, Pittsburgh, PA, USA, 1997, pp. 541 – 548.
- [50] D. D. H. D. of Materials Scien., S. o. E. Engineering, U. o. V. Applied Sciences, J. F. Groves, H. N. G. Wadley, Reactive vapor deposition of metal oxide coatings, *Surface and Coatings Technology* 146-147 (2001) 85 – 93.
- [51] D. L. Hill, Jet Vapor Deposition Coating of Fibers, Master’s thesis, University of Virginia (1994).
- [52] D. L. Youchison, M. A. Gallis, R. E. Nygren, J. M. McDonald, T. J. Lutz, Effects of ion beam assisted deposition, beam sharing and pivoting in EB-PVD processing of graded thermal barrier coatings, *Surface and Coatings Technology* 177 (2004) 158 – 164.
- [53] Z. Lin, T. S. Cale, Flux distributions and deposition profiles from hexagonal collimators during sputter deposition, *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films* 13 (4) (1995) 2183–2188.

- [54] Z. Lin, T. S. Cale, Simulation of collimated flux distributions during physical vapor deposition, *Thin Solid Films* 270 (1995) 627 – 631.
- [55] D. J. Economou, T. J. Bartel, Direct simulation Monte Carlo (DSMC) of rarefied gas flow during etching of large diameter (300-mm) wafers, *IEEE Transactions on Plasma Sciences* 24 (1996) 131 – 132.
- [56] D. Hash, M. Meyyappan, A Direct Simulation Monte Carlo Study of Flow Considerations in Plasma Reactor Development for 300 mm Processing, *Journal of the Electrochemical Society* 144 (11) (1997) 3999 – 4004.
- [57] P. O’Sullivan, F. H. Baumann, G. H. Gilmer, Simulation of physical vapor deposition into trenches and vias: Validation and comparison with experiment, *Journal of Applied Physics* 88 (7) (2000) 4061 – 4068.
- [58] C. C. Hwang, G. J. Huang, S. P. Ju, J. G. Chang, Incident ion characteristics in ionized physical vapor deposition using molecular dynamics simulation, *Surface Science* 512 (2002) 135 – 150.
- [59] J. Cho, S. G. Terry, R. LeSara, C. G. Levi, A kinetic Monte Carlo simulation of film growth by physical vapor deposition on rotating substrates, *Materials Science and Engineering A* 391 (2005) 390 – 401.
- [60] S. G. Terry, Evolution of microstructure during the growth of thermal barrier coatings by electron-beam physical vapor deposition, Ph.D. thesis, University of California, Santa Barbara, CA (2001).
- [61] U. Schulz, S. G. Terry, C. G. Levi, Microstructure and texture of EB-PVD TBCs grown under different rotation modes, *Materials Science and Engineering, A* 360 (1-2) (2003) 319 – 329.
- [62] S. Osher, J. A. Sethian, Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics* 79 (1) (1988) 12 – 49.
- [63] J. A. Sethian, *Level Set Methods*, Cambridge University Press, New York, 1996.
- [64] T. S. Cale, B. R. Rogers, T. P. Merchant, L. J. Boruckic, Deposition and etch processes: continuum film evolution in microelectronics, *Computational Materials Science* 12 (4) (1998) 333 – 353.
- [65] D. G. Coronell, E. W. Egan, G. Hamilton, A. Jain, R. Venkatraman, B. Weitzman, Monte Carlo simulations of sputter deposition and step coverage of thin films, *Thin Solid Films* 333 (1998) 77 – 81.
- [66] V. Arunachalam, S. Rauf, D. G. Coronell, P. L. G. Ventzek, Integrated multi-scale model for ionized plasma physical vapor deposition, *Journal of Applied physics* 90 (1) (2001) 64 – 73.

- [67] D. G. Coronell, D. E. Hansen, A. F. Voter, C.-L. Liu, X.-Y. Liu, Molecular dynamics-based ion-surface interaction models for ionized physical vapor deposition feature scale simulations, *Applied Physics Letters* 73 (26) (1998) 3860 – 3862.
- [68] A. Powell, P. Minson, G. Trapaga, U. Pal, Mathematical modeling of vapor-plume focusing in electron-beam evaporation, *Metallurgical and Materials Transactions A: Physical Metallurgy and Materials Science* 32 (8) (2001) 1959 – 1966.
- [69] D. Lee, C. Bonilla, Viscosity of alkali metal vapors, *Nuclear Engineering and Design* 7 (5) (1968) 455 – 469.
- [70] I. F. Stepanenko, N. I. Sidorov, Y. V. Tarlakov, V. S. Yargin, Apparatus for measuring the viscosity of an alkali-metal vapor at high temperatures, *Measurement Techniques (English translation of Izmeritel'naya Tekhnika)* 30 (8) (1987) 829 – 832.
- [71] D. Timrot, B. Reutov, O. Krechetov, Experimental unit for studying the viscosity of vapors of alkaline-earth metals, *High Temperature (English translation of Teplofizika Vysokikh Temperatur)* 28 (1) (1990) 111 – 114.
- [72] N. Vargaftik, Y. Vinogradov, V. Dolgov, V. Dzis, I. Stepanenko, Y. Yakimovich, V. Yargin, Viscosity and thermal conductivity of alkali metal vapors at temperatures up to 2000 K, *International Journal of Thermophysics* 12 (1) (1991) 85 – .
- [73] M. Ghatee, M. Sanchooli, Viscosity and thermal conductivity of cesium vapor at high temperatures, *Fluid Phase Equilibria* 214 (2) (2003) 197 – 209.
- [74] C. Nieto de Castro, J. Fareleira, P. Matias, M. Ramires, A. Canelas Pais, A. Varandas, Thermophysical properties of alkali metal vapours. part i. theoretical calculation of the properties of monatomic systems, *Berichte der Bunsengesellschaft fuer Physikalische Chemie* 94 (1) (1990) 53 – 59.
- [75] E. Mukhtarov, A. Semenov, Non-empirical calculation of the transport coefficients of lithium and sodium vapor, *High Temperature (English translation of Teplofizika Vysokikh Temperatur)* 28 (1) (1990) 49 – 55.
- [76] M. H. Ghatee, F. Niroomand-Hosseini, Hard-wall potential function for transport properties of alkali metal vapors, *Journal of Chemical Physics* 126 (1) (2007) 014302 – .
- [77] P. Fialho, J. Fareleira, M. Ramires, C. Nieto de Castro, Thermophysical properties of alkali metal vapours part ia. prediction and correlation of transport properties monatomic systems, *Berichte der Bunsengesellschaft fuer Physikalische Chemie* 97 (11) (1993) 1487 – .
- [78] G. A. Bird, Direct simulation of the boltzmann equation, *Physics of Fluids* 13 (1970) 2676.

- [79] K. Nanbu, Theoretical basis of the direct simulation monte carlo method, in: V. Boffi, C. Cercignani (Eds.), *Rarefied Gas Dynamics*, Vol. 1, Stuttgart: Teubner, 1986, pp. 369–383.
- [80] W. Wagner, A convergence proof for bird's direct simulation monte carlo method for the boltzmann equation, *Journal of Statistical Physics* 66 (1992) 1011.
- [81] F. J. Alexander, A. L. Garcia, B. J. Alder, A consistent boltzmann algorithm, *Phys. Rev. Lett.* 74 (26) (1995) 5212–5215.
- [82] S. Chapman, T. G. Cowling, *The Mathematical Theory of Non-Uniform Gases: An Account of the Kinetic Theory of Viscosity, Thermal Conduction and Diffusion in Gases Second Edition*, Cambridge University Press, 1952.
- [83] J. O. Hirschfelder, C. F. Curtiss, R. B. Bird, *The Molecular Theory of Gases and Liquids*, Wiley-Interscience, 1964.
- [84] G. A. Bird, Recent advances and current challenges for DSMC, *Computers & Mathematics with Applications* 35 (1-2) (1998) 1 – 14.
- [85] G. A. Bird, Monte carlo simulation in an engineering context, *Progress in Astronautics and Aeronautics* 74 (1981) 239 – 255.
- [86] K. Koura, H. Matsumoto, Variable soft sphere molecular model for inverse-power-law or lennard-jones potential, *Physics of Fluids A: Fluid Dynamics* 3 (10) (1991) 2459–2465.
- [87] K. Koura, H. Matsumoto, Variable soft sphere molecular model for air species, *Physics of Fluids A: Fluid Dynamics* 4 (5) (1992) 1083–1085.
- [88] J. E. Lennard-Jones, On the Determination of Molecular Fields. I. From the Variation of the Viscosity of a Gas with Temperature, *Journal Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 106 (738) (1924) 441 – 462.
- [89] J. E. Lennard-Jones, On the Determination of Molecular Fields. II. From the Equation of State of a Gas, *Journal Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 106 (738) (1924) 463 – 477.
- [90] J. E. Lennard-Jones, Cohesion, *Proceedings of the Physical Society* 43 (5) (1931) 461 – 482.
- [91] H. A. Hassan, D. B. Hash, A generalized hard-sphere model for monte carlo simulation, *Physics of Fluids A: Fluid Dynamics* 5 (3) (1993) 738–744.
- [92] D. Hash, J. N. Moss, H. Hassan, Direct simulation of diatomic gases using the generalized hard sphere model, *Journal of Thermophysics and Heat Transfer* 8 (4) (1994) 758 – 764.

- [93] A.U.Chatwani, Monte-carlo simulation of nozzle beam expansion, *Progress in Astronautics and Aeronautics* 51 (1977) 461 – 475.
- [94] C. Borgnakke, P. S. Larsen, Statistical collision model for monte carlo simulation of polyatomic gas mixture, *Journal of Computational Physics* 18 (1975) 405 – 420.
- [95] R. G. Lord, A new model of energy exchange in inelastic collisions, in: J. Harvey, G. Lord (Eds.), *Rarefied Gas Dynamics* 19, Vol. 1, Oxford University Press, 1995, pp. 564–570.
- [96] F. Bergemann, I. D. Boyd, New discrete vibrational energy model for the direct simulation Monte Carlo method, *Progress in Aeronautics and Astronautics* 174 (1994) 158.
- [97] J. S. Wu, Y. Y. Lian, Parallel Three-Dimensional Direct Simulation Monte Carlo method and its Applications, *Computers and Fluids* 32 (2003) 1133 – 1160.
- [98] J.-S. Wu, K.-C. Tseng, T.-J. Yang, Parallel Implementation of the Direct Simulation Monte Carlo Method Using Unstructured Mesh and Its Application, *International Journal of Computational Fluid Dynamics* 17 (3) (2003) 405 – 422.
- [99] R. G. Wilmoth, G. J. Lebeau, A. B. Carlson, DSMC Grid Methodologies for Computing Low-Density, Hypersonic Flows About Reusable Launch Vehicles, Tech. rep., NASA (1996).
- [100] Altair Inc., USA, Altair Hypermesh Manual, 7th Edition (2005).
URL <http://www.altair.com>
- [101] J. Wu, K. Tseng, F. Wu, Parallel three-dimensional dsmc method using mesh refinement and variable time-step scheme, *Computer Physics Communications* 162 (3) (2004) 166 – 187.
- [102] K. Tseng, Ph.D. thesis, National Chiao Tung University, Taiwan (2005).
- [103] Y.-Y. Lian, K. H. Hsu, Y. L. Shao, Y. M. Lee, Y.-W. Jeng, J. S. Wu, Parallel Adaptive Mesh Refining Scheme on 3-Dimensional Unstructured Mesh and its Applications, *Computer Physics Communications* 175 (11) (2006) 721 – 737.
- [104] D. Vanderstraeten, R. Keunings, Optimized partitioning of unstructured finite element meshes, *International Journal of Numerical Methods and Engineering* 38 (3) (1995) 433 – 450.
- [105] R. P. Nance, D. B. Hash, H. A. Hassan, Role of boundary conditions in monte carlo simulation of micro-electromechanical systems, *AIAA Journal* 12 (3) (1998) 447 – 449.
- [106] J.-S. Wu, K.-C. Tseng, Parallel DSMC Method Using Dynamic Domain Decomposition, *International Journal for Numerical Methods in Engineering* 63 (2005) 37 – 76.

- [107] D. M. Nicol, J. H. Saltz, Dynamic Remapping of Parallel Computations with Varying Resource Demands, *IEEE Transactions on Computers* 37 (9) (1988) 1073 – 1087.
- [108] S. Dietrich, I. D. Boyd, Scalar and Parallel Optimized Implementation of the Direct Simulation Monte Carlo Method, *Journal of Computational Physics* 126 (2) (1996) 328 – 342.
- [109] T. J. Bartel, S. Plimpton, M. A. Gallis, Icarus: A 2-D Direct Simulation Monte Carlo (DSMC) Code for Multi-Processor Computers, Tech. rep., Sandia National Laboratories, Sandia National Laboratories P.O. Box 5800 Albuquerque, NM 87185-0626 USA (2001).
- [110] J. Bellot, H. Duval, D. Ablitzer, Validity of the kinetic langmuir’s law for the volatilization of metallic elements in vacuum metallurgy: a numerical approach, in: *Gas Interactions in Nonferrous Metals Processing*, 1996, pp. 109 – 119.
- [111] R. Venkataraman, Y. Y. Lian, J. S. Wu, G. Cheng, R. P. Koomullil, Three Dimensional Parallel DSMC Method for High Speed Re-entry Flow Simulations, in: *Proceedings of the 22nd SE Conference on Theoretical and Applied Mechanics*, Vol. 22, 2004, pp. 125 – 134.
- [112] G. C. Cheng, R. P. Koomullil, B. K. Soni, Multidisciplinary and multi-scale computational field simulations-algorithms and applications, *Mathematics and Computers in Simulation* 75 (5-6) (2007) 161 – 170.
- [113] D. P. Weaver, E. P. Muntz, D. H. Campbell, Direct Simulation Monte Carlo Calculations compared with sonic orifice expansion flows of Argon and Nitrogen, *Progress in Aeronautics and Astronautics (AIAA)* 158 (1994) 98 – 108.
- [114] V. K. Dogra, J. N. Moss, R. G. Wilmoth, J. M. Price, Hypersonic Rarefied Flow past spheres including wake structure, *Journal of Spacecraft and Rockets* 31 (5) (1994) 713 – 718.
- [115] J. N. Moss, J. M. Price, Survey of Blunt Body Flows including wakes at Hypersonic Low-Density Conditions, *Journal of Thermophysics and Heat Transfer* 11 (3) (1997) 321 – 329.
- [116] J. Allegre, D. Bisch, J. C. Lengrand, Experimental Rarefied Density Flowfields at Hypersonic Conditions over 70-degree Blunted Cone, *Journal of Spacecraft and Rockets* 34 (6) (1997) 714 – 718.
- [117] J. Allegre, D. Bisch, J. C. Lengrand, Experimental Rarefied Aerodynamic Forces at Hypersonic Conditions over 70-degree Blunted Cone, *Journal of Spacecraft and Rockets* 34 (6) (1997) 719 – 723.
- [118] J. Allegre, D. Bisch, J. C. Lengrand, Experimental Rarefied Heat Transfer at Hypersonic Conditions over 70-degree Blunted Cone, *Journal of Spacecraft and Rockets* 34 (6) (1997) 724 – 728.

- [119] M. Knudsen, The Molecular Heat Conductivity of Gases and the Accommodation Coefficient, *Annalen der Physik* 339 (4) (1911) 593 – 656.
- [120] Lilley, R. Charles, M. N. Macrossan, Methods for Implementing the Stream Boundary Condition in DSMC Computations, *International Journal for Numerical Methods in Fluids* 42 (2003) 1363 – 1371.