A MODULAR PARTICLE-CONTINUUM NUMERICAL ALGORITHM FOR HYPERSONIC NON-EQUILIBRIUM FLOWS

by

Thomas Edward Schwartzentruber

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2007

Doctoral Committee:

Professor Iain D. Boyd, Chairperson
Professor Michael R. Combi,
Professor Kenneth G. Powell
Professor Bram van Leer

Dedicated entirely to Shannon, my wife.

ACKNOWLEDGEMENTS

To begin, I would like to acknowledge my doctoral advisor, Professor Iain Boyd. I could not have hoped for a better advisor during my four years at the University of Michigan. Thank you for allowing me the freedom to pursue my own research ideas while at the same time providing constructive technical criticism and always sound guidance. Thank you for providing the means for me to attend conferences and workshops while always encouraging and challenging me to do so. Finally, in addition to all of the academic and technical research support you have provided me during the past four years, thank you for always ensuring that my long term career goals were a high priority.

I would also like to acknowledge and thank the other faculty members on my dissertation committee for reviewing my research and providing valuable feedback, which helped improve the final version. In general, I would like to thank all of the faculty and staff in the Department of Aerospace Engineering for providing such a great atmosphere for post-graduate education and research. The secretarial and technical staff were always there to help and enabled me to really focus on my research with very few distractions. In addition I would like to acknowledge and thank the staff at the International Center for doing an incredible job supporting international students with very limited resources.

I would like to acknowledge and thank my colleagues (both current and former) in the Nonequilibrium Gas and Plasma Dynamics group (...too numerous to list!). It

was great to be part of such a productive research group where someone was always willing and able to help out when someone else encountered a problem. Thank you all for making each day in the office enjoyable and for the daily invitations to go out for lunch for four years! Special thanks goes to Jon Burt for answering my countless DSMC questions over the years and for our scientific discussions in airports and on flights to and from many conferences. Your expertise and feedback certainly improved the quality of my research. Finally, special acknowledgement goes to Leonardo Scalabrin who developed the LeMANS code used in my research. Developing such a sophisticated and robust CFD code in such a short time was truly a remarkable accomplishment. Leo, your success allowed me to focus entirely on *hybrid* particle-continuum research and enabled me to progress much further than would have been possible without your contribution.

I would like to acknowledge all of the organizations that have funded my education and research over the years. Without such funding I would not have had the opportunity to pursue a post-graduate education and an academic career in research. Acknowledgement goes to the National Aeronautics and Space Administration (NASA), the U.S. Department of Defense (DOD), the U.S. Air Force Office of Scientific Research (AFOSR), and the Natural Sciences and Engineering Research Council of Canada (NSERC). I would like to thank the American Institute of Aeronautics and Astronautics (AIAA) for organizing the journals and conferences which enhance the aerospace community worldwide, as well as for providing scholarships and awards to graduate students such as myself.

I am very fortunate to have parents who have supported me one-hundred percent, during each stage of my education. Thank you for stressing education from day-one, for your un-wavering support in many ways, and for always showing an interest in

what I do. Finally, and most importantly, I wish to acknowledge my wife Shannon, who has worked just as hard as I have during the past four years. Thank you for your constant love and support, and for your complete confidence in me. Having such a great marriage makes it easy for me to work hard when I have to, while reminding me that there is much more to life than science and engineering!

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF NOMENCLATURE

Greek Symbols

$\Delta[\ ]_{collisions}$	the rate of change due to collisions
Δt	simulation timestep
Δt_{ss}	time until simulation has reached steady-state
Δx	local simulation cell size
Ω	solid angle element
β	a weighting coefficient
γ	ratio of specific heats, $C_p\ /C_v$
ζ	number of internal molecular degrees of freedom
θ	a weighting coefficient used within sub-relaxation average
κ	coefficient of thermal conductivity
λ	mean-free-path
μ	coefficient of viscosity
ρ	mass density
σ	collision cross-section
$\sigma d\Omega$	differential collision cross-section
σ_T	total collision cross-section
τ_c	mean-free-time
τ_{chem}	chemical relaxation time
τ_{elec}	electrical energy mode relaxation time
τ_{flow}	characteristic flow time
τ_{ijk}	the shear stress tensor
τ_{rot}	rotational energy mode relaxation time
τ_{trans}	translational energy mode relaxation time
τ_{vib}	vibrational energy mode relaxation time
ω	power-law exponent

Roman Symbols

A	a macroscopic flow property
Br_{cutoff}	the cutoff value for continuum breakdown
C_p	the coefficient of pressure
C_q	the coefficient of heat transfer
$C_{\tau\|\|}$	the coefficient of tangential shear-stress

C_v	the specific heat at constant volume
$\mathbf{C}$	the particle thermal velocity vector, $\mathbf{C}(C_x, C_y, C_z)$
C_x	the x-component of the particle thermal velocity vector
C_y	the y-component of the particle thermal velocity vector
C_z	the z-component of the particle thermal velocity vector
$\mathcal{C}$	the normalized particle thermal velocity vector, $\mathbf{C}\ /\sqrt{2kT/m}$
E	the local percentage difference between two solutions
F	a global mesh scaling factor
F_c	the macroscopic flux computed from a continuum cell
F_p	the macroscopic flux computed from a particle cell
Kn	the Knudsen number
Kn_{GL}	the gradient-length Knudsen number
Kn_g	the global Knudsen number
N_{Br}	the number of timesteps between calculation of Kn_{GL}
Np	the number of particles in a DSMC cell
Nrc	the number of refined DSMC cells within a continuum NS cell
N_{real}	the number of real gas particles in a computational cell volume
P	the probability that a particle undergos a collision within Δt
Q	a macroscopic flow property
$Q(\mathbf{c})$	a function of the particle velocity vector
R_b	the base radius of the planetary probe
R_{cyl}	the radius of the 2D cylinder
R_{gas}	the specific gas constant
R_n	the nose radius of the planetary probe
S_t	the Stanton number (or coefficient of heat transfer)
T	the translational-rotational temperature (in continuum regions)
T_{ROT}	the rotational temperature (in particle regions)
T_{TRA}	the translational temperature (in particle regions)
T_w	the specified wall (surface) temperature
$\mathbf{V}$	the bulk velocity vector, $\mathbf{V}(u, v, w)$
V_{cell}	the volume of a computational cell
W	the effective particle weight
a	the local speed of sound, $\sqrt{\gamma R_{gas} T}$
c_r	the relative speed between colliding particles
$\mathbf{c}$	the particle velocity vector $\mathbf{c}(v_x, v_y, v_z)$
d	the particle diameter
e_{ROT}	the rotational energy contained by a gas particle
f	the velocity distribution function, $f(\mathbf{r}, \mathbf{c}, t)$
f_0	the Maxwellian (or equilibrium) velocity distribution function
g	a non-equilibrium velocity distribution function
k	the Boltzmann constant, $k = 1.38 \times 10^{-23}$ Joules/Kelvin
l_f	the length of a computational cell face
m	the mass of a particle

n	the local number density
n_{DSMCbcs}	the number of DSMC boundary cells
n_{NSbcs}	the number of NS boundary cells
n_{overlap}	the number of overlap cells
$\hat{n}_f$	the unit normal vector of a computational cell face
p	the thermodynamic pressure
q_i	the heat transfer vector
$\mathbf{r}$	the position vector, $\mathbf{r}(x, y, z)$
r_f	the mesh refinement factor corresponding to cell face
s	the distance around the planetary probe surface
t	time
u	the x-component of the bulk velocity vector
v	the y-component of the bulk velocity vector
v_x	the x-component of the particle velocity vector
v_y	the y-component of the particle velocity vector
v_z	the z-component of the particle velocity vector
w	the z-component of the bulk velocity vector
x_{sep}	the location of flow separation

Acronyms

AMAR	Adaptive Mesh and Algorithm Refinement
BTE	Boltzmann Transport Equation
CFD	Computational Fluid Dynamics
CFL	Courant-Friedrichs-Lewy number
CUBRC	Calspan - University of Buffalo Research Center
DSMC	Direct Simulation Monte Carlo
DSMCdata	The DSMC module's data structure
HCdata	The continuum portion of the hybrid data structure
HPdata	The particle portion of the hybrid data structure
HS	Hard-Sphere
IP	Information Preservation
LENS	Large Energy National Shock
LeMANS	The (Le) Michigan Aerothermodynamic Navier-Stokes solver
MONACO	A DSMC code developed at the University of Michigan
MPC	Modular Particle-Continuum
NS	Navier-Stokes
NSdata	The Navier-Stokes module's data structure
NTC	No-Time-Counter
VHS	Variable Hard-Sphere
vdf	velocity distribution function

Superscripts

$*$	refers to a post-collision quantity
T	refers to the transpose of a matrix
tr	refers to the translational energy mode
rot	refers to the rotational energy mode

Subscripts

∞	refers to a free-stream quantity
R	refers to a simulation utilizing mesh refinement
n	refers to a normalized quantity
$post$	refers to the post-shock region
pre	refers to the pre-shock region
ref	refers to a defined constant reference quantity

CHAPTER I

Introduction

The purpose of this dissertation is to improve upon the numerical simulation tools currently available to model hypersonic, non-equilibrium gas flows. Prediction of the aerothermodynamic loads experienced by a vehicle flying at hypersonic speeds involves many physical processes that occur over a large range of spatial and temporal scales. Efficient numerical simulation involves constructing the simplest possible model (at the largest possible scale) able to describe the phenomenon of interest. However, for problems in which important physical effects are generated over a wide range of scales, smaller scale processes must often be included in the model. Unfortunately, such fine spatial and temporal resolution is often beyond the current capabilities of even the most advanced computer systems.

Such is the case for hypersonic, non-equilibrium flows where in general the gas behaves as a continuous fluid, although under certain circumstances the molecular nature of the gas must be modelled as well. This dissertation develops a numerical method that combines both microscopic and macroscopic approaches in order to more accurately and efficiently model the aerothermodynamic environment experienced by hypersonic aerospace vehicles.

1.1 The Molecular Nature of Gases

A gas is comprised of many atoms and molecules (particles) constantly moving around and colliding with each other. At a pressure of 1 atm and a temperature of 273 K, there are approximately 2.7×10^{19} gas particles per cubic centimeter [1]. Collisions occur due to the electrical repulsion between particles, which only acts over distances comparable to the size of the particles themselves. Since particles are extremely small, despite an enormous number of them occupying a given space, the majority of physical space in a gas is empty. As a result, most of the time particles are moving beyond the influence of other particles and will occasionally undergo a collision that occurs virtually instantaneously. A gas under such conditions is termed a dilute gas and is a valid assumption for all aerospace gas flows. This is in contrast to a liquid where particles are so closely packed that they are always in a "collisional" state with surrounding particles.

Of course it is ultimately through molecular collisions that the state of a gas changes (slows down, heats up, chemically reacts, etc.) and interacts with the surface of a vehicle. Due to the discrete nature of collisions in a dilute gas, fundamental spatial and temporal scales can be defined. The mean-free-path (λ) of a gas is defined as the average distance travelled by gas particles between successive collisions. A related quantity, the mean-free-time (τ_c), is defined as the average time between gas particle collisions. Atoms in a gas contain translational (kinetic) energy as well as chemical and electrical (internal) energy. Gas molecules contain additional internal energy in rotational and vibrational modes. A major focus in modelling hypersonic flows involves predicting how energy is transferred between these different modes. Each process has a characteristic time associated with it that can be compared to

τ_c, the mean-free-time. In general, these time scales are distinctly separated and are ordered as follows:

$$\tau_c \approx \tau_{trans} < \tau_{rot} < \tau_{vib} < \tau_{chem} \ . \tag{1.1}$$

Since τ_c represents the average time between collisions, the time scales for each process can also be regarded as a measure of the number of collisions required for energy to be transferred and equally distributed within that particular mode. For example, in a hypersonic wind tunnel experiment, very high speed gas particles are slowed rapidly by a test model and form a shock wave. Here, the free-stream particles with large directed kinetic energy require only a few collisions with post-shock particles for this directed translational energy to be transferred into random or thermal translational energy (temperature). It requires approximately ten collisions ($10\tau_c$) for this increased thermal energy to uniformly increase the rotational energy modes and on the order of one thousand collisions ($1000\tau_c$) to uniformly increase the vibrational energy modes. It is then important to compare these relaxation times with the characteristic flow time τ_{flow}, defined as the time required for the free-stream flow to traverse the vehicle. The relaxation times associated with the above processes all scale with τ_c. Thus for high altitude (rarefied) flows, λ and τ_c may be sufficiently large that some or all of the relaxation times are comparable to τ_{flow}. In such a case, the flow is described as being in thermochemical non-equilibrium. In contrast, for very high density flows ($\tau_c \to 0$), at each point in the flow energy is equally distributed between all modes and chemical composition is instantaneously altered to reflect these conditions. In this case the flow is described as being in equilibrium.

The majority of hypersonic flows of interest consist of mainly near-equilibrium flow. At each location in the flow there is an enormous number of collisions occurring

per unit time such that the properties of all gas particles become normally distributed around a local mean value. Thus, description of the underlying molecular nature is no longer required and the gas flow can be modelled very accurately in terms of the macroscopic mean values alone (such as density, bulk velocity, and temperature). The continuum Navier-Stokes (NS) equations model gas flows in this manner and can be solved very efficiently using techniques from Computational Fluid Dynamics (CFD). CFD methods have proven very successful in simulating a wide range of aerospace flows. By representing vibrational and electrical energy modes with an additional transport equation, the NS equations can further be extended to model flows in thermochemical non-equilibrium.

However, the NS equations are not readily extended to model translational non-equilibrium that occurs when macroscopic gradients begin to vary over length scales comparable to the mean-free-path. This situation can arise within thin flow features such as shock waves, boundary layers, and sharp leading edges and also in regions of rapid expansion and rarefied flow where λ becomes large. Figure 1.1 displays four shadowgraph images produced by wind tunnel tests of early re-entry vehicle concepts. H. Julian Allen pioneered and developed the blunt-body concept, which made possible the heat shield designs used for the Mercury, Gemini and Apollo space capsules. As seen in Fig. 1.1, a blunt-body produces a shockwave in front of the vehicle that actually shields the vehicle from excessive heating. As a result, blunt-body vehicles can stay cooler than sharp, low-drag vehicles. A shadowgraph is a process in which light passing through a flowing fluid is refracted by the density gradients in the fluid, resulting in bright and dark areas on a screen placed behind the fluid. The above-mentioned non-equilibrium flow features are visible in the experimental shadowgraph images shown in Fig. 1.1. The degree of collisional non-equilibrium

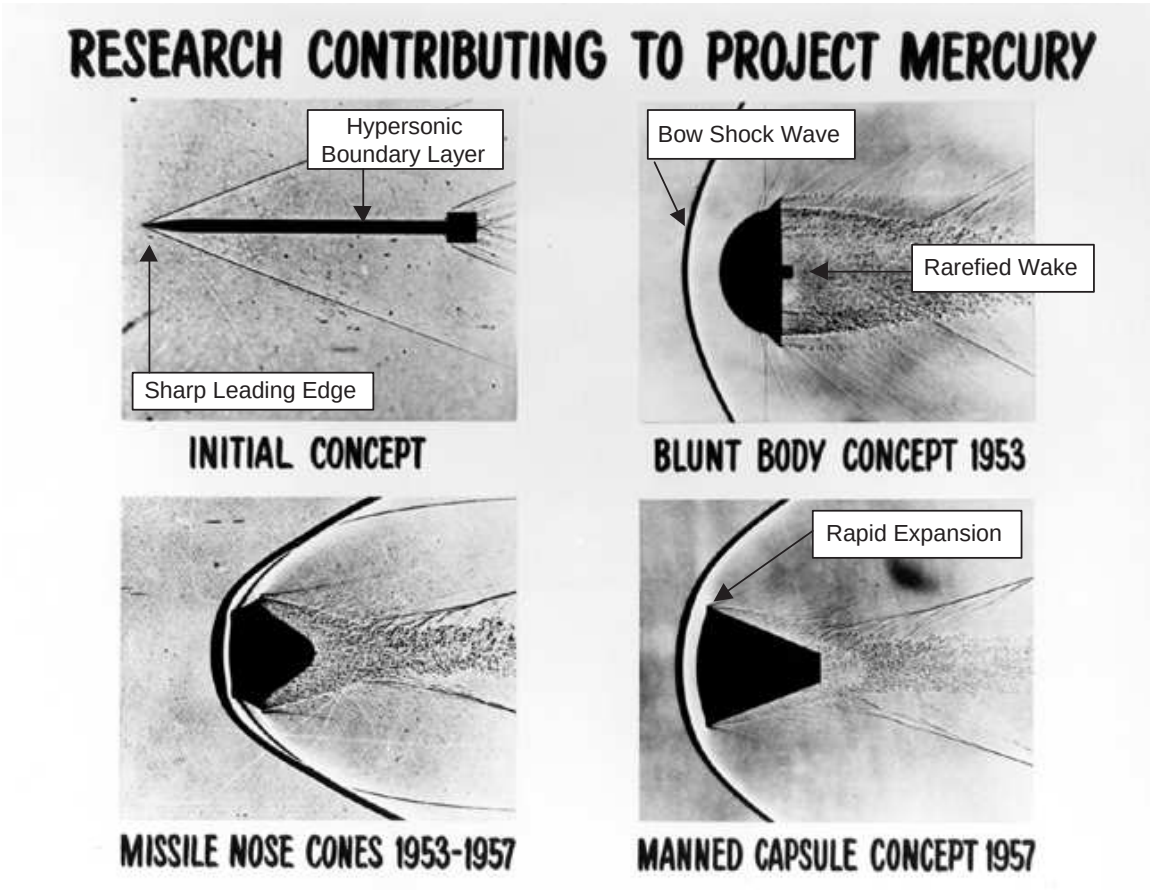


Figure 1.1: Shadowgraph images of early re-entry vehicle concepts. (Great Images in NASA library, http://grin.hq.nasa.gov, image: GPN-2000-001938)

is indicated by the Knudsen number, which is defined as the ratio of λ to the characteristic length (or gradient-length) of the flow feature of interest. In regions of the flow where the Knudsen number is high, locally the distributions of particle velocities and energies are no longer normally distributed due to insufficient collision rates. Thus, modelling only the mean values of highly non-equilibrium distributions becomes physically inaccurate. The most popular numerical method for simulating high-speed, high Knudsen number (rarefied) flows is the direct simulation Monte Carlo (DSMC) particle method developed by Bird [1]. Hypersonic vehicles operate at high altitudes (higher values of λ) and as seen in Fig. 1.1, hyper-velocities induce sharp flow gradients (small characteristic length scales) as well as rarefied regions in the wake of such vehicles. As a result, hypersonic flows can contain a wide range of local Knudsen numbers for which both molecular and continuum descriptions of the gas may be required. The accurate and efficient numerical simulation of such hypersonic non-equilibrium flows is the subject of this thesis.

1.2 Hypersonic Non-Equilibrium Flows

Accurate numerical simulation is vital to the successful design of hypersonic cruise and re-entry vehicles. The high-enthalpy conditions experienced by such vehicles are difficult and in many cases impossible to reproduce in a wind tunnel on Earth. Actual flight data is limited to a small number of hypersonics projects conducted over the past several decades. Obtaining new flight data is extremely expensive and is only practical in the later stages of vehicle development. Typically, experiments are performed at conditions as close to flight conditions as possible. Numerical modelling techniques are then validated using this experimental data as well as existing flight data. The numerical models are then applied to the flight conditions in order to

build a database of the vehicle's expected performance over the entire range of flight conditions. Such databases are then used to design the vehicle and support its missions.

A review of both CFD and DSMC simulations and their agreement with various hypersonic, blunt-body experiments is presented in Ref. [2]. One conclusion of this review article is that while the CFD results generally match DSMC results and experimental data in the fore-body flow, they deviate significantly from DSMC and experiment in the near-wake for both surface and flow features. Indeed, the uncertainty in after-body heating rates is much larger than in the fore-body and is a current area of research. A survey of available after-body aeroheating flight data for planetary probe thermal protection system design is presented in Ref. [3]. The most valuable flight data available for numerical model validation purposes includes four unmanned Apollo heat-shield qualification tests. One particular flight test, the Apollo AS-202 configuration, has been studied in detail using CFD methods. Numerical results for the after-body flow of the Apollo AS-202 command module using CFD methods suggest non-equilibrium flow is likely present near the shoulder and rear apex of the capsule during the high altitude "skip" phase of re-entry [4]. These regions are very localized, but may explain the over-prediction in heating rates at these points, made by the CFD simulations in comparison with the flight data. These numerical models are then used to predict the after-body heating rates for upcoming missions such as the Mars Science Laboratory mission [5]. At different points in the trajectory of such vehicles the flow may lie completely within either the rarefied regime or completely within the continuum regime. Thus, NASA currently uses both DSMC and CFD numerical methods to support their missions. Both modelling techniques are currently being validated using experimental data from the Apollo missions [6] as

well as being used in the design of the new Orion spacecraft [7] planned to replace the Space Shuttle. Other missions such as the successful Mars Global Surveyor [8] and Mars Odyssey [9] relied on DSMC simulations for planning an aero-braking procedure which occurs in the rarefied, outer edge of a planet's atmosphere. Enormous weight savings are gained by the use of aero-braking for orbital insertion, rather than carrying the fuel required for thrusters. Finally, for landing high-mass payloads on Mars, various deceleration devices are being investigated. One such device is a ballute (balloon + parachute) which uses a large inflatable structure to produce large amounts of drag at very high altitudes [10, 11]. The high altitude operation of such a device results in portions of the flow being highly non-equilibrium. However, the large size of the device also results in large regions of continuum flow. One certain difficulty is prediction of the heating rates on the thin tethers that would connect the ballute to the entry capsule. The tethers will create complex flow interactions with the bow shocks and boundary layers of both the capsule and ballute. The flow around these tethers will most certainly be rarefied, however a DSMC simulation of the entire ballute configuration is beyond current computational capabilities.

Numerical simulation using both CFD and DSMC modelling techniques has been very successful for many of the above mentioned experiments and missions. However, new directions and challenges remain for the simulation of fluid dynamic flows [12]. Regarding the field of hypersonics in particular, a recent review paper produced by NASA Ames and Langley Research Centers, discusses the most pressing computational challenges for hypersonics [13]. One of the challenges specifically discussed is the need for multi-physics modelling and multi-scale (hybrid particle-continuum) modelling of hypersonic flows. This dissertation provides research towards this goal.

1.3 Thesis Overview

This chapter has introduced the molecular nature of gases and how, for the majority of aerospace flows, the physical processes occurring at the molecular level can be accurately replaced with a continuum description of the gas. Modelling of hypersonic non-equilibrium flows provides a challenge in that flow features are generated over a large range of spatial and temporal scales and in certain situations the molecular nature of the gas must be modelled as well. The need for hybrid kinetic-continuum modelling was outlined with reference to research recently published in the literature. However, before a proper overview of past and current hybrid kinetic-continuum research is presented, it is important to first summarize the modelling approaches currently used for both kinetic and continuum numerical simulation.

Chapter II first details the governing equations used to model gas flows from both the kinetic and continuum perspectives, after which a complete review of hybrid kinetic-continuum research from the literature is provided. The fundamental governing equation that provides the kinetic description of a dilute gas is the Boltzmann equation. This equation is detailed and the challenges associated with deterministically solving this equation are discussed. Next, simplifying continuum assumptions (such as taking moments of the Boltzmann equation) are outlined that result in the continuum Euler and Navier-Stokes equations. The validity of these continuum assumptions is discussed for hypersonic flows. Chapter II then describes the direct simulation Monte Carlo method; a particle method that statistically simulates the Boltzmann equation. The particle movement and collision procedures employed by the DSMC method are described. In addition, the spatial and temporal resolution required by the DSMC method and the limitations this places on efficient numerical

simulation are discussed. Finally, after outlining these current kinetic and continuum modelling methods, Chapter II presents a detailed summary of past and current research into hybrid numerical methods. Specifically, two categories of hybrid methods, "unified" kinetic-continuum approaches and "coupled" particle-continuum methods, are summarized.

Chapter III details the various components utilized by the hybrid particle-continuum method proposed in this thesis. The first components are the specific numerical schemes employed to both solve the Navier-Stokes equations and simulate the Boltzmann equation using the DSMC method. Another important procedure in a hybrid method is the calculation of a continuum breakdown parameter, which is used to identify flow regions in which a particle description is required and regions where a continuum description is adequate. Additional components outlined in Chapter III include a mesh refinement procedure for particle regions and a procedure to transfer information between particle and continuum domains during a hybrid simulation. Chapter III also completely details the procedures used to generate particles that propagate into non-equilibrium regions as well as averaging procedures that provide macroscopic information to be transferred into continuum regions.

Chapter IV couples DSMC and NS methods using the components described in Chapter III and the resulting hybrid method is tested for 1D normal shock waves. Within Chapter IV, detailed investigations of the progression of the hybrid solution, the effect of boundary conditions at hybrid particle-continuum interfaces, and the magnitude of statistical scatter, are presented. The hybrid particle-continuum method is used to simulate normal shock waves in argon and diatomic nitrogen gas over a wide range of shock Mach numbers. Hybrid results are compared with pure NS simulations, pure DSMC simulations, and with experimental measurements.

Chapter V revisits the detailed analysis of the hybrid method for 2D and axi-symmetric blunt-body flows. The hybrid numerical cycle presented in Chapter IV is extended for 2D and axi-symmetric flows and the finalized hybrid numerical cycle responsible for coupling all hybrid components is completely detailed. The hybrid numerical cycle is carefully constructed in order to ensure that hybrid solutions progress towards the correct physical result as opposed to diverging towards alternate incorrect solutions. Chapter V applies the finalized hybrid particle-continuum method to baseline cases of hypersonic flow over 2D cylinder geometries and an axi-symmetric planetary probe geometry. The resulting mainly continuum flow fields are analyzed for localized regions of non-equilibrium. In Chapter V, hybrid results for flowfield and surface properties as well as for local velocity distribution functions are compared with full DSMC solutions, full NS solutions, and with experimental data where available. The computational efficiency gained by the hybrid method over full DSMC simulation is also investigated and discussed in detail.

Chapter VI applies the finalized hybrid numerical method (completely detailed in Chapter V) to a hypersonic interaction flow. Hypersonic flow over an axi-symmetric hollow cylinder-flare geometry is simulated using full DSMC, full NS, and hybrid methods. The progression of the hybrid solution is again investigated in detail in order to further validate the hybrid numerical cycle. Hybrid results for flowfield and surface properties are compared with full DSMC and NS solutions as well as with available experimental data. Finally, in Chapter VI, the computational efficiency of the hybrid method for this hypersonic interaction flow is presented and discussed.

Chapter VII completely outlines the novel modular implementation used for the hybrid method described in this thesis. The source code and data structure organization of both DSMC and NS modules are first outlined, after which the modular

source code and data structure organization for the entire hybrid code is described. The modular implementation utilizes existing (un-modified) source code for both DSMC and NS modules and requires a limited amount of additional source code to implement the hybrid operations. As a result, the complete implementation of the hybrid particle-continuum method is able to be detailed within Chapter VII and within the referenced Algorithms in Appendix A.

Chapter VIII summarizes all conclusions drawn from each chapter and highlights five major contributions made by this dissertation to the field of hybrid kinetic-continuum modelling for hypersonic gas flows. The chapter and thesis then end with recommendations for, and discussion of, future research in the field.

CHAPTER II

Numerical Modelling of Gas Flows

In this chapter, the governing equations used to model gas flows from both kinetic and continuum perspectives are described. The difficulties involved in obtaining a deterministic solution to the kinetic equations and limitations and assumptions associated with the continuum equations are discussed. The direct simulation Monte Carlo particle method is described. Finally, a detailed review of research on hybrid kinetic-continuum methods from the literature is presented.

2.1 Kinetic Description

2.1.1 Boltzmann equation

The underlying governing equation that provides a kinetic description of a dilute gas is the single-particle distribution Boltzmann transport equation (BTE), which in the absence of body forces is given by:

$$\frac{\partial(nf)}{\partial t} + \mathbf{c} \cdot \frac{\partial(nf)}{\partial \mathbf{r}} = \Delta[f]_{collisions} \ . \tag{2.1}$$

The Boltzmann equation describes the evolution of the dependent variable f, referred to as the *v*elocity *d*istribution *f*unction (*vdf*). The variable f is a probability distribution function that represents the probability that a particle located at a certain spatial position has a certain velocity at a certain point in time. Thus f is a function

of seven independent variables, $\mathbf{r}(x, y, z)$, $\mathbf{c}(v_x, v_y, v_z)$, and time t. In Eq. 2.1, n is the number density, which is a function of $\mathbf{r}$, and t. Physically, the evolution in time of nf is due to both particle movement ($\mathbf{c} \cdot (\partial(nf)/\partial\mathbf{r})$ and particle collisions

$$\Delta[f]_{collisions} = \int_{-\infty}^{+\infty} \int_{0}^{4\pi} n^2 (f^* f_1^* - f f_1) c_r \sigma d\Omega d\mathbf{c_1} \ . \tag{2.2}$$

Here, for particles with pre-collision velocities corresponding to $f = f(\mathbf{c})$ and $f_1 = f(\mathbf{c_1})$, Eq. 2.2 must compute the post-collision velocities and the value of f corresponding to these post-collision velocities, $f^* = f(\mathbf{c}^*)$ and $f_1^* = f(\mathbf{c_1^*})$. In Eq. 2.2, c_r is the relative speed between colliding particles and $\sigma d\Omega$ is the differential cross-section of the colliding particles. In general, the differential cross-section is a function of the relative speed c_r, and the assumed molecular/atomic potential of the particles [14]. To obtain the total change in f due to collisions, integration over particles of all velocities and integration over the total collision cross-sections must be performed.

2.1.2 Deterministic solution

Analytical solution of Eq. 2.1 is limited mainly to collisionless flow over simple geometries. Solutions for more practical flows must be obtained numerically. However, obtaining a deterministic, numerical solution to Eq. 2.1 is extremely computationally demanding due to the high dimensionality of the equation. Not only is a discrete spatial grid required, but velocity space must also be discretized. At least a six-dimensional mesh is thus required and practical simulations become computationally prohibitive. If internal energy modes (rotation and vibration) are modelled, these energy distributions become further dimensions which must also be discretized. In addition, one must also determine how to discretize velocity space and internal energy levels. The number of discrete levels as well as the spacing between both

velocity and internal energy levels must be specified. The effects of velocity space and internal energy level discretization on the accuracy of numerical solutions to the BTE are not well known for hypersonic flows. However, recent work by Agarwal *et al.* [15, 16] presents numerical solutions of the BTE for flow of diatomic gases through normal shock waves and about a blunt-body. Such research is beginning to address the above-mentioned issues associated with deterministic solution of the Boltzmann transport equation.

2.1.3 Moments of the Boltzmann equation

The result of taking the zeroth, first, and second moments of Eq. 2.1 are the inviscid conservation equations (the Euler equations) for mass, momentum, and energy in a gas. Taking a moment of Eq. 2.1 involves multiplying the entire equation by a function $Q(\mathbf{c})$ and integrating over velocity space (over all $\mathbf{c}$). Such integration removes the dependence on velocity space and results in only mean, averaged values. Equation 2.1 then takes the following form:

$$\frac{\partial(n\overline{Q})}{\partial t} + \frac{\partial(n\overline{\mathbf{c}Q})}{\partial \mathbf{r}} = \Delta[Q]_{collisions} \ . \tag{2.3}$$

The zeroth, first, and second moments are taken by setting $Q(\mathbf{c})$ equal to m, $m\mathbf{c}$, and $m(c^2 + \zeta R_{gas}T)/2$. Here, ζ is the number of internal degrees of freedom of the gas particles. These functions representing mass, momentum, and energy are conserved during all collisions and thus the term $\Delta[Q]_{collisions}$ vanishes. It can further be shown that for the term $\Delta[Q]_{collisions}$ to vanish, f must have an equilibrium (or Maxwellian) distribution [1] given by

$$f_0(v_x, v_y, v_z) = \left(\frac{m}{2\pi kT}\right)^{\frac{3}{2}} exp\left[\frac{-m}{2kT}\left((v_x - u)^2 + (v_y - v)^2 + (v_z - w)^2\right)\right] \ . \tag{2.4}$$

A Maxwellian velocity distribution function is thus completely described by macroscopic gas properties $\mathbf{V}(u, v, w)$ and T. When moments corresponding to mass,

momentum, and energy are taken over this distribution function, Eq. 2.3 reduces to the Euler equations which describe the conservation of mass, momentum, and energy within an equilibrium gas.

2.2 Navier-Stokes Equations

If the local velocity distribution function f is assumed to be a slightly perturbed Maxwellian (termed a Chapman-Enskog velocity distribution), the result of taking the zeroth, first, and second moments of the BTE are the well known Navier-Stokes equations [14]. If rotational and translational energy modes are modelled by a single temperature T and the vibrational energy mode is not considered, the two-dimensional, laminar, Navier-Stokes equations for a single gas species can be written as:

$$\frac{\partial}{\partial t}\mathbf{U} + \frac{\partial}{\partial x}\mathbf{F} + \frac{\partial}{\partial y}\mathbf{G} = 0 \quad , \tag{2.5}$$

where $\mathbf{U} = [\rho \;\; \rho u \;\; \rho v \;\; \rho E]^T$,

$$\mathbf{F} = \left\{ \begin{array}{c} \rho u \\ \rho u^2 + p - \tau_{xx} \\ \rho uv - \tau_{xy} \\ (\rho E + p - \tau_{xx})u - \tau_{xy}v + q_x \end{array} \right\} \quad , \quad \mathbf{G} = \left\{ \begin{array}{c} \rho v \\ \rho uv - \tau_{yx} \\ \rho v^2 + p - \tau_{yy} \\ (\rho E + p - \tau_{yy})v - \tau_{yx}u + q_y \end{array} \right\}$$

The viscous stresses are modelled assuming a Newtonian fluid for which, using Stokes' hypothesis,

$$\tau_{xx} = 2\mu \left[\frac{\partial u}{\partial x} - \frac{1}{3}\nabla \cdot \mathbf{V} \right] \quad ,$$

$$\tau_{yy} = 2\mu \left[\frac{\partial v}{\partial y} - \frac{1}{3}\nabla \cdot \mathbf{V} \right] \quad ,$$

$$\tau_{xy} = \tau_{yx} = \mu \left[\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right] \quad , \tag{2.6}$$

where $\mathbf{V}(u, v, w)$ is the bulk fluid velocity vector. The coefficient of viscosity, μ, is modelled using the same power-law relationship used by the DSMC method, which is detailed in the next paragraph. This consistency is important not only at hybrid particle-continuum interfaces (for information transfer) but also whenever comparing DSMC and NS flow field results and surface properties. The heat-flux terms are modelled according to Fourier's law as

$$q_x = -\kappa \frac{\partial T}{\partial x} \quad \text{and} \quad q_y = -\kappa \frac{\partial T}{\partial y} \quad , \tag{2.7}$$

where the thermal conductivity, κ, is determined from Eucken's relation

$$\kappa = \mu \left(\frac{5}{2} C_V^{(tr)} + C_V^{(rot)} \right) \quad . \tag{2.8}$$

Here, $C_V^{(tr)} = \frac{3}{2} R_{gas}$ and $C_V^{(rot)} = R_{gas}$ are the translational and rotational contributions to the specific heat of the gas at constant volume. Finally, Eqns. 2.5 through 2.8 are closed using the perfect gas equation of state,

$$p = \rho R_{gas} T \quad , \tag{2.9}$$

where R_{gas} is the specific gas constant. A wide variety of numerical methods developed in the field of CFD can then be used to solve this set of equations.

Again, these governing continuum equations are obtained by taking moments of the Boltzmann transport equation (Eq. 2.1) where f is assumed to be a Chapman-Enskog distribution function [1]. In the process, new viscous terms are introduced that account for the transfer of energy (heat transfer) and momentum (shear-stress) within the gas. These terms were closed using the Stokes and Fourier models (Eqs. 2.6 and 2.7), which assume that heat and momentum transfer are linear functions

of macroscopic flow gradients. For non-equilibrium flow, where such gradients may occur over a few mean-free-paths (high local Knudsen numbers), these assumptions become invalid. Mass, momentum, and energy are ultimately carried by gas particles; a process which may be completely non-linear. The BTE (Eq. 2.1) fully accounts for this physical process and is therefore accurate for all Knudsen numbers, however the NS equations do not and therefore become inaccurate for high Knudsen number flows.

2.3 The Direct Simulation Monte Carlo Method

Instead of pursuing a deterministic solution to the Boltzmann transport equation for the velocity distribution function f, a probabilistic particle method can be used to simulate the physics embodied in Eq. 2.1. The direct simulation Monte Carlo (DSMC) method models a gas flow with a large number of simulated particles that move through a computational mesh, collide with surfaces, and with other particles. Information is kept on the position, velocity, and internal energy of each particle. Each simulation particle represents a large number of real gas particles and thus the DSMC method is subject to statistical fluctuations several orders-of-magnitude larger than in the corresponding real gas. In general, the scatter follows a Poisson distribution with a standard deviation proportional to the inverse square root of the sample size (number of simulated particles per sample volume).

The DSMC method consists of two basic steps. The first step is to move all particles along their current velocity vectors for a timestep Δt without any collisions. This step also includes particles entering through computational boundaries, colliding/reflecting off surfaces, as well as particles leaving the computational domain. The second step is to randomly select pairs of particles within the same

computational cell to undergo collisions. The angle at which particles scatter after a collision is found to be inconsequential [1] and is therefore chosen randomly within a DSMC simulation. Using post-collision trajectories and the conservation of momentum and energy, the post-collision velocity vectors are computed assuming an elastic collision. When energy is transferred between translational and rotational modes, the Larsen-Borgnakke model [17] is used to compute inelastic collisions involving translational-rotational energy transfer. These two steps (movement and collision) are repeated until the solution reaches a steady-state. After reaching steady-state, the microscopic particle properties are sampled over many timesteps in order to provide low-scatter macroscopic hydrodynamic flow properties in each computational cell. In addition, the rate of momentum and energy transferred to each surface element by simulation particles is also sampled over many timesteps in order to provide local heating, pressure, and shear-stress distributions. Surface boundary conditions are implemented by particles being randomly scattered after colliding with a surface. In addition, the magnitude of a particle's post-surface-collision velocity is sampled from a Maxwellian distribution (Eq. 2.4) centered at zero bulk velocity and based on the surface temperature specified for the simulation. This type of boundary condition is termed diffuse reflection and full thermal accommodation.

The number of particles selected to collide within each computational cell within the collision step of the DSMC method is chosen to statistically match the local macroscopic collision rate of the gas. The probability P of a collision between two simulated particles within a computational cell over the time interval Δt, is equal to the ratio of the volume swept out by their total cross-section (σ_T) moving at the relative speed between them (c_r) to the volume of the cell [1],

$$P = W\sigma_T c_r \Delta t / V_{cell} \ . \tag{2.10}$$

Here, W is the number of real gas particles represented by each simulated particle and V_{cell} is the cell volume. This probability is used to locally determine the number of collision pairs to form in each computational cell. Specifically, Bird's no-time-counter (NTC) scheme is used to select collision pairs [1]. Once the number of required collisions is determined, particle collision pairs are randomly selected within each cell and elastic or inelastic collisions are performed for each pair chosen to collide. As seen in Eq. 2.10, the collision probability is a function of the collision cross-section of the particles. As mentioned previously, the collision cross section is typically a function of the relative speed c_r as well as the molecular/atomic potential of the colliding particles. The hard-sphere (HS) cross-section, given by $\sigma_{HS} = \pi d_{ref}^2$, assumes an infinite potential at a distance equal to the particle's reference diameter, d_{ref}, and no potential anywhere else. This is a simple billiard-ball type of collision model, which is not a function of c_r and is not able to reproduce the correct temperature dependence of viscosity observed in real gases. In order to maintain the simplicity of the HS model but include the dependence on c_r, Bird introduced the variable hard-sphere (VHS) collision cross-section. A VHS particle is equivalent to a HS particle with a diameter d that is a function of c_r through a power law,

$$d = d_{ref}(c_{r,ref}/c_r)^{\omega-\frac{1}{2}} \ . \tag{2.11}$$

Since the relative velocity between particles in a collision is directly related to the local gas temperature, the VHS model allows for a new temperature dependence of the viscosity. The VHS model thus allows for a realistic fluid viscosity to be modelled within a DSMC simulation. Through Chapman-Enskog analysis, the effective macroscopic viscosity model resulting from the VHS collision model within a DSMC

simulation is given by [1]:

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{\omega} , \qquad \mu_{ref} = \frac{15\sqrt{\pi m k T_{ref}}}{2\pi d_{ref}^2 (5-2\omega)(7-2\omega)} . \tag{2.12}$$

For all DSMC simulations presented in this thesis, the power law exponent, ω, is set equal to 0.75. For consistency, this exact viscosity model (Eq. 2.12) is employed in all NS simulations presented in this thesis.

Each of the two major steps in the DSMC method introduce important computational limitations. If particles are translated without colliding for a timestep Δt, then this timestep must remain less than the mean-free-time τ_c. If $\Delta t > \tau_c$, then the collision probability (Eq. 2.10) may become larger than one. This signifies that all particles should have undergone at least one (possibly more than one) collision during time Δt. Thus the local timestep in a DSMC simulation is limited by the local mean-free-time. If a constant DSMC timestep is employed, its value is limited by the smallest mean-free-time found in the flow field. Next, if particles are randomly paired with other particles residing in the same computational cell for collisions, the cell dimension must be less than the local mean-free-path λ. If cell sizes are larger than λ then particles separated by many mean-free-paths may be selected for a collision. This is physically incorrect and introduces numerical diffusion into the solution. Thus both the timestep and cell size in a DSMC simulation are limited to the order of τ_c and λ respectively. Finally, it should be noted that deterministic solution of the Boltzmann equation involves the same restrictions. These limitations of the DSMC method will be discussed in greater detail throughout the remaining chapters of this thesis. Only a summary of the DSMC method is presented in this chapter, however, complete details of the DSMC method can be found in Ref. [1]. In addition, a thorough review of code validation studies for the DSMC method can

be found in Ref. [18].

2.4 Hybrid Numerical Methods for Multi-Scale Gas Flows

Hypersonic vehicles travelling at high altitudes experience conditions ranging from highly rarefied to fully continuum flow. Even within a mostly continuum flow, there may be local regions of rarefied (or non-equilibrium) flow generated by rapid expansion in the wake of a vehicle as well as by strong gradients in shock waves and boundary layers. Separated flow caused by shock-boundary layer interactions can be greatly influenced by non-equilibrium effects as can the flow over sharp leading edges. Ultimately, it is the flow conditions near the vehicle surface and in the wake that determine the drag and heat transferred to the vehicle and its payload. Thus it is important that these regions be simulated accurately using an appropriate physical model. For example, the continuum Navier-Stokes (NS) equations assume collisional near-equilibrium. More specifically, the velocity distribution function (vdf) of gas particles at every point in the flow is assumed to deviate only slightly from an equilibrium (Maxwell-Boltzmann) distribution. As previous described, this perturbed equilibrium distribution is of a special form called a Chapman-Enskog distribution. The NS equations deal only with the macroscopic representations of this distribution, such as the density, temperature, and bulk velocity of the gas. The heat and momentum transferred to the vehicle are then calculated using gradients of these macroscopic values at the vehicle surface. However, in high-speed, high-altitude environments there are regions in the flow that do not reach collisional equilibrium. For example, the rapid flow expansion around a blunt-body creates a low-density wake region where particles collide less frequently. Likewise, while traversing strong shock waves, thin boundary layers, and sharp leading edges, high-speed gas particles

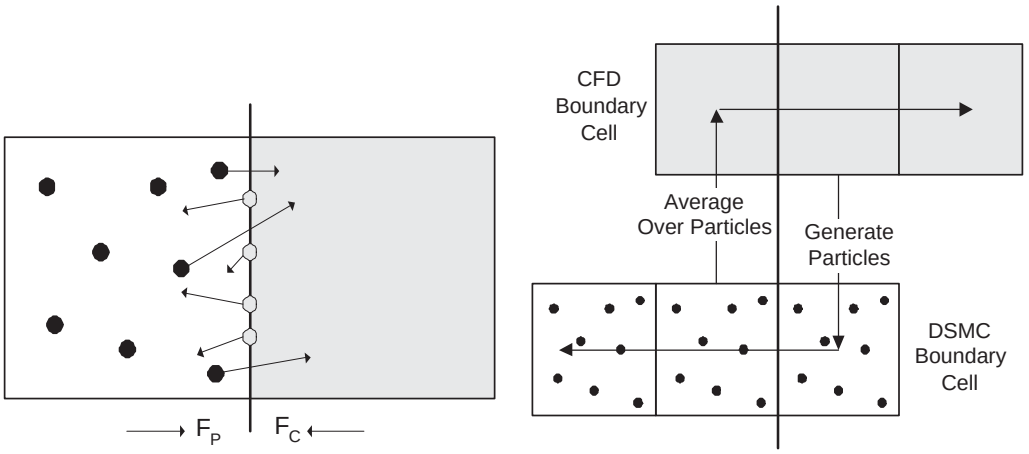
may not undergo a sufficient number of collisions to fully "equilibrate". As a result, gas particles in these non-equilibrium regions do not have a Chapman-Enskog velocity distribution and thus can not be accurately characterized by the NS equations. Macroscopically, such non-equilibrium flow conditions may result in both temperature and velocity slip at the vehicle surface as well as changes in flow separation regions, all of which can significantly influence the heat and momentum transfer to a vehicle. However, although a kinetic description may be required in such localized non-equilibrium regions, a kinetic description of the entire flow field may be computationally prohibitive, even using current large-scale computing facilities. For this reason, during the past decade, significant efforts have been made to develop hybrid kinetic-continuum numerical methods.

2.4.1 Unified methods

In contrast with the NS equations, since no assumption about the form of f is made, the BTE is physically accurate for dilute gases under all conditions ranging from continuum to free-molecular flow. For this reason, various researchers have used the BTE as the basis for numerical simulations of gas flows that contain a mixture of continuum and rarefied conditions. For example Kolobov *et al.* propose a "unified" continuum/kinetic method [19, 20] where the collisionless step of the method (the left-hand-side of Eq. 2.1) is identical in both continuum and non-equilibrium regions. Only the calculation of the collision integral, Eq. 2.2, is different. In non-equilibrium regions a full integration of Eq. 2.2 over all velocity space is performed using various numerical schemes such as a Monte Carlo evaluation. In the continuum regions, calculation of the collision integral is replaced by a "Maxwellization" of the distribution function, f. This simplification is reported as being more than an order of magnitude

less computationally expensive than a Monte Carlo evaluation of the integral using 100 random trials [19]. As a result, the computational time spent on the continuum regions is insignificant compared to that spent on the non-equilibrium regions. Another approach proposed by Le Tallec and Mallinger is to use a BTE solver in non-equilibrium regions and a NS solver for cells in continuum regions. Information transfer is made consistent by matching the half-fluxes at the interface between the BTE and NS solvers. The authors apply this method to 2D hypersonic flow of a monatomic gas over a flat plate and ellipse [21]. Although these methods are more efficient than solving the BTE throughout the entire flow field, it is important to note that solving the BTE is computationally intensive to begin with. As previously discussed, for 3D, multi-species flows, the high dimensionality becomes extremely computationally demanding and spatial resolution is severely limited. Faced with these computational constraints, one must also decide how to set the bounds and resolution of velocity space as well as rotational and vibrational energy space.

An alternate "unified" approach is an all-particle hybrid numerical method. Such methods model non-equilibrium regions using full DSMC simulation, however they also attempt to model continuum regions with the movement and possibly collisions of simulation particles. In order to match the efficiency achieved by conventional CFD methods, such equilibrium particle methods used in continuum regions must be able to use computational cells and timesteps that are not restricted to λ and τ_c. To date, such equilibrium particle methods, such as the method proposed by Tiwari and Klar [22], introduce numerical dissipation that scales with the cell size. More recently, a low-diffusion equilibrium particle method has been proposed by Burt and Boyd [23] in which particles are effectively convected along streamlines within continuum regions. The random thermal motion of particles within continuum



(a) Flux-based coupling. (b) State-based coupling.

Figure 2.1: Typical hybrid coupling procedures.

regions is suppressed, which reduces the diffusion significantly. Additional research is required to properly model both heat and momentum transfer accurately and employ more advanced physical models within such equilibrium particle methods. However, the major benefit of such all-particle hybrid methods is that information transfer between continuum regions and non-equilibrium regions is naturally accommodated.

Finally, moment methods, which are also referred to as extended hydrodynamics, attempt to extend the validity of the continuum formulation for higher Knudsen number flows. By taking moments of the BTE and modelling each moment with its own transport equation, the resulting sets of partial differential equations are physically valid at higher Knudsen numbers and can still be solved efficiently from techniques of CFD. As an example, research into the accuracy and efficiency of numerically solving the 10 moment equations for high Knudsen number flows is currently being undertaken [24, 25].

2.4.2 Coupled particle-continuum methods

The DSMC method effectively solves the BTE (Eq. 2.1) by directly simulating the movement and collision of particles within the gas and with the vehicle walls. This allows for temperature and velocity slip to be natural results of a DSMC simulation and the heat and momentum flux to the vehicle is actually calculated by particle collisions with the surface. A major limitation of DSMC is that the cell size must be on the order of the mean free path (λ) and each DSMC cell must contain at least 20 simulated particles for statistical reasons [1]. As a result, 2D and 3D DSMC simulations can require prohibitively high numbers of computational cells and therefore simulated particles, especially in regions where λ is very small. However, it is precisely in these regions where collisional equilibrium is reached and the continuum equations can be solved. For example, Pareschi and Caflisch present a novel method [26] where f is represented as a combination of a Maxwellian distribution (f_0) and a non-equilibrium distribution of DSMC particles (g); $f = \beta f_0 + (1 - \beta)g$. In regions approaching collisional equilibrium particles are removed from the DSMC simulation and more weight is transferred to the Maxwellian distribution. Thus, in the continuum limit, the method essentially becomes an Euler solver.

An alternate approach taken by many researchers is to directly couple DSMC with traditional NS solvers. Unlike numerical methods based on the BTE, this approach involves coupling two entirely different methods, one deterministic in nature and one statistical. Reference [27] presents a discussion of the major considerations involved as well as a summary of published work on such coupled schemes. One important aspect is the transfer of information between particle and continuum cells and the inherent statistical scatter in that information. Information transfer is typically handled using one of the two methods depicted in Fig. 2.1. Flux-based coupling,

Fig. 2.1(a), involves calculating the fluxes of mass, momentum, and energy at the interface according to the particle cell (F_P), and according to the continuum cell (F_C). F_P is calculated by tracking individual simulation particles as they cross the interface whereas F_C is extrapolated using macroscopic gradients in the continuum domain. In general $F_P \neq F_C$ and a modified flux must be computed such that mass, momentum, and energy is conserved within the simulation. This modified flux is then used to update the continuum solution and is also used to create a distribution of particles on the interface which propagate into the particle simulation. On the other hand, state-based coupling (Fig. 2.1(b)) temporarily averages particle information to obtain a macroscopic state on one side of the interface and at the same time generates a distribution of particles from a macroscopic state on the other side of the interface. In this way, the flux into both the continuum domain and particle domain is handled exclusively by its respective solver through standard boundary procedures.

Wadsworth and Erwin developed a strongly coupled, flux-based, hybrid DSMC-NS scheme and applied it to both 1D shocks [28] and 2D rarefied slit flow [29]. In these studies, a Maxwellian distribution was used to generate simulation particles at the interface and the domain boundaries remained fixed. Flux-based coupling was also used by Hash and Hassan to couple the DSMC method with a NS solver for simulations of Couette flow [30] and hypersonic flow over a blunted cone [31]. More recently, Wijesinghe *et al.* also used flux-based coupling to embed a DSMC solver in the finest level of an adaptive mesh and algorithmic refinement (AMAR) scheme for the Euler equations [32]. However, when averaging particle information, the statistical scatter involved in determining F_P in Fig. 2.1(a) is much higher than that associated with averaging to obtain a macroscopic state [33] as seen in Fig.

2.1(b). Additionally, as discussed in Ref. [34], fluxes normal to the interface may become very small in regions where the interface is aligned with the flow direction. As a result, the statistical scatter inherent in the flux quantities becomes even more pronounced; a significant disadvantage of flux-based coupling.

Roveda *et al* used state-based coupling to combine the DSMC method with an Euler solver to simulate moving 1D shock waves [35] and 2D unsteady slit flow [36]. Although use of state-based coupling involved less statistical scatter, their method was time-accurate and particle information was averaged at each timestep. This significantly constrained the number of samples used for the average. In order to deal with this, the authors employed a novel algorithm to effectively "clone" particles near the interface in order to reduce the statistical scatter transferred to the continuum domain. Wang and Boyd used the Information Preservation (IP) scheme [37] which, in addition to microscopic information, also preserves macroscopic information for each DSMC simulation particle in order to reduce the statistical scatter. Although successful for certain 2D flows [38], when applied to 1D normal shock waves, it was found that the IP scheme produced an incorrect post-shock state and a shock wave that was too thin. A new formulation for the IP energy flux [39] was able to remedy these problems somewhat, however, at large computational expense.

Recently, a loosely-coupled approach similar to the hybrid method described in this dissertation, was used by Wu *et al.* [40] to simulate hypersonic flow over a wedge as well as expansion of nitrogen gas from a 3D nozzle into a near-vacuum. The authors use a shell script to couple existing DSMC and NS codes. The resulting numerical cycle differs from the algorithm described in this thesis in a subtle but important manner regarding *when* information is transferred between DSMC and NS regions; an issue which is discussed further in Chapter V. Finally, while their

hybrid solution for the wedge flow agreed well with full DSMC flow field results, the simulation was reported to take longer than a pure DSMC simulation.

CHAPTER III

A Hybrid Numerical Method for Steady-State Hypersonic Flows

A coupled particle-continuum approach has one significant advantage over a unified approach. The DSMC method used to simulate the Boltzmann equation and the CFD methods used to solve the Euler or Navier-Stokes equations are mature numerical methods that have been optimized and validated with experimental data for decades. On the other hand, a unified approach involves either solving the continuum equations with a particle method, or solving the Boltzmann equation with a deterministic method; both of which currently lack the accuracy and efficiency achieved by DSMC and CFD methods and also lack much of the physical modelling required to simulate hypersonic flows of engineering interest. A coupled particle-continuum (DSMC-NS) approach is able to utilize the latest DSMC and CFD techniques and physical models to ensure that the most accurate and efficient numerical method is used in both continuum and non-equilibrium regions of the flow. Existing, state-of-the-art, DSMC and NS codes can then be incorporated (un-modified) as modules within a hybrid code where they can be used with confidence. This is the approach taken for the hybrid method developed in this thesis and the resulting *M*odular *P*article-*C*ontinuum numerical algorithm is termed the MPC method.

In this chapter, the various components incorporated within the MPC method are described. Such components include the specific numerical methods used to solve both DSMC and NS regions, prediction of continuum breakdown, mesh refinement procedures, and a procedure that transfers information between particle and continuum regions. The averaging procedures and particle generation procedures used to facilitate information transfer are also described. In general, these procedures have been developed by previous researchers through prior work on hybrid methods. The components described in this chapter have thus been adopted from the literature and incorporated into the MPC numerical method.

3.1 Particle and Continuum Numerical Methods

DSMC code

Particle regions are simulated using MONACO [41], a general, cell-based implementation of the DSMC method [1] that statistically simulates the Boltzmann transport equation (Eq. 2.1). MONACO is able to simulate 2D, axi-symmetric, and 3D flows using unstructured grids fitted to complex geometries and is able to run on large parallel computing architectures. For the results presented in this thesis, MONACO employs the variable hard sphere (VHS) collision model, which results in the macroscopic viscosity model shown previously in Eq. 2.12. Bird's no time counter (NTC) algorithm [1] is used to compute collision probabilities. For simulations of diatomic gases, energy exchange between translational and rotational modes is determined using the Larsen-Borgnakke model [17], modified by the variable rotational energy exchange probability model of Boyd [42, 43]. For the simulations presented in this thesis, the reference temperature for rotational energy exchange is specified as 91.5 K and the maximum rotational collision number as 18.1. Energy

transfer to vibrational modes is not considered. Complete details of the MONACO code can be found in Ref. [41].

Navier-Stokes solver

Continuum regions are simulated using the Michigan Aerothermodynamic Navier-Stokes (LeMANS) solver [44, 45, 46]. For the results presented in this thesis, it is assumed that rotational and translational energy modes can be described by a single temperature T in continuum regions. The vibrational energy mode is not considered. The resulting governing equations are the laminar, compressible, Navier-Stokes (NS) equations previously detailed in Chapter II. The viscosity in continuum NS regions is modelled using Eq. 2.12 in order to match exactly the viscosity model used in DSMC regions. This is important not only at particle-continuum interfaces (for information transfer) but also whenever comparing DSMC and NS flowfield results and surface properties. LeMANS solves this set of equations on a general unstructured mesh using a finite-volume formulation. LeMANS is able to simulate 2D, axi-symmetric, and 3D flows on large parallel computing architectures. Time integration of the governing equations is performed using a point-implicit method. The viscous terms are calculated using the values of properties at the cell centers and at the nodes. The inviscid fluxes between the mesh volumes are discretized using a modified form of the Steger-Warming Flux Vector Splitting [47]. A major difficulty in solving the NS equations for hypersonic flows comes from the fact that the inviscid flux terms lead to numerical instabilities in the vicinity of strong shock waves. In order to suppress these instabilities, a certain amount of numerical dissipation must be added to the flux function in order to obtain a solution of the shock wave in a stable manner. Unfortunately this numerical dissipation can have adverse affects throughout the

remainder of the flowfield, especially in boundary layer and wake regions. The original Steger-Warming flux function uses a simple upwind-based stencil that is highly dissipative. The modified Steger-Warming flux function effectively extends the stencil and significantly reduces the numerical dissipation. The modified form is thus adequate to calculate boundary layer and wake regions. A pressure switch is used to revert back to the original Steger-Warming flux function only in the vicinity of strong shock waves where it is able to eliminate numerical instabilities and obtain a converged solution. The modified Steger-Warming flux function is outlined in Refs. [47, 48, 44]. Complete details of the numerical algorithms used in the LeMANS code are contained in Ref. [49].

3.2 Components of a Coupled Particle-Continuum Method

3.2.1 Continuum breakdown

The MPC method begins with a solution to the NS equations for the geometry and flow conditions under investigation. This initial continuum solution may be incorrect in non-equilibrium regions. The MPC method should detect this and proceed to improve the solution using DSMC. The efficiency of a hybrid particle-continuum method comes from restricting use of the DSMC method to only that portion of the flowfield where large non-equilibrium effects are felt. However, the interface between particle and continuum regions must still lie in a region that is accurately modelled by the continuum NS equations. Ensuring this is a vital aspect of any hybrid method. Typically, particle and continuum regions are determined by evaluating a measure of continuum breakdown in each computational cell. The MPC method computes a continuum breakdown parameter called the gradient-length Knudsen number

$$Kn_{GL-Q} = \frac{\lambda}{Q}|\nabla Q| \ , \tag{3.1}$$

where Q represents a local flow quantity such as density (ρ), translational temperature (T_{TRA}), or bulk velocity magnitude ($|\mathbf{V}| = \sqrt{u^2 + v^2}$). In low speed regions of the flow (where $|V| \rightarrow 0$), it is useful to normalize the velocity gradient by the local speed of sound, a. This can be implemented by setting the denominator $Q = \max(\ |V|,\ a\ )$. The local mean-free-path is calculated using the macroscopic properties in each cell as

$$\lambda = \frac{2(5-2\omega)(7-2\omega)}{15}\sqrt{\frac{m}{2\pi kT}}\left(\frac{\mu}{\rho}\right)\ , \tag{3.2}$$

where μ is the local viscosity computed using Eq. 2.12. This expression for λ is consistent with the VHS viscosity model and all variables have the same values as previously outlined in Eq. 2.12. The final value used to quantify the degree of local continuum breakdown is then taken as the maximum for the flow quantities of interest,

$$Kn_{GL} = \max(Kn_{GL-\rho}, Kn_{GL-T_{TRA}}, Kn_{GL-|\mathbf{V}|})\ . \tag{3.3}$$

For hypersonic flows, it has been shown that in regions of the flowfield where $Kn_{GL} < 0.05$, the discrepancy between a NS and DSMC solution is less than 5% [50, 51]. Thus, these regions could be modelled using the NS equations with little error.

The MPC method begins with a full NS solution to the problem and uses this continuum solution to initially predict where the flow may be in non-equilibrium. Figure 3.1 (top) plots the value of Kn_{GL} from Eq. 3.3 applied to a NS solution of Mach 6 flow of N_2 gas over a 2D cylinder where the global Knudsen number $Kn_g = 0.01$. The continuum breakdown parameter is seen to be largest in the bow shock, fore-body boundary layer, and in the near wake region. Cells in which $Kn_{GL} > 0.05$ are labelled as DSMC cells and the remainder are labelled as NS cells. This results in the initial interface (dashed line) shown in the bottom of Fig. 3.1. This

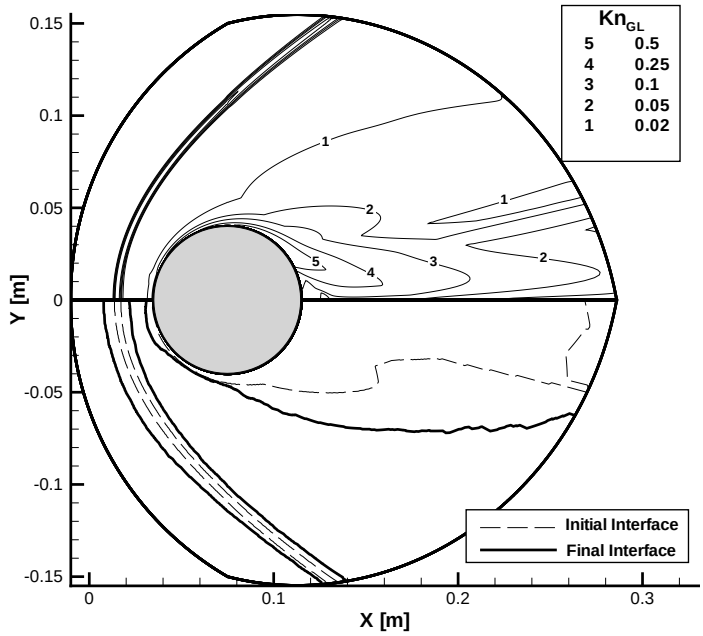


Figure 3.1: Continuum breakdown and interface locations.

interface corresponds to the $Kn_{GL} = 0.05$ contour in the top of Fig. 3.1 after a simple smoothing algorithm has been applied to the interface. This smoothing algorithm is detailed in Algorithm [16] in Appendix A. It is important to note that since the Kn_{GL} parameter and the cutoff value of 0.05 are somewhat empirical themselves, that any slight modification to the interface location due to the smoothing algorithm is acceptable. As long as the interfaces are placed conservatively (i.e. they remain within near-continuum regions), their precise location is not important for accuracy. As the MPC simulation progresses, the continuum breakdown parameter is applied periodically to the current solution in order to update the interface locations. As an example, the final steady-state interface locations are also shown in the bottom of Fig. 3.1 for this particular case.

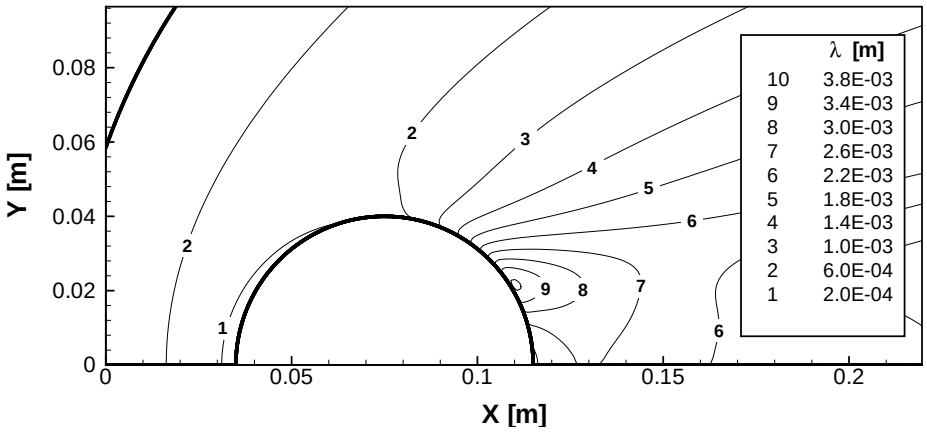


Figure 3.2: Variation of mean-free-path for a Mach 6, $Kn_g = 0.01$ flow.

3.2.2 Mesh refinement

The current implementation of the MPC method requires an initial solution to the NS equations obtained on a quadrilateral mesh. Although research into the use of triangular (or tetrahedral) meshes for hypersonic flows is on-going, quadrilateral (or hexahedral) meshes are currently required by CFD methods to accurately resolve the very strong gradients found in shock waves and boundary layers of hypersonic flows [52, 53, 54]. Figure 3.2 plots contours of λ for the same solution as used in Fig. 3.1 (a Mach 6 flow of N_2 where $Kn_g = 0.01$). In addition to its use in determining continuum breakdown, λ is also a measure of the spatial resolution necessary in DSMC regions. The bow shock is clearly defined in Fig. 3.2 by contour number 2 and λ is seen to be more than an order of magnitude smaller in the fore-body stagnation region compared to the near wake region. As previously outlined, accurate DSMC simulation requires that the local computational cell size be less than the local value of λ. In order to accomplish this, the cell-centered value of λ is compared to the length of each cell face in order to determine the refinement required in each cell of the initial continuum NS mesh. This restriction is most important in the direction

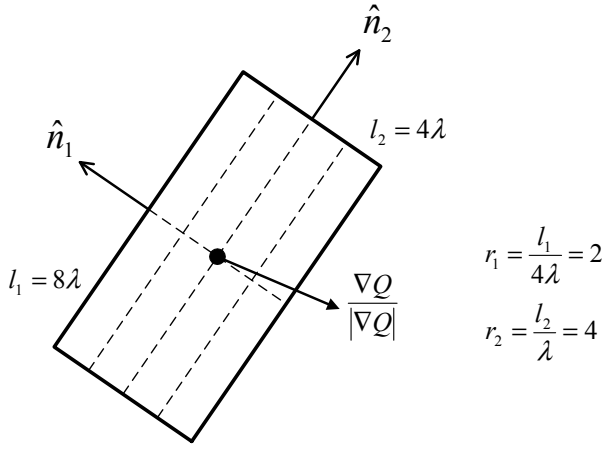


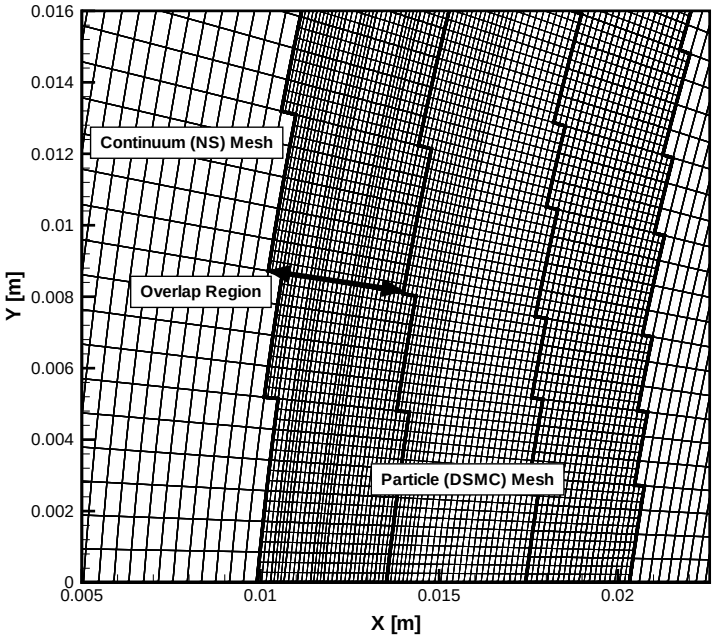
Figure 3.3: Mesh refinement procedure for each continuum cell.

of the flow gradient, however, and it becomes less important in the direction normal to the gradient. The mesh refinement procedure used by the MPC method restricts the cell size in the direction of the gradient to λ and allows the cell size in the direction normal to the flow gradient to be as large as 4λ. The refinement procedure implemented in the MPC method is depicted in Fig. 3.3 where the refinement factor for each cell face (r_f) is calculated as

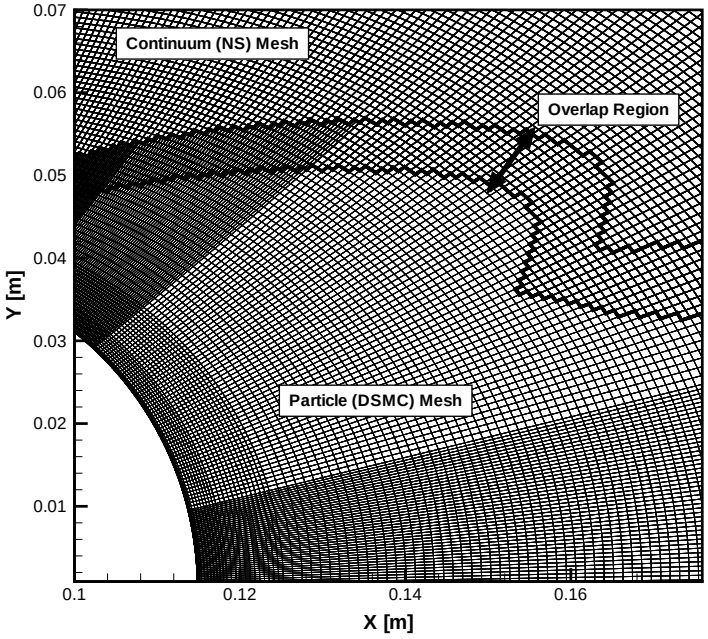
$$r_f = l_f \times \left[F\lambda \left(3 \left| \hat{n}_f \cdot \frac{\nabla Q}{|\nabla Q|} \right| + 1 \right) \right]^{-1} . \tag{3.4}$$

Here, l_f is the length of the cell face which is compared to a multiple of the mean-free-path, λ. The dot product between the face unit normal vector, $\hat{n}_f$, and the unit gradient vector determines the alignment of the cell face with the direction of the gradient. If the cell face is perfectly aligned with the flow gradient (which means $\hat{n}_f$ is *normal* to the flow gradient), the dot product in Eq. 3.4 is zero, and the refinement is scaled to $F\lambda$. Whereas, if the cell face is perpendicular to the flow gradient (meaning $\hat{n}_f$ is *parallel* to the gradient vector), the refinement is relaxed to $4F\lambda$. This is depicted in Fig. 3.3 where more refinement is added in the direction of

the flow gradient. Since the refinement is required because of continuum breakdown, for consistency, the gradient in Eq. 3.4 is calculated using the same flow quantity (ρ, T_{TRA}, or $|V|$) that resulted in the maximum value of Kn_{GL} in that cell (Eq. 3.3). Finally, the factor F in Eq. 3.4 allows scaling of the entire mesh by any fraction (or multiple) of the local mean-free-path. The entire expression is rounded to the nearest integer to obtain r_f. Since opposite faces in a quadrilateral cell must be refined equally, each cell will contain only two integers (r_1 and r_2) specifying the minimum required refinement in orthogonal directions. The current DSMC module also requires that each cell face border only one neighboring cell. As a result, the refinement can not be set arbitrarily in each cell but must be consistent with the refinement in neighboring cells. A simple algorithm is employed that ensures all cells are refined consistently while still meeting the minimum required refinement. The above refinement procedure is performed only once at the beginning of an MPC simulation using the initial NS solution. Values or r_1 and r_2 are stored for each cell in the continuum NS mesh. As DSMC regions are created during an MPC simulation, these stored values are used to create new refined DSMC cells. For each continuum cell labelled as a DSMC cell, $r_1 \times r_2$ refined cells are created and added to the DSMC module's data structure. This creates a separate mesh refined to the mean-free-path to be used by the DSMC module. For the same flow used for Figs. 3.1 and 3.2, the above procedure results in the mesh shown in Fig. 3.4 at the start of the simulation. Here the refined DSMC regions correspond to the initial interface locations shown in the bottom of Fig. 3.1. The mesh refinement in both the shock and wake regions is shown in Figs. 3.4(a) and 3.4(b) respectively. The refinement inside the shock varies from 6 to 12 DSMC cells for each NS cell. In the wake, as the flow expands and λ becomes large, a corresponding reduction in the refinement level is clearly seen



(a) Mesh refinement in the bow shock.



(b) Mesh refinement in the cylinder wake.

Figure 3.4: NS and refined DSMC meshes for a Mach 6, $Kn_g = 0.01$ flow.

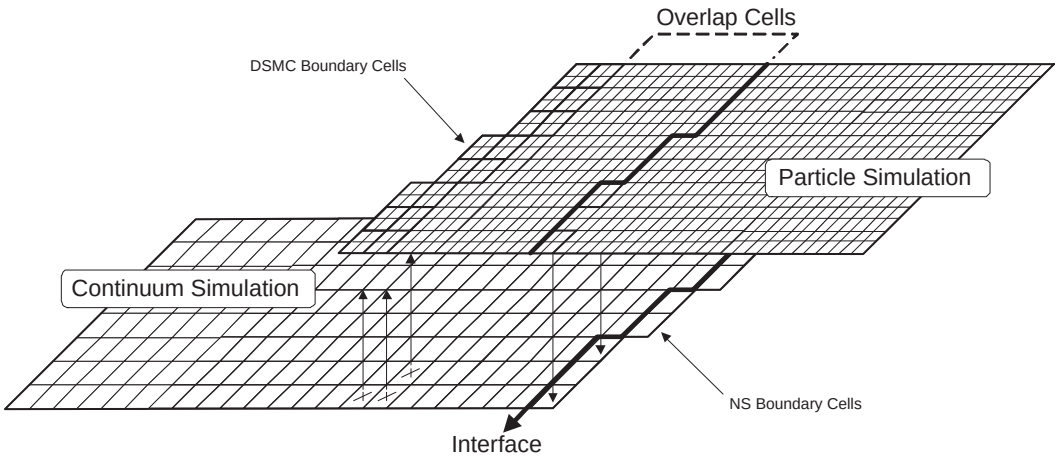


Figure 3.5: State-based information transfer.

in Fig. 3.4(b). The refinement in the wake region varies from 4 cells to 1 (or no refinement).

3.2.3 Information transfer

The MPC method uses state-based coupling to transfer information between particle and continuum regions as detailed in Fig. 3.5. This type of coupling has been used by many researchers [35, 36, 38, 55, 40] in the literature and is sometimes referred to as the Schwarz alternating method. After application of the breakdown parameter (Kn_{GL}), particle regions are immediately extended a number of cells further into the continuum region creating an overlap region. The number of overlap cells, n_{overlap}, is a user defined input. The purpose and required size of this overlap region will be discussed in Chapters IV and V when the hybrid numerical cycle is described. After creating the overlap region, cells bounding the NS domain are labelled as NS boundary cells and cells bounding the DSMC domain are labelled as DSMC boundary cells. The number of DSMC boundary cells (n_{DSMCbcs}) and the number of NS boundary cells (n_{NSbcs}) are also user defined inputs. However, as depicted in Fig.

3.5, these are generally set to 2 and 1, respectively. The reason that more than one DSMC boundary cell may be necessary stems from the fact that in a DSMC simulation occasionally particles may move across multiple cells during a single timestep. In that case, particles may enter the first interior DSMC cell not only from immediate neighbors but also from cells further away. If only one DSMC boundary cell is used, the first interior DSMC cell may receive a slightly incorrect flux from the boundary during each timestep. Although particles occasionally traverse two cells in a single DSMC timestep, it becomes exceedingly unlikely that they traverse three or more cells. Hence, two DSMC boundary cells are typically employed and adding a third produces no noticeable difference. It should also be noted that since the breakdown parameter is applied at the continuum level, cell-types are assigned to the continuum mesh. All refined DSMC cells generated within a single NS cell (as portrayed in Fig. 3.3) are then assigned the same cell-type. Thus as seen in Fig. 3.5, an overlap region of $n_{\text{overlap}} = 6$ continuum cells will contain 12 DSMC cells if the mesh is refined by a factor of 2. Likewise, setting $n_{\text{DSMCbcs}} = 1$ will result in 2 refined DSMC boundary cells in each direction which is also depicted in Fig. 3.5. Note that when significant mesh refinement is employed, setting $n_{\text{DSMCbcs}} = 1$ will still involve multiple refined boundary cells which are sufficient to ensure the correct flux into the first interior DSMC cell.

Now that all regions of NS and DSMC cells (including boundary cells) have been defined, the regions must be coupled by transferring information across the interface. When using state-based coupling this simply involves updating the boundary conditions in either the particle or continuum mesh. In this way, information transfer into both the particle and continuum regions is handled through existing procedures already used by both numerical methods. As will be fully described in Chapter VII,

the two overlapping meshes depicted in Fig. 3.5 are stored separately within the hybrid code. The NS mesh is stored in the data structure of the NS module and the DSMC mesh is stored in the data structure of the DSMC module. In this way, both modules are able to operate on their respective regions without modification and only the information within these data structures (inside the boundary cells) needs to be updated periodically. This process is depicted with vertical arrows in Fig. 3.5. On one side of the interface, updating DSMC boundary conditions involves generating a distribution of particles within DSMC boundary cells based on the corresponding NS solution. On the other side of the interface, updating NS boundary conditions involves averaging over particle information within refined DSMC cells to obtain a macroscopic state.

3.2.4 Particle generation

During DSMC simulation in non-equilibrium regions, each DSMC boundary cell is continually filled with particles consistent with the flow properties in the corresponding NS cell. For each timestep, all simulation particles in the DSMC boundary cell are first deleted and then re-generated based on current NS information. The number of new particles is determined directly from the NS cell density, the volume of the computational cell, and the simulation particle weight W. This weight W is defined as the number of real gas particles represented by each simulation particle. W is typically the same for all particles in a given cell, but may vary from cell to cell. Particles are then randomly distributed in space throughout the refined DSMC cells residing within the boundary cell. The velocities of these newly generated particles are sampled from a Chapman-Enskog velocity distribution [56] based on the local macroscopic state and gradients, known from the NS solver. The same distribution

is therefore used to generate particles in all refined DSMC cells within a boundary cell. The boundary cells then become an extension of the DSMC domain for one standard DSMC cycle. The Chapman-Enskog velocity distribution is a perturbed Maxwell-Boltzmann distribution which is presented in Ref. [56] as

$$f(\mathcal{C}) = \Gamma(\mathcal{C}) \frac{1}{\pi^{3/2}} e^{-\mathcal{C}^2} \quad . \tag{3.5}$$

The normalized thermal velocity is given by $\mathcal{C} = \mathbf{C}/(2kT/m)^{1/2}$ where $\mathbf{C}(C_x, C_y, C_z)$ is the thermal velocity of the particle, k is the Boltzmann constant, T is the local temperature of the gas and m is the particle mass. For the 2D and axi-symmetric simulations presented in this thesis, the perturbation term simplifies to

$$\begin{aligned} \Gamma(\mathcal{C}) \quad = \quad & 1 + \tfrac{1}{p} \Big[\sqrt{\tfrac{2m}{kT}} (q_x \mathcal{C}_x + q_y \mathcal{C}_y)(\tfrac{2}{5}\mathcal{C}^2 - 1) \\ & -2\tau_{xy}\mathcal{C}_x\mathcal{C}_y - \tau_{xx}(\mathcal{C}_x^2 - \mathcal{C}_z^2) - \tau_{yy}(\mathcal{C}_y^2 - \mathcal{C}_z^2) \Big] \quad . \end{aligned} \tag{3.6}$$

Here p is the pressure, τ_{ij} are components of the shear stress tensor, and q_i are components of the heat transfer vector. Calculation of the shear stress tensor and heat transfer vector (see Eqs. 2.6 and 2.7) are standard procedures involved in the computation of the viscous fluxes during each timestep of a NS simulation. Thus, the Chapman-Enskog distribution is completely defined by available NS information. As outlined in Ref. [56], thermal velocities are then sampled from this distribution (Eq. 3.5) using an acceptance-rejection scheme and super-imposed on the bulk velocity resulting in particle velocities $\mathbf{c}(v_x, v_y, v_z)$ with components

$$v_x = u + C_x \quad , \qquad v_y = v + C_y \quad , \qquad v_z = w + C_z \quad . \tag{3.7}$$

The internal rotational energy of each particle is sampled from a Boltzmann (equilibrium) distribution based on the single temperature T in the corresponding NS cell. As described in Chapter II, the NS equations inherently assume a Chapman-Enskog

velocity distribution and thermal equilibrium between translational and rotational energy modes. These assumptions are physically accurate in near-equilibrium regions and thus as long as particle-continuum interfaces remain in near-equilibrium regions the imposed DSMC boundary conditions are also physically accurate. Finally, in addition to generating particles within DSMC boundary cells, occasionally within an MPC simulation particles must be generated in newly created DSMC regions. For example, at the beginning of an MPC simulation, particles must be generated throughout the entire initial DSMC regions setup by the breakdown parameter. Thus, initially, all particles in DSMC regions will be drawn from near-equilibrium (Chapman-Enskog) velocity distributions; consistent with the NS solution to the problem.

3.2.5 Averaging techniques

As DSMC regions are allowed to iterate and simulation particles proceed to move and collide with each other, the particle velocity distributions and flow properties should progress and relax towards the correct non-equilibrium result. As particle regions progress, a method of calculating the *macroscopic* variation in each cell is required.

At a given timestep, macroscopic values such as the fluid density, velocity, translational temperature, and rotational temperature can be calculated by simply averaging over the microscopic properties of all particles contained within a computational cell. Macroscopic properties are desired at the continuum level and thus the average is taken over all refined DSMC cells residing within the larger NS cell. As mentioned previously, different DSMC cells may have different cell-weighting factors (W). This means that some simulation particles used in the average may represent a different

number of real particles than others. Defining the number of particles in each refined cell (r) as Np_r and the number of refined DSMC cells within each NS cell as Nrc where each refined cell has a weight W_r, the macroscopic properties in each continuum cell at each timestep can be calculated using the following relations:

$$N_{real} = \sum_{r=1}^{Nrc} \sum_{i=1}^{Np_r} W_r \tag{3.8}$$

$$\rho = \frac{mN_{real}}{V_{cell}} \tag{3.9}$$

$$u = \frac{1}{N_{real}} \sum_{r=1}^{Nrc} \sum_{i=1}^{Np_r} v_{x_i} W_r \quad v = \frac{1}{N_{real}} \sum_{r=1}^{Nrc} \sum_{i=1}^{Np_r} v_{y_i} W_r \quad w = \frac{1}{N_{real}} \sum_{r=1}^{Nrc} \sum_{i=1}^{Np_r} v_{z_i} W_r \tag{3.10}$$

$$T_{TRA} = \left(\frac{s-1}{s} \right) \frac{m}{3k} \left[\left(\frac{1}{N_{real}} \sum_{r=1}^{Nrc} \sum_{i=1}^{Np_r} \left(v_{x_i}^2 + v_{y_i}^2 + v_{z_i}^2 \right) W_r \right) - u^2 - v^2 - w^2 \right] \tag{3.11}$$

$$T_{ROT} = \frac{1}{N_{real}} \sum_{r=1}^{Nrc} \sum_{i=1}^{Np_r} e_{ROT_i} W_r \quad . \tag{3.12}$$

Here s is the total number of simulation particles (the number of samples) within a continuum NS cell of volume V_{cell}, and N_{real} is the number of real particles represented these simulated particles. It has been observed that sampling from a small number of DSMC particles consistently depresses the value of the translational temperature. The coefficient $(s-1)/s$ in Eq. 3.11 has been used by other researchers [35, 57] and is used in the current work to correct this error. Finally, e_{ROT_i} is the rotational energy associated with each DSMC particle i.

Figures 3.6 and 3.7 plot the macroscopic velocity and translational temperature evaluated at each timestep in a NS boundary cell for 5000 DSMC timesteps. This particular NS boundary cell contains only one DSMC cell (no mesh refinement) and there are 20 particles within this DSMC cell on average. This simple spatial average of particle information evaluated over the cell volume (denoted by the symbols in Figs. 3.6 and 3.7) suffers from large statistical scatter. The bulk x-velocity (u) is

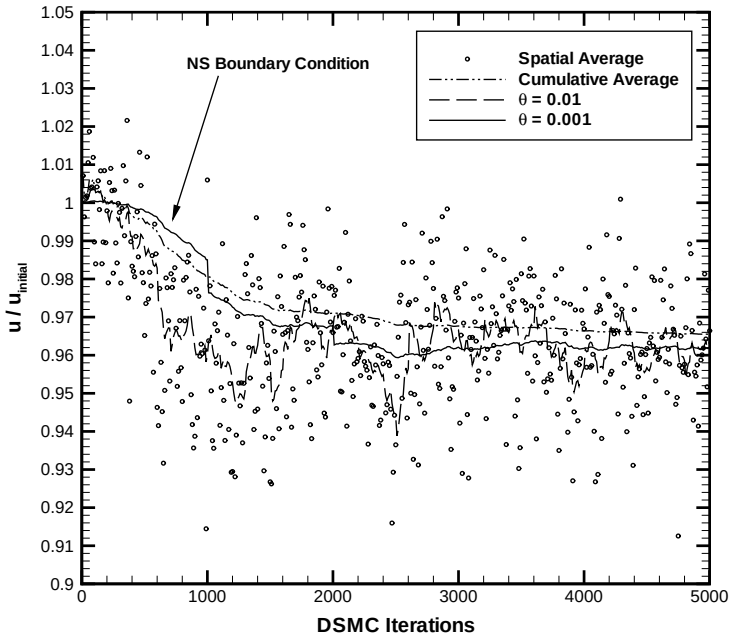


Figure 3.6: Reduction of DSMC velocity scatter for NS boundary condition.

seen to drop by approximately 4% (from 1614 to 1550 m/s) and the temperature to rise by approximately 200% (from 300 to 600 K) over a number of DSMC timesteps. For this case, the magnitude of statistical scatter is seen to completely overwhelm the physical variation. In particular, the velocity is seen to fluctuate by up to 8% and the temperature by up to 100%. Prescribing NS boundary conditions that contain such levels of scatter would induce numerical instabilities in the NS solver. Figures 3.6 and 3.7 also plot the cumulative time-average of the simple spatial average. The inclusion of the history of the flow properties is seen to reduce the scatter almost completely while at the same time introducing a time lag.

Recently, a method for evaluating instantaneous macroscopic hydrodynamic properties during a DSMC simulation has been proposed. The sub-relaxation technique by Sun and Boyd [57] includes the cumulative history using a relaxation factor (θ) and in addition, allows for "old history" to be removed from the average when it begins

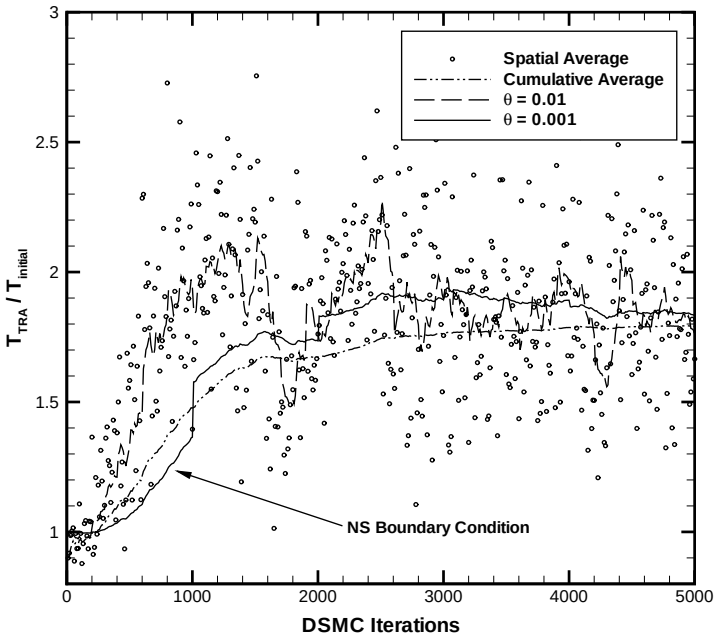


Figure 3.7: Reduction of DSMC temperature scatter for NS boundary condition.

to pollute, or lag, the true variation. If the spatially-averaged macroscopic properties (Eqns. 3.9-3.12) at timestep j are denoted as A_j (i.e. $A = \rho, u, v, w, T_{TRA}, T_{ROT}$), then the sub-relaxation formula for the *temporal* average $\overline{A_j}$ is:

$$\overline{A_j} = (1-\theta)\overline{A_{j-1}} + \theta A_j \tag{3.13}$$

and the correction, which removes the "old history" is given by:

$$\overline{A_j}' = \overline{A_j} + \frac{(1-\theta)^{j-i}}{1-(1-\theta)^{j-i}}\left(\overline{A_j} - \overline{A_i}'\right) \tag{3.14}$$

where j is the current timestep, i is the previous timestep at which a correction was made, and $\overline{A_i}'$ is the temporal average recorded at this previous timestep i. As detailed in Ref. [57], Eqn. 3.14 effectively removes the history before timestep i. In order to maintain stability, this correction is performed only when the coefficient of this correction is between zero and one. This condition is approximately satisfied when $j = \frac{1}{\theta} + i$. Note that the level of scatter associated with a given θ value is the

same as the scatter resulting from a cumulative average taken over $\frac{1}{\theta}$ timesteps [57]. Figures 3.6 and 3.7 show the success of this averaging technique applied to the DSMC velocity and temperature variations, respectively. Although when using $\theta = 0.01$ the average follows the DSMC variation with almost no lag, the scatter is far too large to use this average as a NS boundary condition. Again, large scatter in a NS boundary condition will induce large numerical waves in the continuum domain that will likely cause instability and failure of the hybrid code. However, since the MPC method is designed to solve steady-state flows, a time lag is acceptable. By decreasing θ to 0.001, the scatter is reduced significantly with an acceptable lag. Notice that the sub-relaxation technique reaches the steady state value in approximately 2500 timesteps while the cumulative average still has not reached it in 5000 timesteps. Additionally, in contrast to the cumulative average, the sub-relaxation average behaves smoothly at early timesteps since only a small weight of θ is applied to each successive raw DSMC property. For these reasons, the sub-relaxation technique is used within the MPC method with a value for θ on the order of 0.001. Finally, it should be noted that if mesh refinement is utilized, more refined cells (and therefore more particles) are available when determining the average macroscopic properties in each continuum cell. Thus, mesh refinement lowers the level of statistical scatter further.

These sub-relaxation averages are stored at the continuum level and thus provide a semi-time-accurate measure of the macroscopic variation inside all particle regions with low statistical scatter. As will be outlined in Chapters IV and V along with the hybrid numerical cycle, this information is used to monitor the progression of the hybrid solution and adapt the interfaces during the simulation. Finally, this information is used to periodically set the boundary conditions in the NS solver. It is important to note that the current implementation of the MPC method sets

the entire state in each NS boundary cell. This is physically valid if the flow is supersonic in the direction of information transfer at the boundary. However, in regions of subsonic flow only two state variables should be specified. Since particle-continuum interfaces may pass through complex flow features involving both subsonic and supersonic flow, implementing a general boundary condition in all NS boundary cells is not trivial. In general, specifying the entire state within a NS boundary cell limits the level of convergence; that is, the residual does not reach machine zero. However, the residual that remains in NS regions is still orders-of-magnitude less than both the level of statistical scatter in DSMC regions and the overall accuracy of the hybrid method. Investigation of the accuracy of the MPC method in later chapters will address the effect of this simplified NS boundary procedure.

3.2.6 Hybrid numerical cycle

The above components of the MPC method outline how particle and continuum regions are determined, how interface locations are updated, and how information is transferred across these interfaces within a hybrid simulation. Perhaps the most vital aspect of a coupled hybrid method is the numerical cycle that dictates *when* to update the interfaces and *when* to transfer information. The hybrid numerical cycle must incorporate the above components in such a way as to ensure that the hybrid solution progresses towards the correct non-equilibrium result in a robust manner. The MPC numerical cycle is completely detailed in Chapters IV and V where it is developed in context with the simulation of shock waves and blunt-body flows.

CHAPTER IV

Normal Shock Wave Study

The simulation of a normal shock wave is a very popular validation case for testing new numerical methods designed to solve non-equilibrium gas flows. Shock waves are, by their very nature, regions of highly non-equilibrium flow and high quality experimental data exists that has been used extensively for validation in the literature. Convective forces due to the bulk motion of the gas attempt to form a discontinuity in flow properties given by the Rankine-Hugoniot jump condition [58]. However, this rapid change in flow properties is ultimately accomplished through gas particle collisions that occur over a finite length scale; the mean-free-path λ. As a result, the thickness of a normal shock wave is always on the order of 5 to 20 upstream mean-free-paths, since that is roughly the number of collisions required for gas particles to fully equilibrate to the new conditions. In fact the degree of non-equilibrium inside a shock is only a function of the shock Mach number and not the free-stream density. For a given shock strength, the shock thickness decreases linearly with increasing density, however, so does the mean-free-path [1]. As a result, the gradient length Knudsen number (Eq. 3.1) remains constant and the interior of a strong shock wave is always a non-equilibrium region, no matter the altitude.

In this chapter, DSMC and NS methods are coupled using the components de-

scribed in Chapter III and the resulting hybrid method is tested for 1D normal shock waves. Detailed investigations of the progression of the hybrid solution, the effect of boundary conditions at hybrid particle-continuum interfaces, and the magnitude of statistical scatter are presented. The hybrid particle-continuum method is used to simulate normal shock waves in argon and diatomic nitrogen gas over a wide range of shock Mach numbers. Hybrid results are compared with pure NS simulations, pure DSMC simulations, and with experimental measurements.

4.1 Experiment and Numerical Parameters

In this study, the MPC method is applied to one-dimensional normal shock waves in argon and diatomic nitrogen gas. Numerical results are compared with an existing data-set of detailed experimental measurements for these flows obtained by Alsmeyer [59]. Measurements were made using an electron beam absorption technique and consist of density profiles over a range of shock Mach numbers between 1.2 and 9.0. The grid employed for *all* MPC, full DSMC, and full NS simulations presented in this chapter has 400 uniform cells in the flow direction and 5 cells in the transverse direction. Thus the mesh refinement algorithm is not used for this study. Although computed on a 2D mesh, the resulting flowfields are strictly 1D and are only presented for the middle (3^{rd}) row of cells. The grid spans approximately 40 upstream mean-free-paths, specified by Alsmeyer as $\lambda = 1.098 \times 10^{-3}\ m$ for argon. When presenting the experimental results, Alsmeyer used this value of λ to normalize the spatial coordinate in the flow direction (x/λ) for both argon and nitrogen flows. The results of this chapter are also presented using the same normalization for both gases. The free-stream conditions are $T_1 = 300\ K$ and the number density $n_1 = 1.6095 \times 10^{21}\ particles/m^3$. This results in free-stream density values

of $\rho_1 = 1.069 \times 10^{-4}\ kg/m^3$ and $\rho_1 = 7.480 \times 10^{-5}\ kg/m^3$ for argon and nitrogen respectively. The velocity is then adjusted to achieve the desired flow Mach number. The simulation particle weight (W), equal to the number of real gas particles represented by each simulated particle, is set to 5.0×10^{19} for all simulations in this study. This results in the number of simulation particles per cell ranging from 30 upstream of the shock to 50-100 downstream of the shock for Mach numbers ranging from 1.55-10 respectively. The DSMC timestep, Δt_{DSMC} is specified such that $u_{\text{free-stream}} \times \Delta t_{DSMC} = \frac{1}{5}\Delta x$, and the NS timesteps are matched to this value. In order to remain consistent with the VHS collision model employed in DSMC regions, the viscosity model used in the NS solver is

$$\mu = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{0.75} \tag{4.1}$$

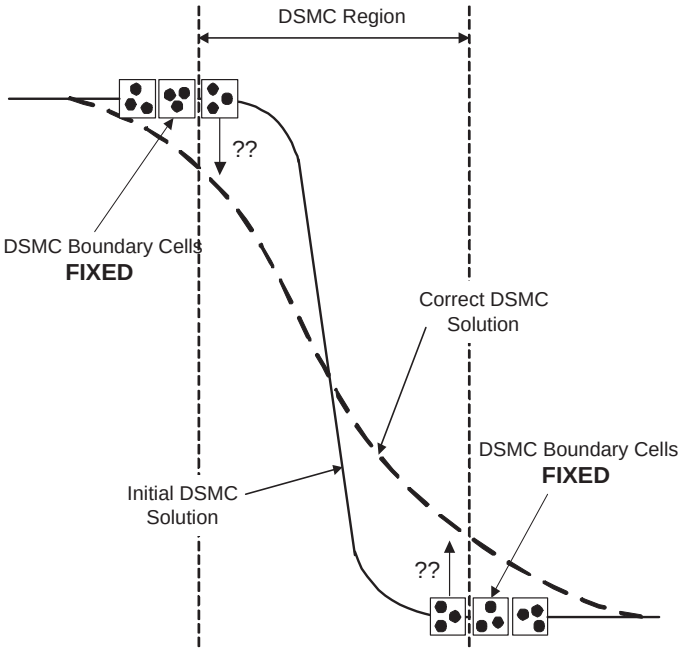
where $\mu_{ref} = 2.13 \times 10^{-5}\ N\ s/m^2 \quad and \quad 1.67 \times 10^{-5}\ N\ s/m^2$ are the reference viscosities of argon and nitrogen respectively, at $T_{ref} = 273\ K$.

4.2 Progression and Relaxation in Particle Regions

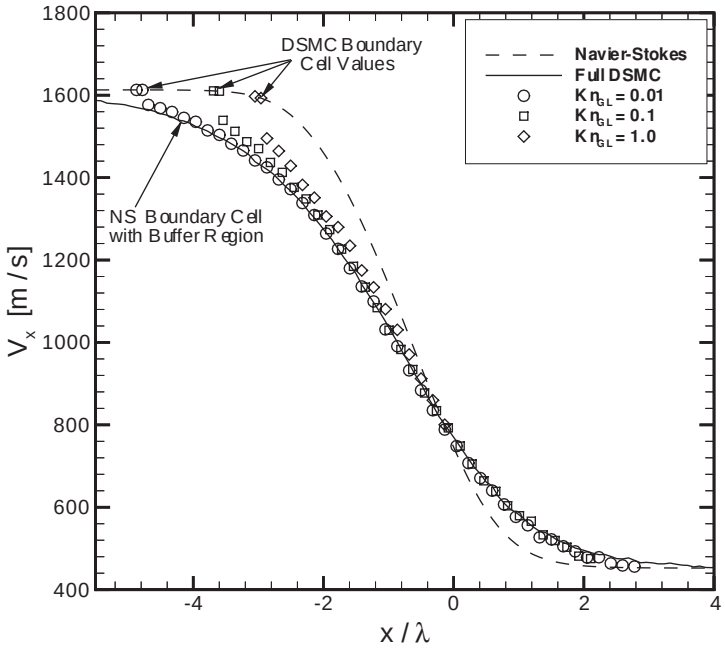
4.2.1 Effect of DSMC boundary cell state

As previously mentioned, one of the important tasks in a hybrid scheme is determining in what regions a particle formulation is necessary. Again, the MPC method begins with a full NS solution to the shock wave and applies the continuum breakdown parameter (Eq. 3.3) to determine the initial particle and continuum domains. Particles are then generated in all cells of the particle domain using the Chapman-Enskog distribution, based on this initial NS solution. It is well known that the NS equations predict a shock that is too thin. As portrayed in Fig. 4.1(a), since there is zero gradient on either side of the thin shock profile, even the application of a very conservative breakdown parameter will result in the initial DSMC domain being too

narrow. Focusing only on the DSMC region in Fig. 4.1(a), the hybrid simulation will begin with an incorrect DSMC solution that is enclosed by boundary cells fixed at a physically incorrect *state*. This raises a fundamental question, namely, how will the DSMC solution respond to such ill-posed boundary conditions? To answer this question, for a Mach 5 shock wave in argon gas, three DSMC domains are determined by applying three values of the continuum breakdown parameter, $Kn_{GL} = 0.01,\ 0.1,$ and 1.0. This results in initial DSMC domains that are too narrow, where hybrid interfaces now lie inside the physically correct shock profile as seen in Fig. 4.1(a). The initial particle region is then simulated using DSMC until the solution no longer varies noticeably (5000 timesteps) and is then sampled to reduce the statistical scatter. During these DSMC iterations, the boundary conditions remain fixed (using the procedures outlined in Chapter III) and correspond to the initial incorrect NS solution. The resulting macroscopic velocity profiles in the narrow DSMC computational domains are shown in Fig. 4.1(b). Here the solid line is the resulting profile of a full DSMC simulation and is taken as the physically correct result. It is clear that although the boundary conditions are fixed on the incorrect NS solution, the entire DSMC solution (including the cells immediately adjacent to the incorrect boundary cells) proceed towards the correct, full DSMC solution. Thus, these ill-posed boundary conditions seem to have little effect on the interior domain. This may be a result of the fact that relatively few simulation particles enter via the boundary cells compared to the vast majority of particles that were initially created to fill the interior of the DSMC domain. This behavior, verified to be independent of mesh density and Mach number, is an important aspect of the hybrid cycle. Clearly, if interior cells were strongly influenced by these ill-posed boundary conditions then this incorrect state might prevent the solution in DSMC regions from progressing



(a) Typical velocity profile showing initial DSMC domain.



(b) Effect of ill-posed DSMC boundary conditions.

Figure 4.1: Imposing incorrect DSMC boundary conditions.

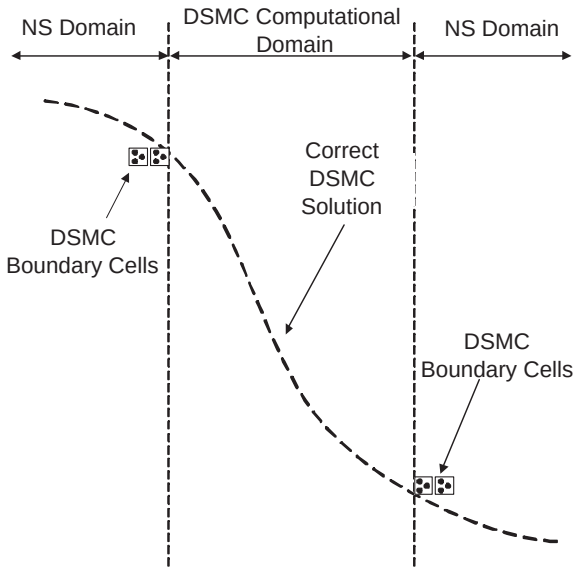


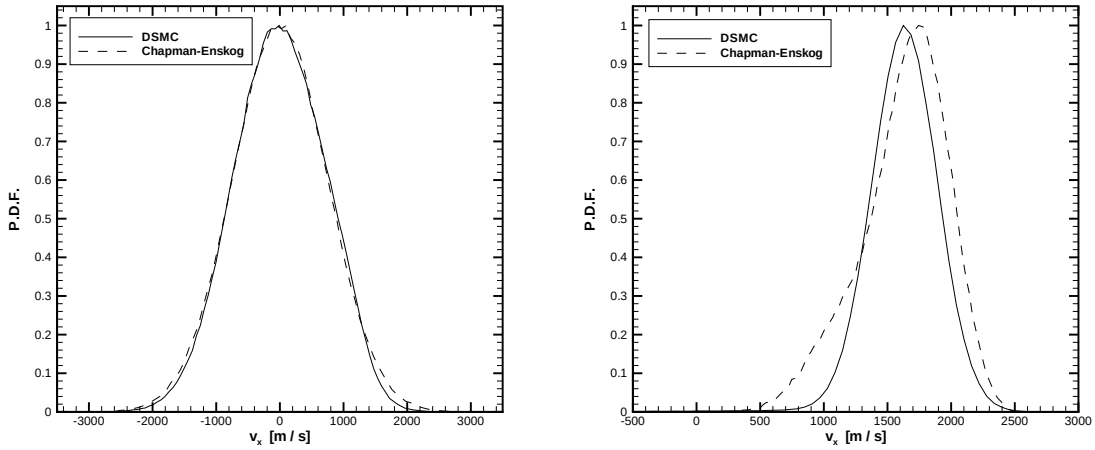
Figure 4.2: Generation of velocity distributions in boundary cells.

in the proper direction. However, it is evident from Fig. 4.1(b) that these ill-posed boundary conditions do affect interior DSMC cells to some degree. One way to reduce this effect would be to generate the NS boundary condition from a DSMC cell located slightly inside the interface as opposed to the cell immediately touching the interface. As shown in Fig. 4.1(b), the information passed back to the continuum NS region would then be in very good agreement with the correct, full DSMC solution. Indeed, one important reason why the MPC method uses an overlap region (refer to Fig. 3.5) is that such a "buffer" prevents local effects of ill-posed DSMC boundary conditions from being transferred back into the continuum domain.

4.2.2 Effect of DSMC boundary cell velocity distribution

Ultimately, a DSMC simulation requires boundary cells not only representing the correct macroscopic state, but whose particles represent the physically correct velocity distribution at that location in the flowfield. Most DSMC simulations apply boundary conditions in equilibrium (free-stream) flow regions where a Maxwellian

distribution is physically correct. A hybrid method attempts to place boundary conditions in regions of near-equilibrium flow and therefore requires a Chapman-Enskog velocity distribution. However, if the DSMC boundary cells are located in regions of *non*-equilibrium, even if the correct macroscopic state is specified, a Chapman-Enskog distribution may not be the correct velocity distribution. This situation is portrayed in Fig. 4.2. Here, the physically correct state is prescribed in the DSMC boundary cells and both the generated and true velocity distributions are investigated. Specifically, the investigation is performed for the $Kn_{GL} = 0.1$ case with the pre and post-shock boundary locations seen in Fig. 4.1(b). The true velocity distributions in both post and pre-shock boundary cells, determined from a full DSMC simulation of the shock wave, are plotted in Figs. 4.3(a) and 4.3(b) respectively. These are compared with Chapman-Enskog velocity distributions generated by within the boundary cells. It is important to note that for this comparison, the correct macroscopic state and the correct local gradients (taken from the full DSMC result) are used to compute the Chapman-Enskog distribution (Eq. 3.5). The generated distribution in the post-shock boundary cell (Fig. 4.3(a)) is seen to agree well with the true distribution, signifying that this boundary cell lies within a near-equilibrium region. However, the generated Chapman-Enskog distribution in the pre-shock boundary cell (Fig. 4.3(b)) is seen to differ significantly from the true distribution computed using DSMC at this location. This implies that the pre-shock boundary is located in a non-equilibrium region where the velocity distribution cannot be described by a Chapman-Enskog distribution. Thus, the results of this section show that until the interfaces are correctly located in regions of near-equilibrium, the DSMC boundary conditions may contain both an un-physical state and velocity distribution. Fortunately, these ill-posed boundary conditions were shown not to have



(a) Post-shock boundary cell.

(b) Pre-shock boundary cell.

Figure 4.3: Velocity distributions in DSMC boundary cells.

a large influence on the interior DSMC solution, which is still able to progress significantly towards the correct non-equilibrium result.

4.3 A tightly coupled hybrid numerical cycle

Now that it has been determined that despite initially ill-posed boundary conditions, the DSMC region evolves towards the correct solution in time, this information must be transferred into the continuum domain. For example, the variation in macroscopic velocity calculated in the DSMC region for a Mach 5 shock wave with boundaries set using $Kn_{GL} = 0.01$ was shown in Fig. 4.1(b). As suggested at the end of the previous section, a small overlap region is created such that the NS boundary cell is placed 5 cells inside the interface. Figure 4.1(b) shows that in the pre-shock NS boundary cell, the velocity decreases from 1614 m/s to 1550 m/s (roughly 4%) during approximately 5000 DSMC timesteps. It is desirable to represent this changing DSMC information as a time varying NS boundary condition. The degree of statistical scatter and the success of the sub-relaxation average for calculating the

macroscopic variation in this particular cell was previously shown in Figs. 3.6 and 3.7 of Chapter III. Although not shown, the statistical variations in density, velocity, and temperature in the post-shock NS boundary cell are roughly 30%, 50%, and 20% respectively. For each flow variable in both the pre and post-shock NS boundary cells, the sub-relaxation average is able to provide low scatter, semi-time-accurate macroscopic variations.

In summary, the NS boundary condition is able to follow the changing DSMC solution with manageable lag and low scatter. The DSMC boundary condition was shown to be valid only in regions of near-equilibrium flow and thus it is desirable to locate the proper interface location quickly. Note however, results also concluded that incorrect DSMC boundary conditions do not prevent the DSMC solution from progressing towards the correct result. This section will implement all of the above results and recommendations into a strongly-coupled hybrid cycle that transfers information between particle and continuum regions at each timestep. The proposed hybrid cycle involves the following steps:

1. Obtain an initial solution to the NS equations.

2. Generate DSMC and NS domains by applying the Kn_{GL} continuum breakdown parameter to the current solution. Generate new simulation particles as needed using the Chapman-Enskog velocity distribution.

3. Create particles in the DSMC boundary cells from the current NS solution using the Chapman-Enskog velocity distribution.

4. Advance the DSMC domain by one timestep, Δt_{DSMC}.

5. Set the NS boundary conditions using the sub-relaxation technique applied to

the updated DSMC particles.

6. Advance the NS domain in time by Δt_{DSMC}.

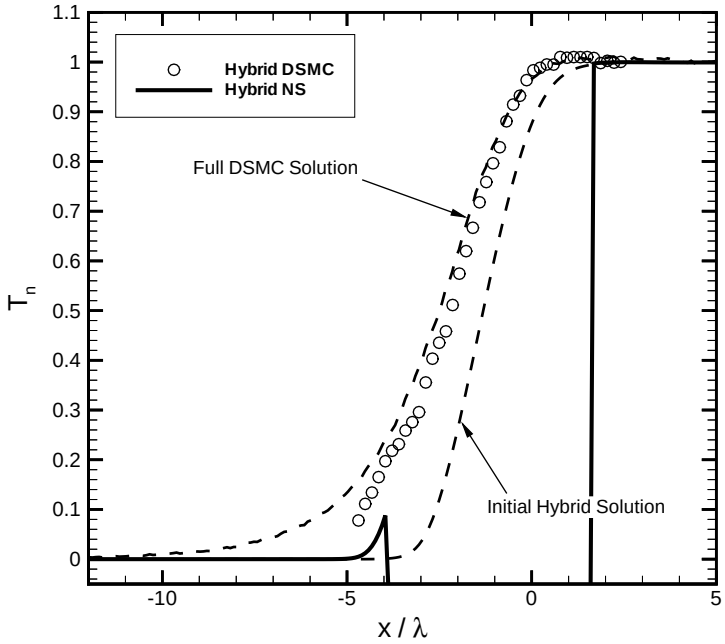
7. Return to step 3. -OR- Every N_{Br} timesteps, return to step 2.

8. If step 2. no longer modifies the DSMC/NS domains, then lock the domains. Begin cumulative DSMC sampling and now generate the NS boundary condition based on this cumulative average. Repeat steps 3. to 6. until the degree of scatter in the DSMC solution, and the NS residual, fall below threshold values.

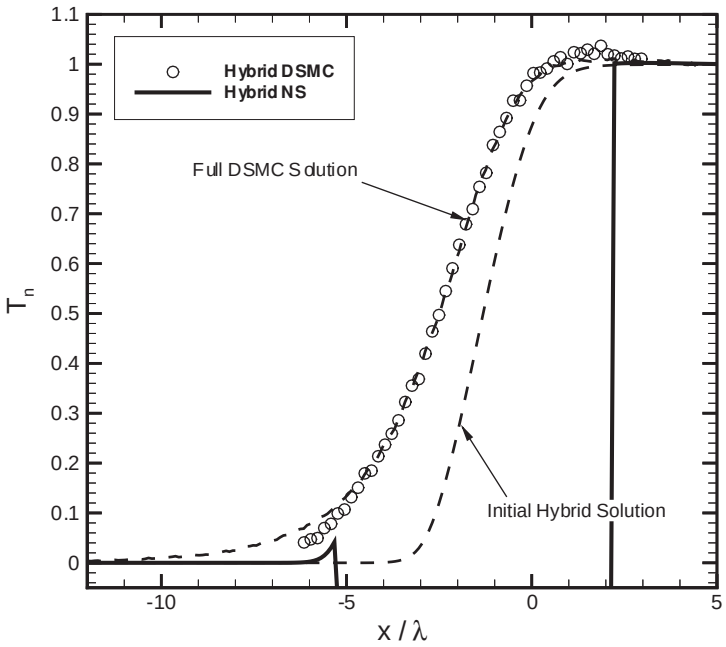
Figure 4.4 exhibits how this hybrid cycle proceeds to solve a Mach 6.5 shock wave. The recommended [50, 51] value of $Kn_{GL} = 0.05$ is used and $n_{\text{overlap}} = 5$ cells are used for the overlap region. In order to clearly visualize the progression of the hybrid solution, the breakdown parameter is only applied every $N_{Br} = 2000$ timesteps. All variables are normalized as,

$$\rho_n = \frac{\rho - \rho_1}{\rho_2 - \rho_1}, \qquad Vx_n = \frac{Vx - Vx_2}{Vx_1 - Vx_2}, \qquad T_n = \frac{T - T_1}{T_2 - T_1} \tag{4.2}$$

with the spatial dimension (x) normalized by the mean free path of the gas upstream of the shock. Temperature profiles are presented, as they involve the largest variations in both the DSMC and NS solutions, especially upstream of the shock wave. Initially, the entire hybrid solution is a NS domain set to a NS solution of the shock wave problem. Immediately, the breakdown parameter is applied resulting in the initial DSMC and NS domains depicted by vertical lines in Fig. 4.4(a). Here, the hybrid solver has already cycled for 2000 timesteps and as a result, the hybrid DSMC solution has moved significantly towards the full DSMC solution. Recall, this important behavior was observed previously in Fig. 4.1(b). Through information transfer using the sub-relaxation technique, the hybrid NS solution is seen to have followed



(a) After 2000 timesteps.



(b) After 4000 timesteps.

Figure 4.4: Progression of a hybrid simulation for a Mach 6.5 shock.

this changing DSMC solution smoothly with some lag. The breakdown parameter is then applied to the current hybrid solution seen in Fig. 4.4(a). Since the gradient in the NS region has now been extended further in each direction, the new DSMC domain (shown in Fig. 4.4(b)) is now larger. Here, the hybrid solver has cycled for another 2000 timesteps with these new interface locations and the DSMC solution has progressed even further towards the correct solution. The DSMC domain continues to expand every 2000 timesteps, until at $22,000$ timesteps application of the breakdown parameter no longer results in a change in the DSMC or NS domain size. This signifies that the DSMC solution is no longer changing in time and has thus reached steady state. At this point the domain sizes are locked and cumulative DSMC sampling begins in order to reduce the scatter in the hybrid DSMC solution to an arbitrarily low value. Since the DSMC solution is now in steady-state, the sub-relaxation averaging is abandoned and the NS boundary condition is set using the cumulative DSMC average. The hybrid cycle is run for 3000 more timesteps with the finalized interfaces in order to reduce the scatter in the final solution which is shown in Fig. 4.5. Clearly the hybrid solution reproduces exactly the full DSMC solution. Notice that the DSMC portion of the solution transitions very smoothly to the NS portion and that the two solutions agree very well in the overlap region. This signifies that the choice of $Kn_{GL} = 0.05$ does in fact place the interface in a region of near-equilibrium that can be modeled successfully with the NS equations.

The actual temperature boundary condition in the pre-shock NS boundary cell is plotted in Fig. 4.6. Although some of the variations are quite large, they occur over a large number of timesteps. Since the NS boundary condition is updated at every DSMC timestep (strongly-coupled), these large variations present no stability problems for the NS solver. It should be noted that the discontinuous variations

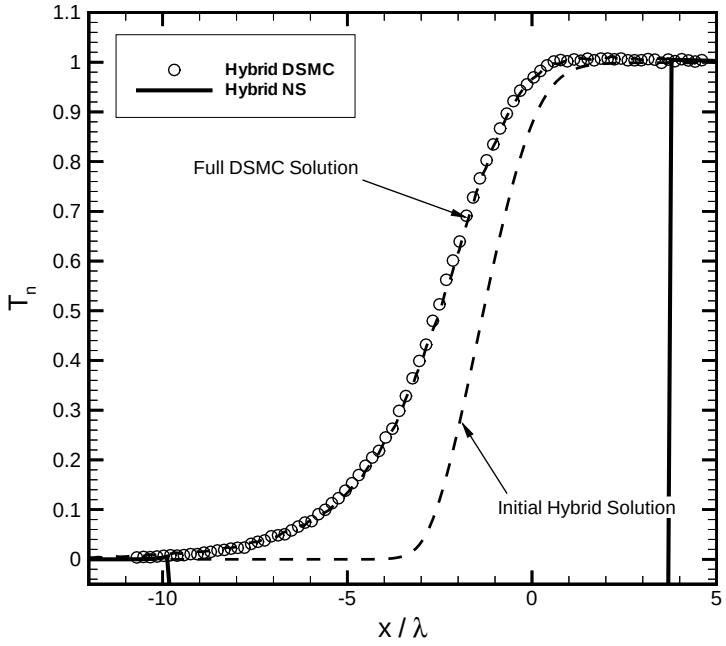


Figure 4.5: Final hybrid solution.

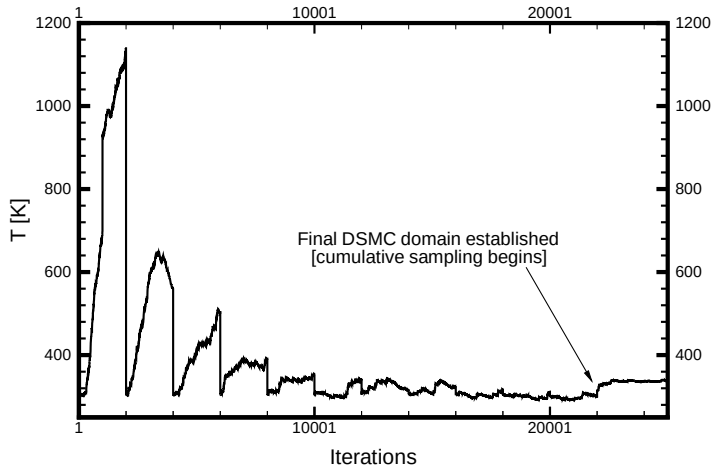


Figure 4.6: Temperature variation in pre-shock NS boundary cells.

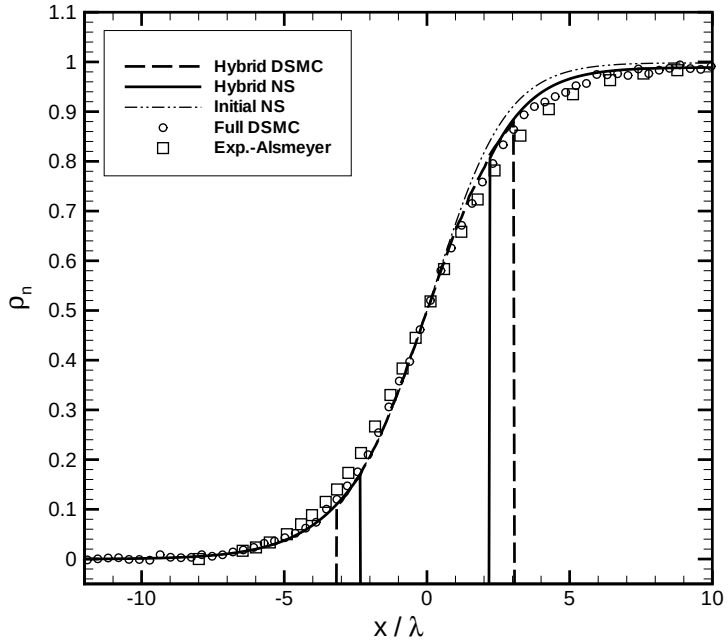
(every 2000 timesteps) do not occur in the *same* NS cell. Recall that every 2000 timesteps, the interfaces and thus the NS boundary cells are relocated. Figure 4.6 clearly shows that there is still scatter present in the NS boundary condition, but its magnitude is far less than the scatter associated with the simple spatial average presented in Chapter III. Finally, the scatter is virtually eliminated by switching to cumulative DSMC sampling after the final domains are established (in this case after $22,000$ timesteps).

4.4 Comparison with Experimental Data

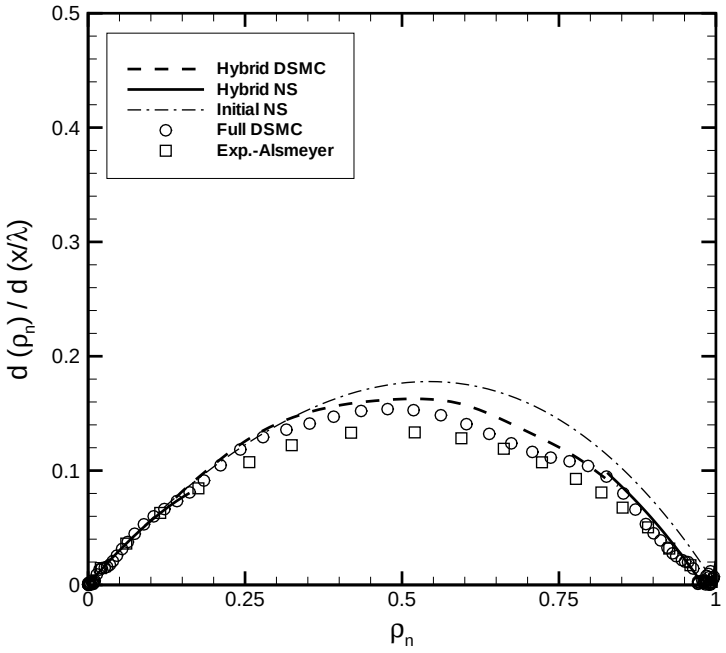
In the previous section the number of timesteps between application of the breakdown parameter was set at a large value, $N_{Br} = 2000$, in order to *clearly* demonstrate the stages involved in the hybrid cycle. However, one of the previous conclusions drawn is that the final interface locations should be established as quickly as possible. This may be achieved by applying the breakdown parameter more often. However, N_{Br} must remain large enough to allow the DSMC solution to change before the breakdown parameter is applied again, otherwise this would waste computational time. For all cases presented in this section, an optimal value of $N_{Br} = 500$ timesteps was determined to find the correct interfaces in the least amount of time. In fact, on average, the final interfaces are located in 4000 timesteps. It is very important to note that this is approximately the same number of timesteps (only slightly more) that a full DSMC simulation requires to reach steady state. Thus, for these cases, the correct interfaces are effectively determined *while* the DSMC portion is proceeding towards steady state. The experimental flow conditions and numerical parameters for all simulations have been outlined previously in Section 4.1. The numerical cycle used for all hybrid simulations is that just described in Section 4.3 with the exception

that N_{Br} is set to 500. The hybrid code is then able to produce all of the results in this section by simply changing the gas properties and free-stream velocity.

All hybrid results will be compared with full DSMC simulations, full NS solutions, and with experimental data from Alsmeyer [59]. Figures 4.7, 4.8, and 4.9 show the hybrid results in argon gas for Mach numbers of 1.55, 3.38, and 9.0, respectively. Figure 4.7(a) compares the density profiles computed using full DSMC, NS, and the tightly-coupled hybrid method with experimental data for a Mach number of 1.55. For the hybrid solutions presented in this chapter, the entire hybrid-DSMC and hybrid-NS regions are plotted. The agreement between particle and continuum solutions within the overlap regions is thus evident. Although all profiles are in close agreement at this low Mach number, the NS solver results in too-thin a shock wave, while the full DSMC solution is in slightly better agreement with the experimental result. Notice that the final, steady-state hybrid interfaces are located well within the density gradient for this case. Although, the hybrid profile does not exactly reproduce full DSMC results, it is certainly in better agreement than the initial NS profile in the downstream portion of the shock. Figure 4.7(b) depicts profiles of the normalized density gradient for each simulation method. The gradient at $\rho_n = 0.5$ is also referred to as the *reciprocal shock thickness*. Note that Fig. 4.7(b) is entirely consistent with Fig. 4.7(a) as the density gradients computed by all methods agree exactly in the fore-shock region and begin to differ near the center of the shock wave. The density profiles for a stronger, Mach 3.38 shock wave are shown in Fig. 4.8(a). Here the final particle region is much larger than for the low Mach 1.55 case. This signifies a larger degree of non-equilibrium, which is confirmed by increased error in the full NS solution. For this case, both Figs. 4.8(a) and 4.8(b) verify that the hybrid solution reproduces the full DSMC solution with a high level of accuracy. Additionally, the

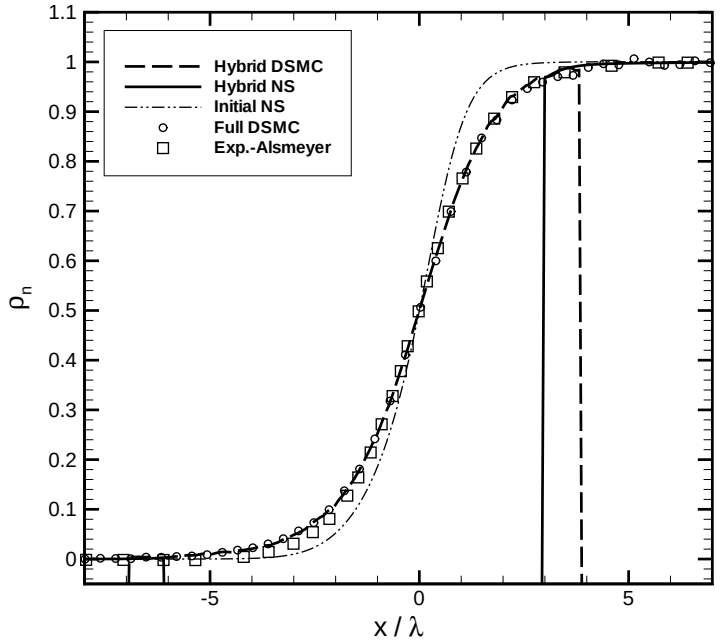


(a) Density profile.

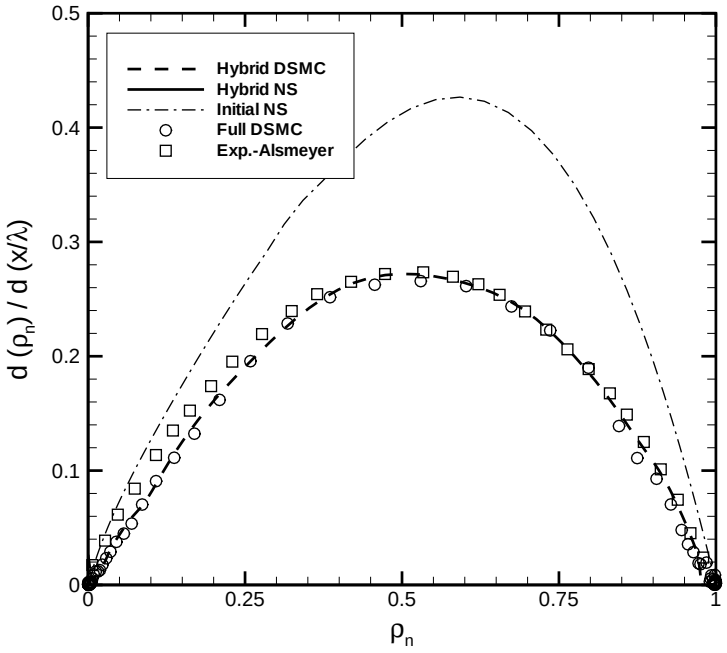


(b) Reciprocal shock thickness.

Figure 4.7: Results for a Mach 1.55 argon shock wave.

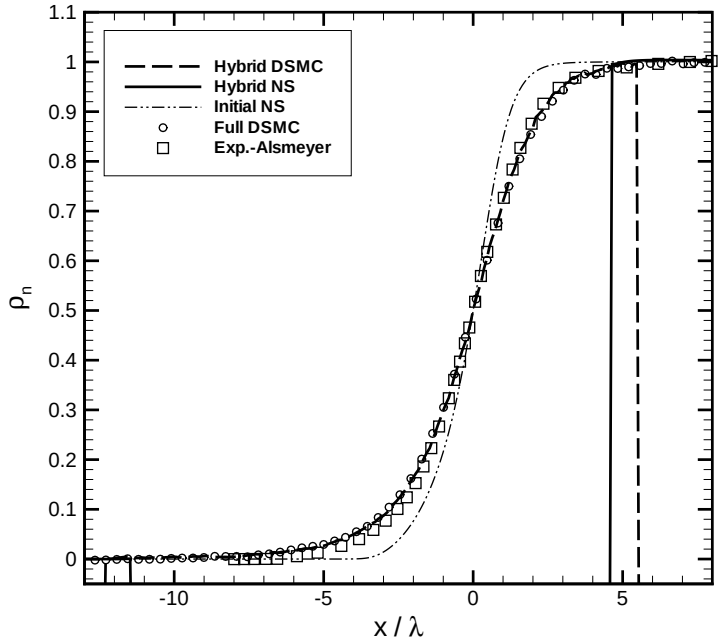


(a) Density profile.

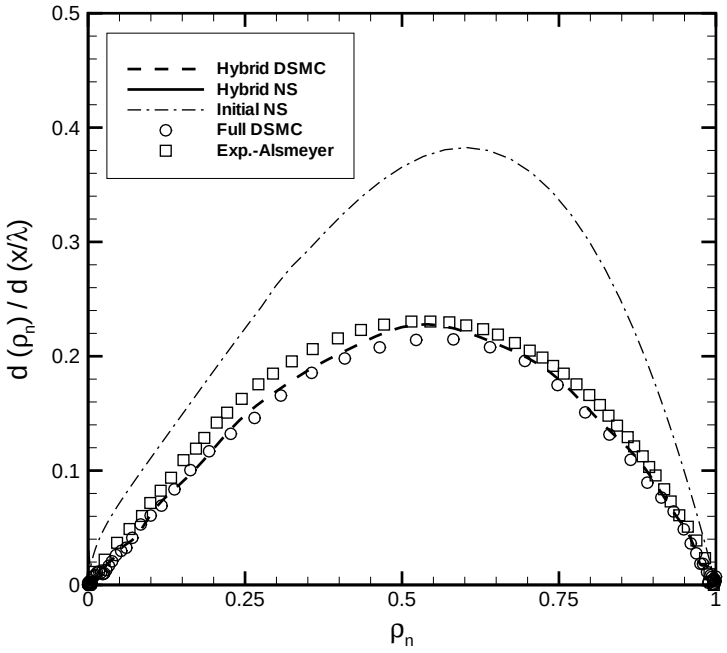


(b) Reciprocal shock thickness.

Figure 4.8: Results for a Mach 3.38 argon shock wave.



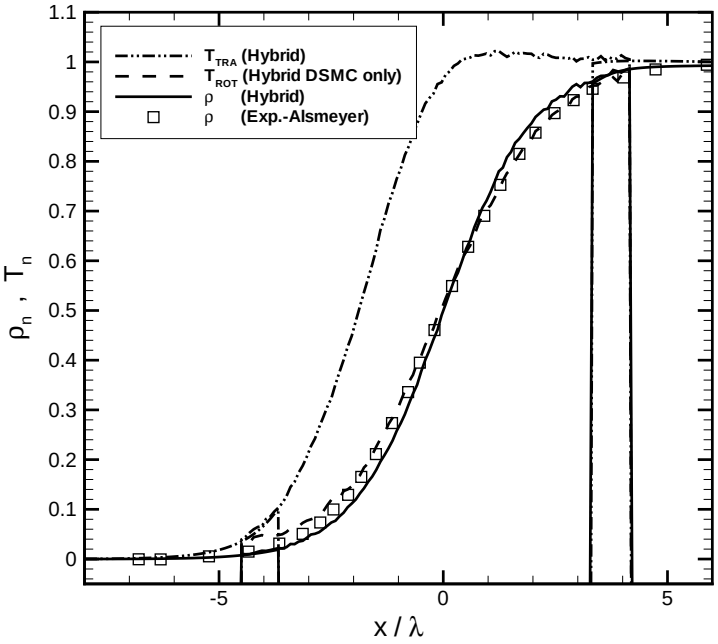
(a) Density profile.



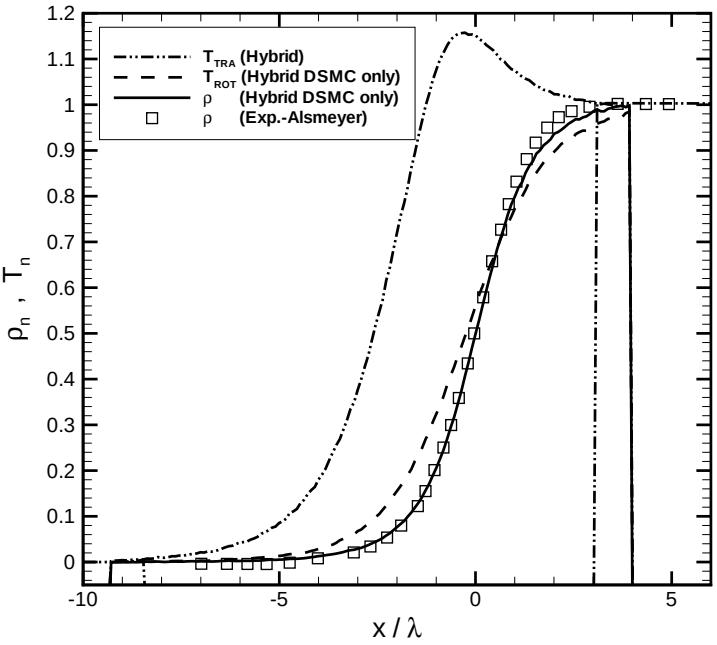
(b) Reciprocal shock thickness.

Figure 4.9: Results for a Mach 9 argon shock wave.

full DSMC solution is seen to accurately predict the experimental results. Similarly, the results for the Mach 9.0 case, depicted in Figs. 4.9(a) and 4.9(b), show the hybrid solution accurately reproduces the full DSMC solution, which again agrees very well with experimental data. For this high Mach number case, the final DSMC domain is very large. In fact, the final interfaces extend beyond any significant density gradient (especially upstream of the shock). The reason for this is that the temperature gradient precedes the density gradient in high-speed shock waves. Although Alsmeyer did not measure temperature, this phenomenon was seen earlier in Fig.4.5. Since the breakdown parameter detects the maximum of density, velocity, and temperature gradients (see Eq. 3.3), the temperature gradient seen in Fig. 4.5 will push the interface upstream. Similar results are presented for simulations of diatomic nitrogen at shock Mach numbers of 2.0 and 6.1 in Figs. 4.10(a) and 4.10(b) respectively. For the nitrogen cases, full DSMC and full NS results are not plotted, however for Mach numbers ranging from 1.5 to 10.0, the hybrid results are verified to match full DSMC solutions and result again in thicker shock waves than those obtained by full NS solutions. Experimental results exist only for the density profiles and the hybrid density results (solid lines) shown in Figs. 4.10(a) and 4.10(b) agree well with these experimental measurements. Since nitrogen is a diatomic molecule, the DSMC region now contains both a translational temperature (T_{TRA}) and a rotational temperature (T_{ROT}). No vibrational degrees of freedom are included in the DSMC simulations. It is interesting to note that at the hybrid particle-continuum interfaces the gas is close to thermal equilibrium; that is, there is only a small difference between T_{TRA} and T_{ROT}. Thus although the Kn_{GL} parameter does not directly account for thermal non-equilibrium, it positions the interfaces in regions quite close to thermal equilibrium. However, since the gas is not exactly in



(a) Density and temperature profiles (Mach 2.0).



(b) Density and temperature profiles (Mach 6.1).

Figure 4.10: Results for nitrogen shock waves of various Mach number.

thermal equilibrium, future simulations may need to reduce the continuum cutoff parameter below $Kn_{GL} = 0.05$ or possibly incorporate thermal non-equilibrium into the definition of continuum breakdown. This issue will be analyzed in Chapter V.

Full NS, DSMC, and hybrid simulations were run for 13 Mach numbers between 1.2 and 10. The reciprocal shock thickness predicted by these methods for argon gas was calculated and compiled into one graph for comparison with experimental data over the entire Mach number range. This result is shown in Fig. 4.11. Clearly, the NS solver predicts a shock wave that is far too thin, while the DSMC method, using the viscosity law in Eq. 4.1, agrees with the experimental results very well. In addition, the hybrid solver is seen to reproduce the DSMC results very accurately for each Mach number. Finally, it should be noted how all three solvers predict the same reciprocal shock thickness when the Mach number is small. In fact, for a Mach 1.244 case, application of the continuum breakdown parameter on the initial NS solution results in no DSMC domain at all. Thus, the hybrid solver immediately returns the full NS solution as the final solution. The NS solution for this case does indeed agree both with experiment and a full DSMC simulation and thus lends further support for using $Kn_{GL} = 0.05$ as the cutoff for continuum breakdown.

4.5 Computational Efficiency

In order to maintain consistency for computational efficiency comparisons, a full DSMC simulation *begins* with particles initialized by a Chapman-Enskog distribution based on a NS solution, similar to the initial particle regions in a hybrid simulation. The full DSMC simulation is considered to have reached steady-state when the total number of simulation particles no longer varies appreciably. In contrast, a hybrid simulation is considered to have reached steady state only after the interfaces have

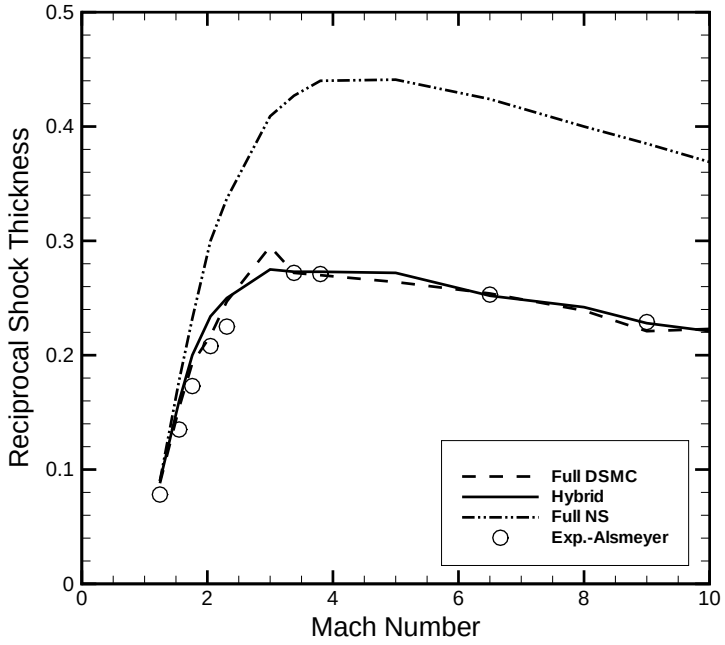


Figure 4.11: Reciprocal shock thickness over entire Mach number range.

adapted to their final positions *and* the total number of particles has reached steady state (the beginning of step 8 in the hybrid numerical cycle). In order to obtain the same level of statistical scatter in the final solutions, both full DSMC and hybrid simulations are then sampled for 3000 timesteps after reaching steady state. For the hybrid simulation, this means that steps 3 through 6 (from Section 4.3) are repeated 3000 times.

Computational profiling of the hybrid code reveals an average speed-up factor of 2 compared with a full DSMC simulation for 1D shock waves in argon. Clearly, both the size of the overall computational domain as well as the average number of particles per DSMC cell will greatly influence the efficiency of the hybrid code and are indeed chosen somewhat arbitrarily for this study. The efficiency comparisons are thus only included for completeness and cannot be readily extended to other physical problems or multi-dimensional simulations. Having said this, Fig. 4.12 displays the

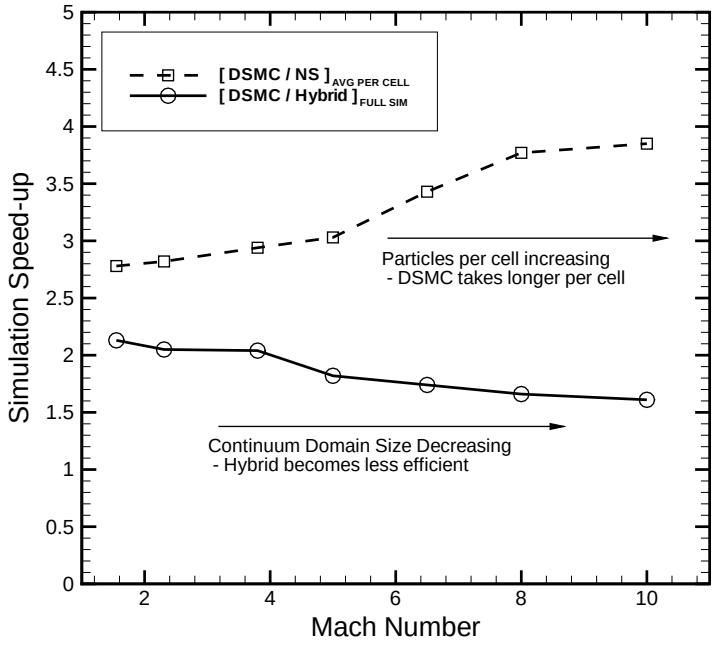


Figure 4.12: Hybrid efficiency for 1D normal shock waves.

increase in efficiency gained by the hybrid code for this study. As the Mach number increases, the density in the post-shock region increases as well. Since, on average, there are now more particles per DSMC cell, the ratio of time spent by the DSMC method for one cell to that spent by the NS solver increases with Mach number. For this reason, the hybrid solver would be expected to increase in efficiency as Mach number increased. However, at the same time, the hybrid particle domain becomes larger as Mach number increases. This reduces the efficiency of the hybrid solver as fewer cells may be simulated as continuum cells. The overall result is that the efficiency degrades slightly as Mach number increases and a full DSMC simulation remains roughly 2 times slower than a hybrid simulation. According to the profiling, the DSMC method takes only 3-4 times longer per cell than the NS solver. Thus, even if the entire domain could be represented as continuum cells, the maximum speed-up would be a factor of 3-4 times. It is very important to realize, however, that the

increase in efficiency gained by a hybrid solver ultimately comes from simulating far fewer total cells for fewer timesteps. In this study, for simplicity, the continuum mesh maintained the same density as the particle mesh and the same timestep was used by both methods. The true advantage of a hybrid code, as mentioned in Chapter III, is that the continuum mesh is able to be far coarser than the particle mesh. In multiple dimensions such a large decrease in the number of cells simulated will result in much larger computational savings. Additionally, although this 1D study strongly couples an explicit NS solver to the DSMC method, simulations presented in Chapters V and VI will investigate the benefits of using an implicit NS solver. This will enable significant time-scale decoupling and therefore allow the continuum cells to be updated far less frequently than the particle cells. The combination of simulating fewer cells for fewer timesteps in large continuum portions of the flowfield is expected to greatly increase the computational efficiency of the hybrid scheme.

4.6 Conclusions Drawn from Normal Shock Wave Study

Three important conclusions are drawn from application of the MPC method to 1D normal shock wave problems.

1) If particles in DSMC regions are initialized to be consistent with a continuum NS solution, as DSMC iterations are performed the DSMC solution progresses towards the correct non-equilibrium result. This progression occurs in a stable and robust manner despite particles generated within DSMC boundary cells representing a slightly incorrect state and having an incorrect velocity distribution. This behavior of the DSMC method is vital to the success of a hybrid particle-continuum method. Furthermore, the results shown in Fig. 4.1(b) demonstrate that particle regions are capable of progressing and relaxing significantly towards the correct non-equilibrium

result with no coupling to continuum regions at all. This suggests that tight coupling (transferring information at each timestep) may not be necessary for the hybrid simulation of steady-state flows and a loosely coupled approach may be sufficient. A loosely coupled approach has many advantages and is fully investigated in the next chapter.

2) For high-speed steady-state flows, a sub-relaxation averaging technique (spatial and temporal averaging) used in conjunction with state-based coupling is able to provide low scatter, semi-time-accurate macroscopic properties within particle regions. These hydrodynamic variations within particle regions are then available for use as boundary conditions for continuum regions. In addition, the level of scatter is low enough that the continuum breakdown parameter (Kn_{GL}) can be applied within DSMC regions. In this way, the MPC method succeeds in adapting the particle-continuum interface locations as particle regions progress towards the correct steady-state non-equilibrium solution.

3) A tightly-coupled hybrid particle-continuum algorithm based on these findings was able to transition from an initial continuum solution and ultimately reproduce experimental and full DSMC simulation results in half the time for 1D argon and nitrogen shock waves over a large Mach number range.

CHAPTER V

Hypersonic Blunt-Body Simulations

The modular particle-continuum (MPC) method is now extended for two-dimensional and axi-symmetric simulations by closely examining its performance for blunt-body flows. Blunt-body flows are possibly the best application for a hybrid particle-continuum method due to the extreme variation in spatial and temporal collisional scales between stagnation and wake regions and also due to the presence of a very strong, highly non-equilibrium bow shock. In the one-dimensional normal shock simulations, continuum regions were simply constant regions at the pre and post-shock states. For 2D and axi-symmetric blunt-body simulations, continuum regions are much more complicated. The boundaries are not single cells but may form a complicated boundary through complex regions of the flow.

In this chapter, the MPC numerical cycle presented in Chapter IV is extended for 2D and axi-symmetric flows and is completely detailed. The hybrid numerical cycle is carefully constructed in order to ensure that hybrid solutions progress towards the correct physical result as opposed to diverging towards alternate incorrect solutions. This chapter then applies the finalized hybrid particle-continuum method to baseline cases of hypersonic flow over 2D cylinder geometries and an axi-symmetric planetary probe geometry. Hybrid results for flowfield and surface properties as well as for

local velocity distribution functions are compared with full DSMC solutions, full NS solutions, and with experimental data where available. Finally, the computational efficiency gained by the hybrid method over full DSMC simulation is investigated.

5.1 Particle and Continuum Simulation of Cylinder Flows

5.1.1 Flow Conditions

Flow conditions are chosen to lie in the transitional regime containing regions of both continuum flow and regions which exhibit significant non-equilibrium effects. Such effects include a strong bow shock, thermal non-equilibrium, velocity slip and temperature jump on surfaces, and rarefied, recirculating flow in the wake. Specifically, two-dimensional, hypersonic flow of N_2 gas over an 8 cm diameter cylinder at an altitude of approximately 70 km is investigated. Using the U.S. standard atmosphere, this altitude corresponds to a free-stream density, $\rho = 7.48 \times 10^{-5}\ kg/m^3$, and temperature, $T = 217.45\ K$. Assuming the gas is comprised only of N_2, these conditions correspond to a number density, $n = 1.61 \times 10^{21}\ 1/m^3$, pressure, $p = 4.83\ Pa$, and result in a global Knudsen number (based on diameter) of 0.01. In order to investigate varying degrees of non-equilibrium flow, the free-stream velocity is set to 902, 1804, and 3608 m/s resulting in Mach 3, 6, and 12 flow, denoted as cases **M3**, **M6**, and **M12**. For each simulation, the cylinder wall temperature is assumed constant at 300 K, 500 K, and 1000 K for cases **M3**, **M6**, and **M12**, respectively. The DSMC module assumes diffuse reflection and full thermal accommodation at the cylinder wall. The NS solver employs no-slip conditions for both velocity and temperature.

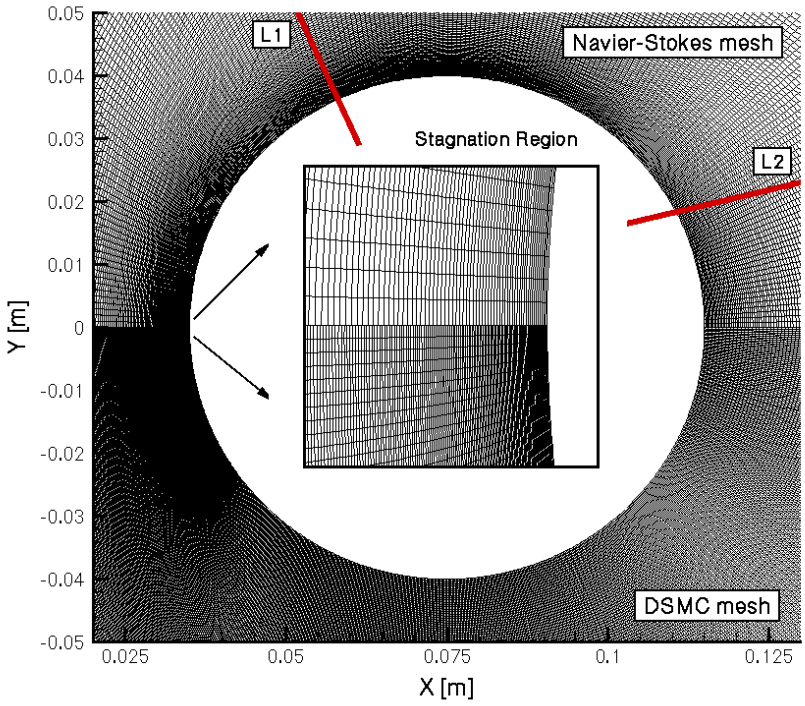


Figure 5.1: NS and DSMC grids.

5.1.2 Spatial and Temporal Scale Resolution

The baseline DSMC mesh is composed of quadrilateral cells and consists of 600 evenly spaced cells along the surface of the cylinder and 300 cells in the radial direction (180,000 total cells). In the fore-body, the cell size is clustered towards the cylinder surface. For all cases the cell size is verified to be less than the local mean-free-path throughout the domain which is a requirement of the DSMC method. A portion of this mesh is shown in the bottom of Fig. 5.1. For DSMC simulations, the reference particle weight is set to obtain at least 20 particles per cell in the wake region which results in approximately 50 particles per cell, on average, in the fore-body flow. No particle weighting, nearest-neighbor collision routine, or sub-cell sampling is used. Constant DSMC timesteps of 5×10^{-8} s are used for both the **M3** and **M6** cases, while a timestep of 2×10^{-8} s is used for case **M12**. These timesteps

are verified to be less than one-half of the shortest mean-free-time anywhere in each simulation. For MPC simulations where no mesh refinement is used, the initial NS solution and the MPC solution are both obtained using the baseline DSMC mesh described above. Thus all three simulation methods use the same mesh (shown in the bottom of Fig. 5.1) and each continuum cell in an MPC simulation corresponds to only a single DSMC cell. The constraint on cell size required for the baseline DSMC mesh ($\Delta x < \lambda$) is more than sufficient for an accurate and grid-converged NS simulation.

For MPC simulations utilizing the mesh refinement algorithm, the initial NS solution is obtained on a significantly coarser mesh containing only 30,000 quadrilateral cells. This mesh is shown in the top of Fig. 5.1 and contains 300 equally spaced cells around the surface of the cylinder and 100 cells in the radial direction. In the forebody, cells are clustered towards the surface in order to ensure a sufficient number of cells in the boundary layer. The NS solution obtained on this mesh is verified to reproduce the NS solution obtained on the much finer baseline DSMC mesh. Thus, this level of mesh refinement (30,000 cells) is sufficient to obtain a grid-converged solution to the NS equations. In fact, other researchers have used very similar mesh resolution for NS simulation of problems with similar geometry and flow conditions [60]. MPC simulations using the mesh refinement algorithm thus begin with a NS solution obtained on the 30,000 cell continuum mesh. In DSMC regions, the mesh is then refined to the local mean-free-path using Eq. 3.4, resulting in a resolution similar (but not identical) to the baseline DSMC mesh. Specifically, the initial mesh refinement setup by the MPC method for the Mach 6 flow was shown earlier in Fig. 3.4 of Chapter III. MPC simulations utilizing the mesh refinement algorithm for Mach 3, 6, and 12 flows are denoted as cases $\mathbf{M3_R}$, $\mathbf{M6_R}$, and $\mathbf{M12_R}$. These MPC

simulations obtained on hybrid meshes (refer to Fig. 3.4) will thus be compared with NS solutions on the coarse continuum mesh (top of Fig. 5.1) and compared with full DSMC simulations obtained on the baseline DSMC mesh (bottom of Fig. 5.1).

Clearly, compared with the continuum mesh, the refinement required by the DSMC method is much more significant and therefore more computationally demanding. The large variation in mean-free-path for typical blunt-body flows, shown earlier in Fig. 3.2 for the Mach 6 flow, presents a challenge for full DSMC simulation. If λ varies by an order of magnitude between stagnation and wake regions, then for the area simulated by one cell in the wake, 100 cells (1000 cells in 3D) are required to simulate the same area (or volume) in the fore-body. Since each DSMC cell must contain a minimum of 10-20 particles for statistical reasons, this directly corresponds to an enormous number of simulation particles required in the fore-body region. In addition, the mean-free-time may also be an order of magnitude smaller in the fore-body region than in the wake. DSMC simulations are therefore limited to very small timesteps when simulating stagnation regions of blunt-body flows. In contrast, NS simulation does not have such restrictions. As seen in the top of Fig. 5.1, the fore-body region is simulated using larger continuum cells. Due to the CFL condition inherent in all CFD solution techniques, larger cells allow for larger simulation timesteps. More importantly, the NS equations can be solved using an implicit time-marching scheme allowing for even larger simulation timesteps. Thus a converged steady-state solution to the NS equations can be obtained very efficiently. For example, initial NS solutions and NS updates within the MPC simulations for cases $\mathbf{M3_R}$, $\mathbf{M6_R}$, and $\mathbf{M12_R}$ use a CFL number of 10. This results in the timesteps used to simulate continuum regions being 20-25 times larger than the timesteps used to simulate particle regions. The efficiency of the MPC method over pure DSMC

thus comes from the combination of simulating fewer cells for fewer timesteps in continuum regions where the NS equations provide an accurate model for the flow.

5.1.3 Continuum Breakdown and Accuracy of DSMC and NS Methods

A review of numerical simulations using both DSMC and Navier-Stokes methods and their agreement with various hypersonic, blunt-body experiments is presented in Ref. [2]. One conclusion of this review article is that while the NS results generally match DSMC and experimental data in the fore-body flow, they deviate significantly from DSMC and experiment in the near-wake for both surface and flow features. In order to obtain more accurate results in the near-wake region without resorting to a full DSMC simulation, "zonally decoupled" DSMC-NS simulations [61] have been performed. Here, the fore-body flow is solved separately using a NS solver and the exit-plane solution is specified as the inflow condition to a decoupled DSMC simulation of the entire wake region. Results of the zonally decoupled simulations agree well with full DSMC solutions, while requiring only half of the computational resources (CPU time and memory) [61]. Comparison between zonally decoupled, full DSMC, and full NS simulations also emphasizes the importance of modelling rarefied effects such as thermal non-equilibrium and slip flow in order to predict surface properties and wake closure. Using a similar decoupled approach, the flow about the Mars Sample Return Orbiter was simulated at both flight conditions [62] and for a wind tunnel experiment [63]. The interface between NS and DSMC regions was taken as a plane cutting through the aeroshell lip. Although this practical approach was much more efficient than a full DSMC simulation, in one simulation, 20 million cells were used in the wake region. This required 200 million particles taking 18 gigabytes of memory and it is reported that regions of the flow may still be under-resolved [63].

Thus, there is substantial efficiency to be gained if the DSMC method is restricted only to that portion of the flow field where significant non-equilibrium effects are felt.

For cases **M3**, **M6**, and **M12**, an initial NS solution is obtained on the baseline DSMC mesh. Figures 5.2, 5.3, and 5.4, plot contours of Kn_{GL} computed using the initial NS solutions for each case (Eq. 3.3). Figures 5.2, 5.3, and 5.4, also plot contours of the error in these NS solutions compared with full DSMC solutions. The error E is defined as

$$E = \max \left| \frac{Q_{NS} - Q_{DSMC}}{Q_{DSMC}} \right| \quad \text{where} \quad Q = \rho,\ T,\ \text{or}\ |V| \quad . \tag{5.1}$$

Since the contours depict a maximum of several quantities, any apparent discontinuities in Figs. 5.2, 5.3, and 5.4 are regions where a new quantity has become dominant. As expected, the breakdown parameter is highest in the shock, fore-body boundary layer, and in the near-wake. Regions where Kn_{GL} is large are seen to correspond very well with regions of high error, E. In the near-wake region the error in the NS solution is seen to range from 10 to 50% for each case. It is also evident that the higher the Mach number, the larger the regions of high error E. The dominant error in the near-wake region is determined to be the velocity magnitude, which would likely improve with the use of slip boundary conditions in the NS solver. It should be noted in Fig. 5.2 that along the outflow boundary below the cylinder (and after the bow shock) contours of error E are visible. This error is a consequence of not positioning the outflow boundary where free-stream conditions are present (sufficiently far from the cylinder). Although flow at the outflow boundary is supersonic, kinetic simulation requires particles to enter the simulation through such a boundary. The boundary procedure used in MONACO does not properly account for this, which is

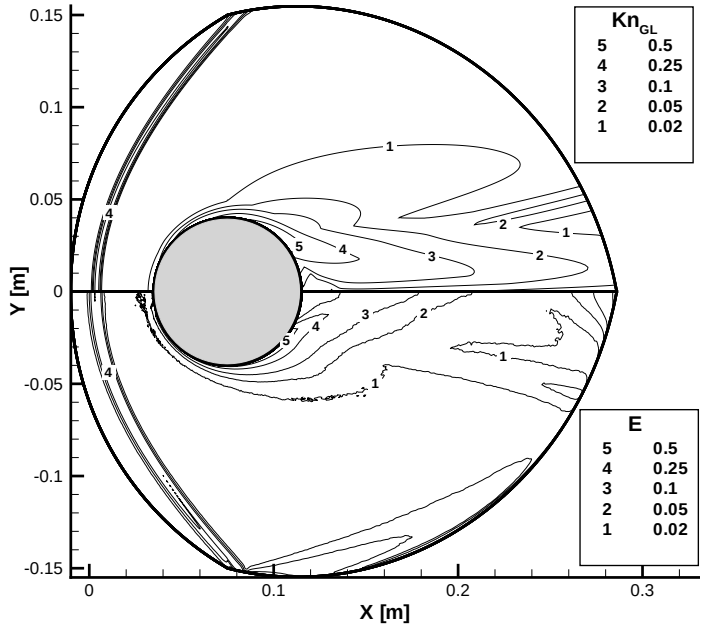


Figure 5.2: Continuum breakdown and modelling discrepancy for case **M3**.

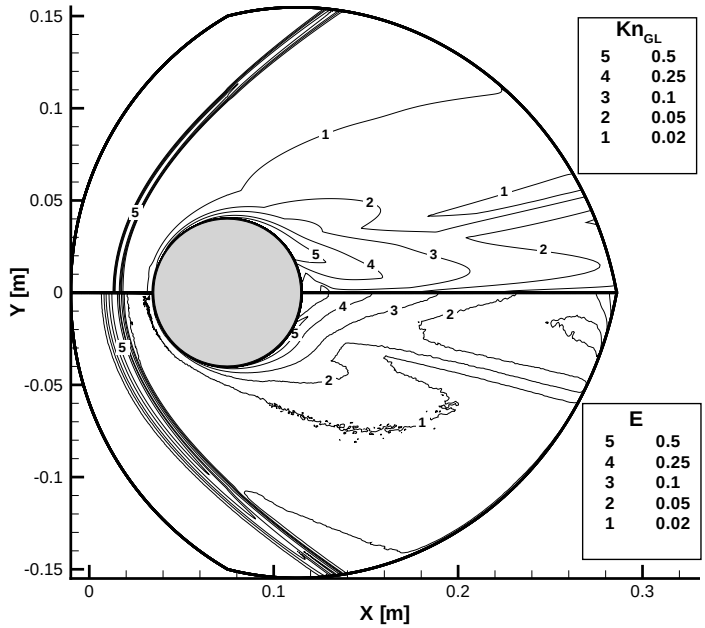


Figure 5.3: Continuum breakdown and modelling discrepancy for case **M6**.

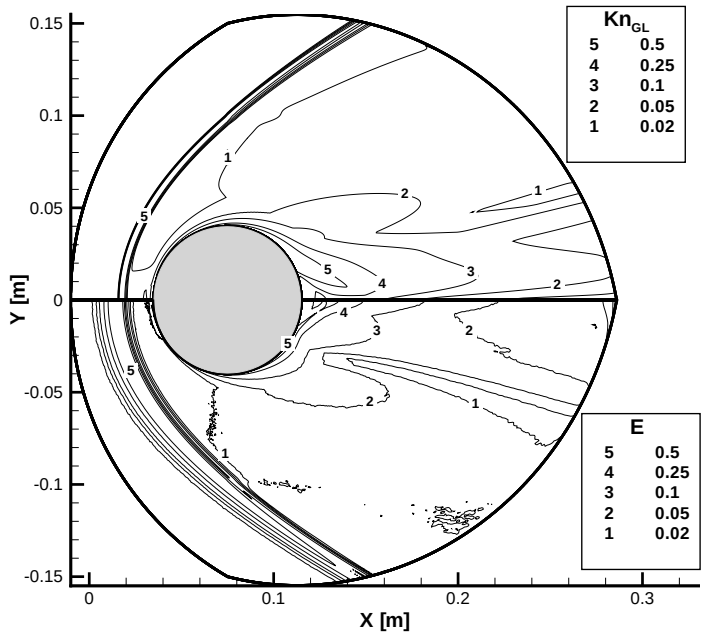


Figure 5.4: Continuum breakdown and modelling discrepancy for case **M12**.

the source of the discrepancy between DSMC and NS solutions at this location. This error is verified to not affect the remainder of the supersonic flow field and for the higher Mach number flows, as seen in Figs. 5.3 and 5.4, this effect becomes even less noticeable.

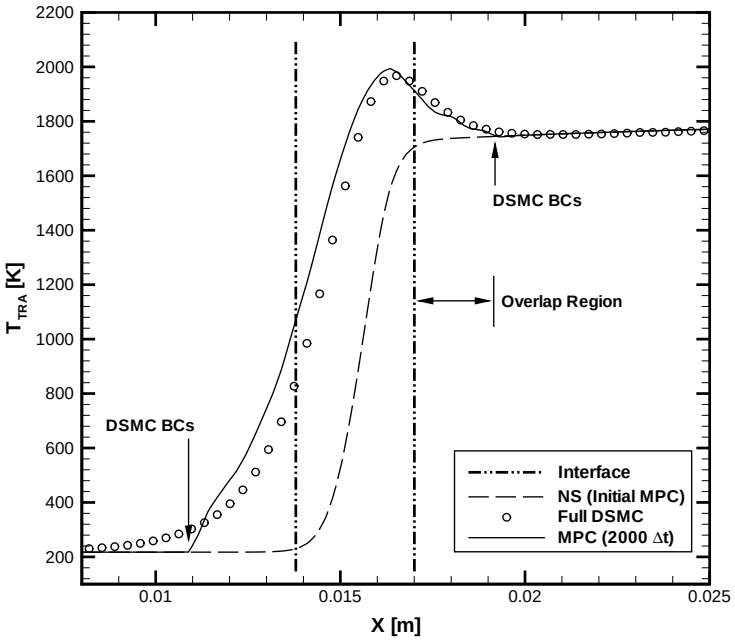
Quantitatively, Figs. 5.2, 5.3, and 5.4 reveal that the breakdown parameter predicts the *magnitude* of error quite well. More specifically, Kn_{GL} is found to uniformly over-predict the error by approximately 40%. That is, the contours of Kn_{GL} match quite closely the contours of $1.4 \times E$. These results lend further support to previous recommendations [50, 51] as well as the results presented in Chapter IV of setting continuum breakdown cutoff at $Kn_{GL} = 0.05$. Thus any cell in which the value of $Kn_{GL} > 0.05$ is set as a particle (DSMC) cell and the remaining cells are set as continuum (NS) cells. The initial and final interface locations for case (**M6**) were shown earlier in the bottom portion of Fig. 3.1.

5.2 Development of the MPC Numerical Cycle

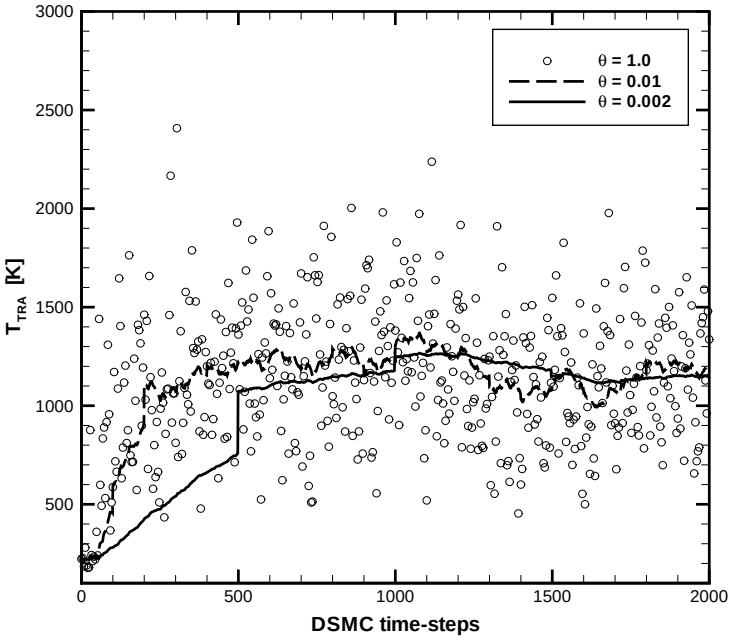
Perhaps the most important aspect of a hybrid particle-continuum method is the hybrid numerical cycle, which incorporates the various components (outlined in Chapter III) such that the hybrid solution advances in a robust manner. Specifically, the numerical cycle must determine *when* to transfer information between particle and continuum regions and *when* to update interface locations. In Chapter IV, a tightly-coupled approach was used successfully to simulate 1D normal shock waves. However, results indicated that a loosely-coupled approach may also be sufficient. Additionally, hybrid simulation of 2D and axi-symmetric blunt-body flows is significantly more complex than simulation of 1D normal shock waves. Thus, in this section, the MPC numerical cycle is re-evaluated for 2D and axi-symmetric flows. The finalized MPC numerical cycle *loosely* couples particle and continuum regions and ensures that the hybrid solution progresses towards the correct non-equilibrium result.

5.2.1 Progression of the MPC solution

As previously described, the MPC cycle begins with a continuum NS solution to the problem. The cutoff for continuum breakdown, $Kn_{GL} > 0.05$, is then applied to initially setup particle and continuum regions and an overlap region is extended by $n_{\text{overlap}} = 10$ cells. Next, particles are generated throughout the newly created particle region based on the NS solution as outlined in Chapter III. The MPC cycle then proceeds to simulate particle regions using the DSMC method *without any coupling to the continuum regions*. The results for translational temperature along a stagnation line for the **M6** case are shown in Fig. 5.5(a). Here the DSMC region has been simulated for 2000 timesteps and the particle distributions have evolved significantly



(a) T_{TRA} profile along stagnation line.



(b) Sub-relaxation average in pre-shock NS boundary cell.

Figure 5.5: Progression of shock profile in DSMC regions (case **M6**).

away from their initial continuum distributions towards the correct non-equilibrium distributions. This important behavior was previously observed in Chapter IV for 1D simulations of normal shock waves. The MPC temperature profile (solid line) in Fig. 5.5(a) is the result of the sub-relaxation average used in each DSMC cell with a value of $\theta = 0.002$. In fact, the temperature variation in the cell at the pre-shock interface is plotted in Fig. 5.5(b) for the same 2000 timesteps. Here, the large discontinuity is the result of the correction used in the sub-relaxation average (Eq. 3.14), which reduces the time-lag of the temporal average. Note that evaluating the sub-relaxation average with a value of $\theta = 1$ (depicted by the symbols in Fig. 5.5(b)) is equivalent to a simple spatial average with large associated statistical scatter. Figure 5.5(a) thus demonstrates how, on their own, DSMC regions do progress towards the correct non-equilibrium solution in a robust and predictable manner.

Ensuring that the coupled hybrid solution continues to progress towards the correct non-equilibrium result and does not diverge towards some other incorrect result is accomplished by carefully controlling *when* information is transferred between particle and continuum regions. Specifically, it is very important that information only be transferred to continuum regions after the interfaces have been positioned in a region of near-equilibrium flow where the NS equations provide an accurate model. Figure 5.6 shows the error introduced when NS boundary conditions are updated while the interfaces are improperly positioned. Here, the interfaces have not been adapted, rather they remain at their initialized location as seen in Fig. 5.5(a). Notice that the post-shock interface lies inside the temperature overshoot (a non-equilibrium region). Figure 5.6 shows an intermediate (t_1) and final steady-state (t_2) solution in the NS regions when this state is used as a NS boundary condition. Clearly, the NS equations are not capable of predicting the tail of the temperature

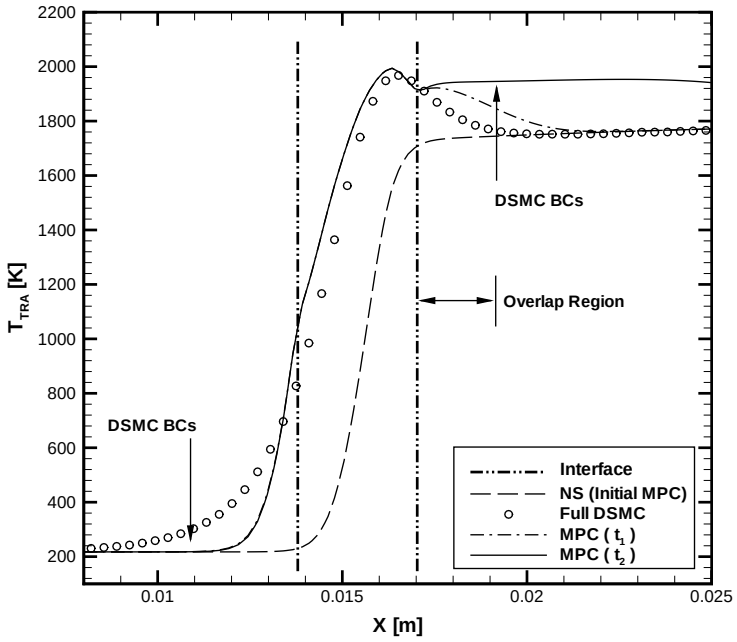


Figure 5.6: Effect of incorrect interface location (case **M6**).

overshoot and instead predict a different post-shock state. This large error will eventually be transmitted back into the particle region via the DSMC boundary cells where it will accumulate and may never leave the simulation. This important issue is not addressed by other researchers who have loosely coupled particle and continuum methods. For example, the hybrid numerical cycle employed by Wu *et al.* [40] involves precisely the same steps that led to the incorrect result shown in Fig. 5.6. Although no such error is evident in their hybrid simulation results, they apply their method to a wedge flow and an expansion jet, neither of which involve a strong bow shock. It is also important to note that for a loosely-coupled cycle, NS regions should be converged significantly before transferring information back to the particle region. For implicit steady-state flow simulations, the numerical waves travelling through the continuum region, such as the solution at t_1 in Fig. 5.6, are not physical and should not be used to generate distributions of particles. In order to avoid

such problems, the MPC method utilizes the overlap region to carefully position the hybrid particle-continuum interfaces before any information is transferred into the continuum domain. In Fig. 5.5(a), since significant flow gradients are now present in the overlap region, application of Kn_{GL} to the MPC profile will convert cells in the overlap region to pure-particle cells. In this case, the particle region would enlarge, a new overlap region would be extended further by a further $n_{\text{overlap}} = 10$ cells, and the process would repeat. At some point, the entire overlap region will be located in a near-continuum region (small flow gradients), and further application of Kn_{GL} will not alter the interface location. It should be noted that the interfaces can be updated far more frequently, effectively adapting *while* the DSMC region is proceeding to steady-state. Only when the interface stops moving, and is ensured to be in a near-continuum region, does the MPC method transfer information to the continuum domain.

This loosely-coupled progression and relaxation procedure is depicted in Fig. 5.7. Here, the translational temperature profile along line $L1$ in the fore-body flow, shown previously in Fig. 5.1, is plotted as the DSMC region is simulated (without any information transfer). Again, the MPC solution is initialized to the NS shock profile. As iterations are performed, the solution begins to relax towards the correct non-equilibrium temperature profile predicted by a full DSMC simulation. As the MPC solution changes, gradients spread into the overlap regions which causes the interfaces to move and the DSMC region to grow. At the same time, the temperature profile in the boundary layer also transitions from the original NS profile towards the DSMC result which is seen to have a small temperature jump at this location on the cylinder surface. This same relaxation process occurs in the near-wake region and is shown in Fig. 5.8 along line $L2$ from Fig. 5.1. Here DSMC results show

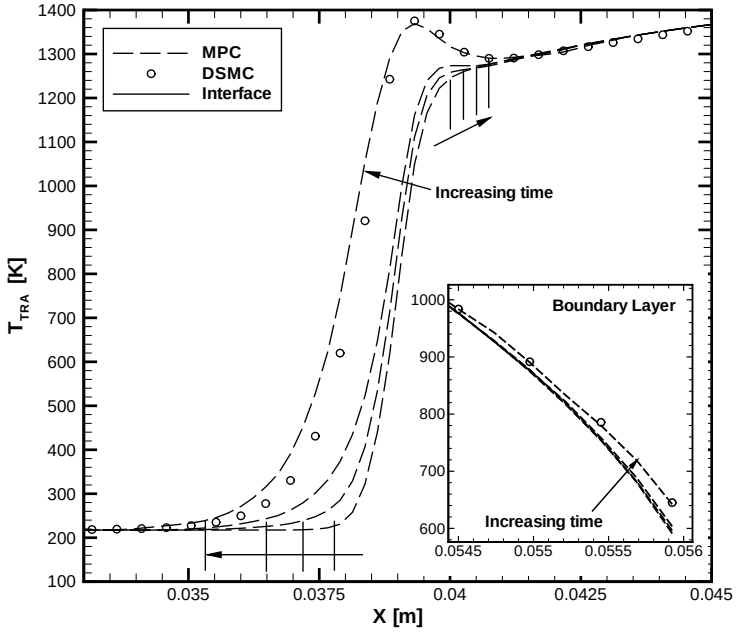


Figure 5.7: Progression of the MPC solution in the fore-body (case **M6**).

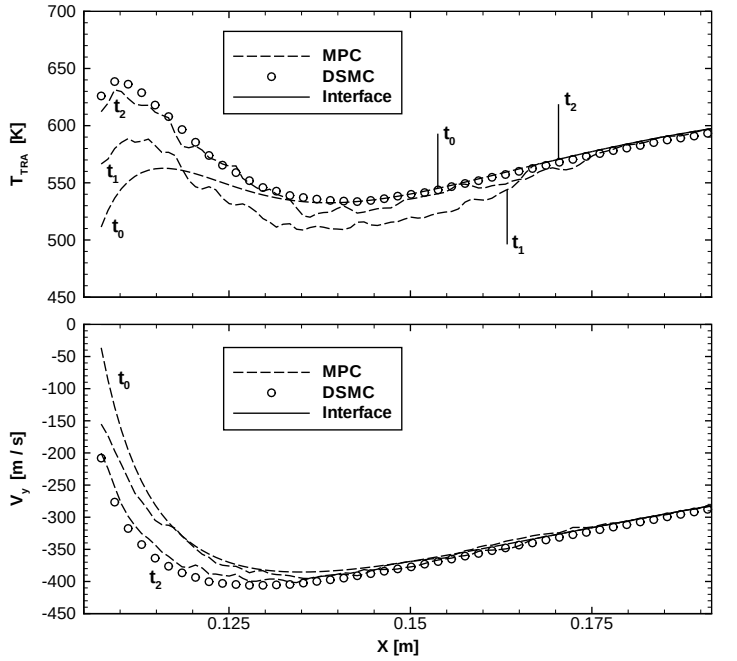


Figure 5.8: Progression of the MPC solution in the near-wake (case **M6**).

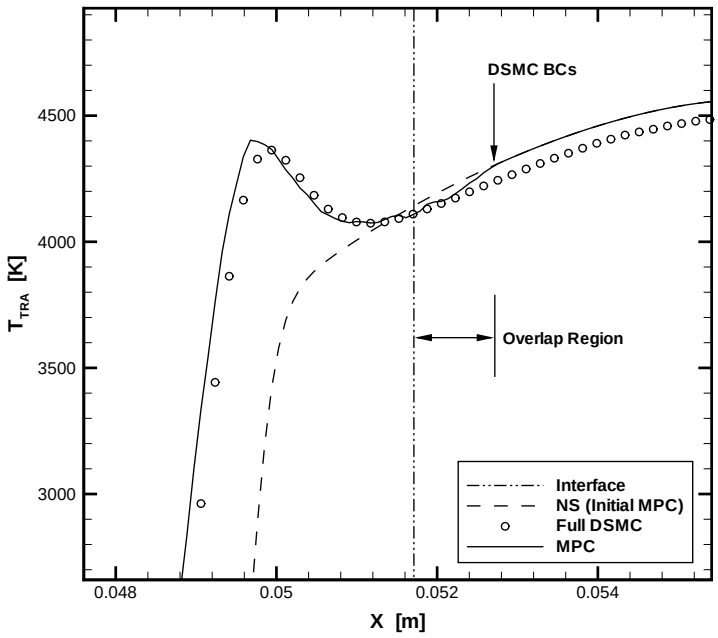


Figure 5.9: T_{TRA} profile along line $L1$ (case **M12**).

a much greater temperature jump and velocity slip at the wall. The MPC solution is seen to progress significantly from the initial no-slip NS solution towards the full DSMC solution. Thus the hybrid solution progresses in the proper direction simply by allowing the DSMC simulation particles to fully relax to their non-equilibrium distributions (without coupling) while utilizing an overlap region to properly position the interfaces.

Along the stagnation line in Fig. 5.5(a), the post-shock state for both continuum and particle solutions is identical and thus information transfer will not alter the post-shock NS boundary condition. However, along line $L1$ in the fore-body flow (refer to Fig. 5.1) the gas immediately expands after passing through the shock. For high Mach number flow (case **M12**), this expansion induces thermal non-equilibrium and as seen in Fig. 5.9 the post-shock state predicted by DSMC is now slightly different. Since significant energy remains 'frozen' in the rotational mode, the translational

temperature predicted by DSMC is lower than the single temperature predicted by the NS equations. Figure 5.9 adds further evidence that incorrect boundary conditions imposed on the DSMC region have only a small influence on neighboring interior cells. While downstream of the overlap region the MPC profile is fixed to the NS solution, within the overlap region the MPC solution quickly recovers and is in very good agreement with the full DSMC result. Thus, as well as enabling the interfaces to adapt, the overlap region also serves to isolate the interior DSMC region from incorrect boundary conditions supplied from the NS solution. In Fig. 5.9, when information is transferred at the interface, the NS temperature boundary condition is now lower. This will eventually shift the entire profile in the NS region into better agreement with the full DSMC solution, and as a result, will then improve the DSMC boundary conditions as well.

5.2.2 Accounting for thermal non-equilibrium

As described in the previous section, it is important that interfaces are located in near-equilibrium regions before information is transferred to the continuum domain. However, the continuum breakdown parameter defined in Eq. 3.3 does not account for thermal non-equilibrium and as a result, it has trouble positioning the post-shock interface correctly in blunt-body flows. As suggested in Chapter IV, the MPC method adds a condition of thermal equilibrium to the breakdown parameter which now becomes

$$Kn_{GL} = \max\left(Kn_{GL-\rho}, Kn_{GL-T}, Kn_{GL-|V|},\ 5 \times \frac{T_{TRA} - T_{ROT}}{T_{ROT}}\right) . \qquad (5.2)$$

In continuum (NS) regions, $T_{TRA} = T_{ROT} = T$, however in particle regions these temperatures may not be equal. Only the very strong thermal non-equilibrium in the shock wave needs attention. This is a compression region in which $T_{ROT} < T_{TRA}$,

whereas the remainder of the flow field is in expansion non-equilibrium with $T_{ROT} > T_{TRA}$. In order to focus only on the shock region, the temperature difference in Eq. 5.2 is calculated without absolute value signs. Thus, in regions where $T_{TRA} > T_{ROT}$ by more than 1%, the breakdown parameter is given a value, $Kn_{GL} > 0.05$. The continuum breakdown parameter expressed by Eq. 5.2 is used in all 2D and axi-symmetric MPC simulations presented in this thesis.

5.2.3 Loosely-coupled MPC numerical cycle

The above ideas and components are incorporated in the MPC numerical cycle, which is completely detailed by the following 4 steps.

1. Obtain a continuum solution to the problem by solving the NS equations. Use the continuum breakdown parameter to initialize DSMC regions and generate particles consistent with the NS solution.

2. Simulate particle regions using DSMC with the current NS solution for boundary conditions. Use the continuum breakdown parameter and an overlap region to adapt the interfaces without transferring any information to the NS regions.

3. After the DSMC solution and interfaces stop changing, use the current DSMC solution to update the NS boundary conditions. Proceed to significantly converge the NS region.

   IF execution of step (2), using this new NS solution, does not modify the interface locations $\longrightarrow$ continue to (4).

   ELSE $\longrightarrow$ return to (2).

4. Since the DSMC solution and interface locations are no longer changing *and* the NS region is converged, steady-state has been reached. Lock the interfaces,

remove the overlap regions, and cycle both the DSMC and NS solvers (coupling occasionally) until the DSMC scatter and NS residual fall below threshold values.

The implementation of step (1) has already been outlined in previous sections. The precise implementation of step (2) is detailed by the following steps:

(2a) Cycle the DSMC solver for one timestep with boundary conditions set using the current NS solution. Use the sub-relaxation average to determine macroscopic properties in each DSMC cell.

IF any flow quantity in any cell touching the interface changes by more than 20% *or* the DSMC solver has cycled for $\frac{1}{\theta}$ timesteps $\longrightarrow$ continue to (2b)

ELSE $\longrightarrow$ return to (2a)

(2b) Apply the breakdown parameter to the current MPC profile and re-label cells as DSMC or NS.

(2c) Extend an overlap region and re-label DSMC and NS boundary cells.

(2d) Generate particles in any *newly-created* DSMC cells based on the corresponding state from the NS solution.

IF the # of DSMC cells has changed by more than 1% $\longrightarrow$ return to (2a).

ELSE if the # of DSMC cells has changed by less than 1% $\longrightarrow$ continue to (3).

For cases **M3**, **M6**, and **M12**, $n_{\text{overlap}} = 10$ cells are used and θ is set to 0.002. Applying the breakdown parameter after every 20% change allows the interfaces to adapt *while* the DSMC solution is proceeding towards steady-state. If no significant changes are detected, the DSMC regions are allowed to iterate for $\frac{1}{\theta}$ timesteps

at which point the next correction to the sub-relaxation average is made (see Fig. 5.5(b)). This allows sufficient time for the DSMC solution to change further (in case it has not yet reached steady-state) before re-application of the breakdown parameter. Currently, in step (3), the state in each NS boundary cell is completely specified from the DSMC solution in both supersonic and subsonic regions of the flow. As mentioned in Chapter III, although this is not found to cause any stability problems, it does prevent the NS solution from converging to machine zero. Instead, during step (3), 500 implicit timesteps are taken using a CFL value of 10.0. This converges the solution by roughly 3 orders of magnitude after which no visible changes in the NS regions are observed. Finally in step (4), since the interfaces are now located in regions of near-equilibrium flow and are no longer moving, the overlap regions are not necessary and are removed. Depending on the size of the overlap region, this can eliminate a substantial number of particles for the remainder of the MPC simulation. Now that the interfaces are locked in place and the DSMC solution has reached steady-state, the sub-relaxation average is no longer required. Instead, standard DSMC cumulative sampling is used to calculate the macroscopic flow quantities. Every 5000 DSMC timesteps, the sampled information is transferred to the NS regions which are converged for 100 implicit timesteps. This cycle repeats and continued sampling is able to reduce the statistical scatter to an arbitrarily low value. It is very important to note that since the MPC method is designed to simulate steady-state flows, it is not necessary to simulate each region for the same physical time. As the scatter in the DSMC solution is eliminated, so too is the scatter transferred to the NS regions. The result is a scatter-free DSMC solution in non-equilibrium regions that transitions to a converged NS solution in continuum regions.

5.3 2D Cylinder Simulation Results

In this section, the MPC method will be shown capable of reproducing full DSMC simulation results for microscopic velocity distribution functions (vdfs), macroscopic flow field quantities, and surface properties. As a reference, the final interface locations for simulations **M3**, **M6**, and **M12**, are shown in the bottom of Fig. 5.10. As the flow Mach number is increased, the size of the non-equilibrium regions simulated using DSMC increases. For the **M3** case, the particle region terminates before the domain exit. By comparison, the **M12** case requires a larger particle region where the shock and boundary layer particle regions actually merge in the fore-body.

5.3.1 Velocity distribution functions

This thesis presents, for the first time in the literature, a detailed analysis of the particle velocity distribution functions (vdfs) predicted by a hybrid method. The motivation behind a hybrid particle-continuum method is the fact that particle simulation is only required where particles have significantly non-equilibrium (non-Maxwellian) vdfs. Indeed, in near-equilibrium regions, the NS equations contain all of the necessary information to describe the correct vdf. Since hybrid interfaces lie in near-equilibrium regions, the NS equations should be capable of generating the exact particle vdfs at the interface that are seen in a full DSMC simulation. In order to demonstrate this for the **M6** case, Fig. 5.11 plots the local vdfs at positions **b**, **c**, **g**, and **h** (seen in Fig. 5.10) for both MPC and full DSMC simulations. Figures 5.11(a), 5.11(b), and 5.11(c) plot the distributions of particles generated in the DSMC boundary cells by the MPC method at locations **b**, **c**, and **g** respectively. These distributions are seen to be identical to the particle distributions present in the same cell of a full DSMC simulation. As expected, at hybrid particle-continuum

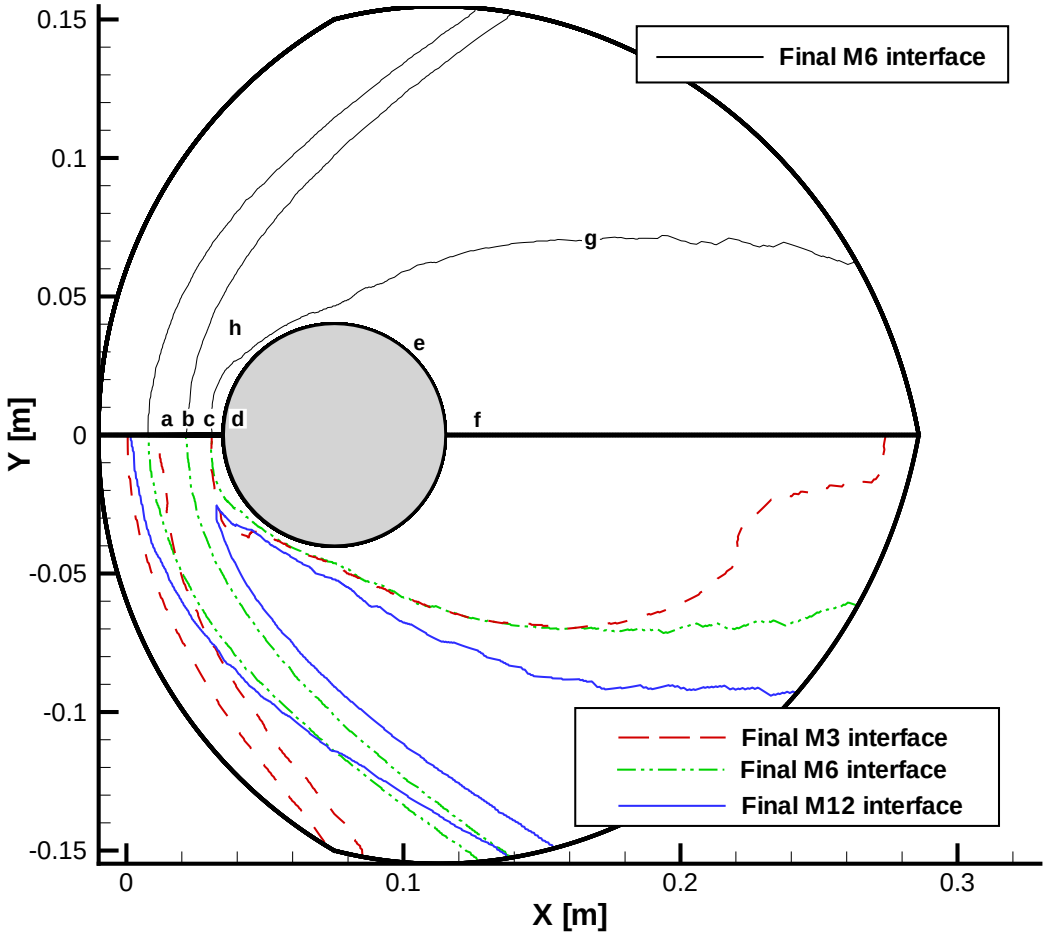
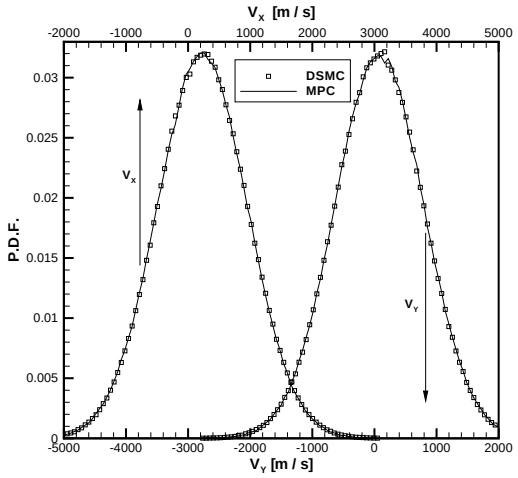
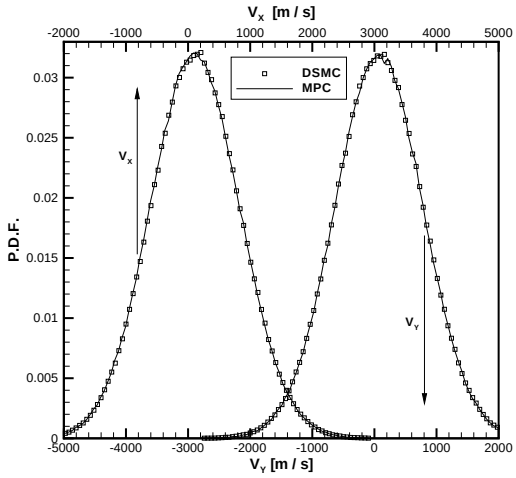


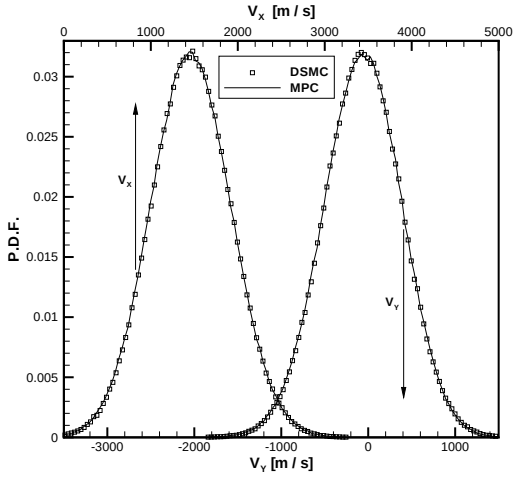
Figure 5.10: Final MPC interface locations.



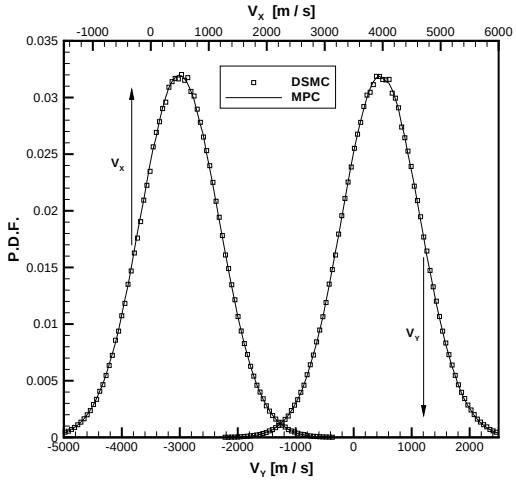
(a) Local vdf at b.



(b) Local vdf at c.



(c) Local vdf at g.

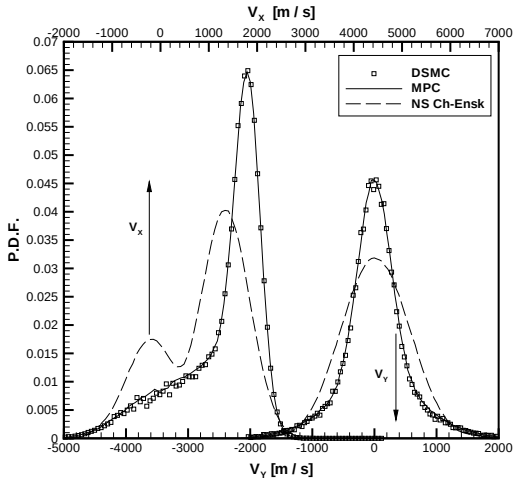


(d) Local vdf at h.

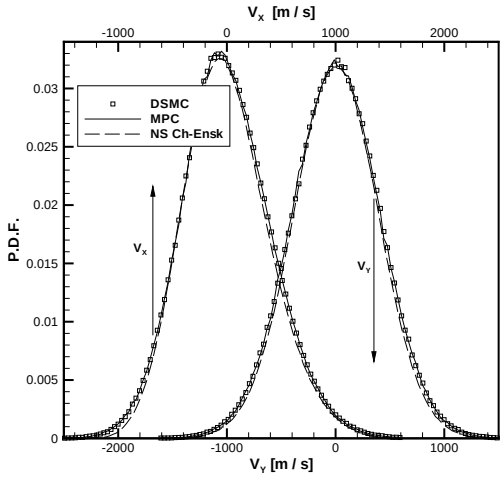
Figure 5.11: Local velocity distribution functions in near-equilibrium regions for case **M6**.

interfaces, particles are seen to have near-equilibrium vdfs and therefore the NS regions are capable of providing accurate boundary conditions for the DSMC regions. Within the continuum region between the shock and boundary layer (location **h**), the velocity distribution sampled from a full DSMC simulation is seen to be a near-equilibrium distribution. The MPC method solves the Navier-Stokes equations at location **h** which are in fact modelling the correct velocity distribution function in this continuum region of the flow. Thus, if the hybrid-DSMC regions seen in Fig. 5.10 are surrounded by boundary cells containing the same particles that are present in full DSMC simulations, the interior particle solution should also agree exactly with a full DSMC simulation. In order to demonstrate this, the vdfs in various non-equilibrium regions (locations **a**, **d**, **e**, and **f**) are plotted in Fig. 5.12.

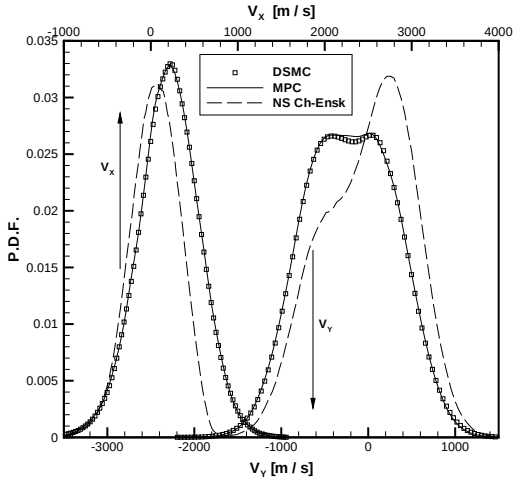
The first observation from all vdfs plotted in Fig. 5.12 is that the MPC particles do indeed have the same velocity distribution as particles within a full DSMC simulation. The most severe non-equilibrium region is in the shock center along the stagnation streamline (region **a**), shown in Fig. 5.12(a). In order to adjust for the different shock positions between DSMC and NS solutions, the shock center is taken as the point where $T_{TRA} = 0.5(T_{TRA\ post} + T_{TRA\ pre})$ for DSMC and MPC, and where $T = 0.5(T_{post} + T_{pre})$ in the NS solution. All dotted lines shown in Fig. 5.12 are Chapman-Enskog distributions generated [56] using the local state and gradients of a full NS solution to the problem. In Fig. 5.12(a), the vdf in the x-direction shows a peak of fast moving particles, but also a significant number of post-shock particles that have travelled upstream through the thin shock and maintained their low speed. The Chapman-Enskog distribution effectively assumes more collisions and thus the fast moving peak is lowered and the slow moving region of the vdf is raised. Certainly the NS equations do not contain the information needed to describe the



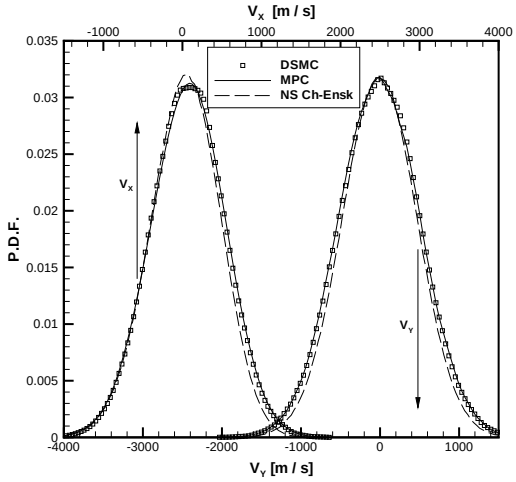
(a) Local vdf at a.



(b) Local vdf at d.



(c) Local vdf at e.



(d) Local vdf at f.

Figure 5.12: Local velocity distribution functions in non-equilibrium regions for case **M6**.

vdf in either the x or y direction inside the shock. The velocity distribution in a cell at the stagnation point (location **d**) is plotted in Fig. 5.12(b). Here the flow is highly compressed and is expected to be in near-equilibrium. Fig. 5.12(b) confirms that the velocity distributions predicted by both DSMC and NS (Chapman-Enskog) agree very well. Close inspection reveals that the DSMC distributions are slightly wider and that the MPC method very accurately reproduces the vdf predicted by full DSMC simulation. Another region of the flow that is highly non-equilibrium is region **e**, located 135^o around the cylinder surface. The local vdfs are plotted in Fig. 5.12(c) where the Chapman-Enskog vdfs are centered around zero velocity due to the no-slip condition in the NS solver. In addition to significant velocity slip, evident in the DSMC and MPC vdfs, Fig. 5.12(c) reveals that the vdf in the y direction is quite far from having an equilibrium shape. Finally, the local vdf in a cell 2 cm directly behind the cylinder (region **f**) is plotted in Fig. 5.12(d). Here the DSMC and MPC vdfs appear quite similar to the Chapman-Enskog distribution from the NS solution, only they are slightly wider. It will be shown in the next section that this width does indeed correspond to a higher temperature in this region. It is important to note that because of the large velocity scales in these vdf plots, it is difficult to determine the degree to which small variations in a vdf translate into noticeable macroscopic effects. For this reason the flow field and surface properties are now investigated.

5.3.2 Flow field quantities

The purpose of the MPC method is ultimately to reproduce the flow field and surface property results of a full DSMC simulation in less time. The T_{TRA} fields calculated by full DSMC, full NS, and MPC methods are compared in Fig. 5.13 for the **M12** case. The DSMC temperature shock profile is significantly thicker and begins

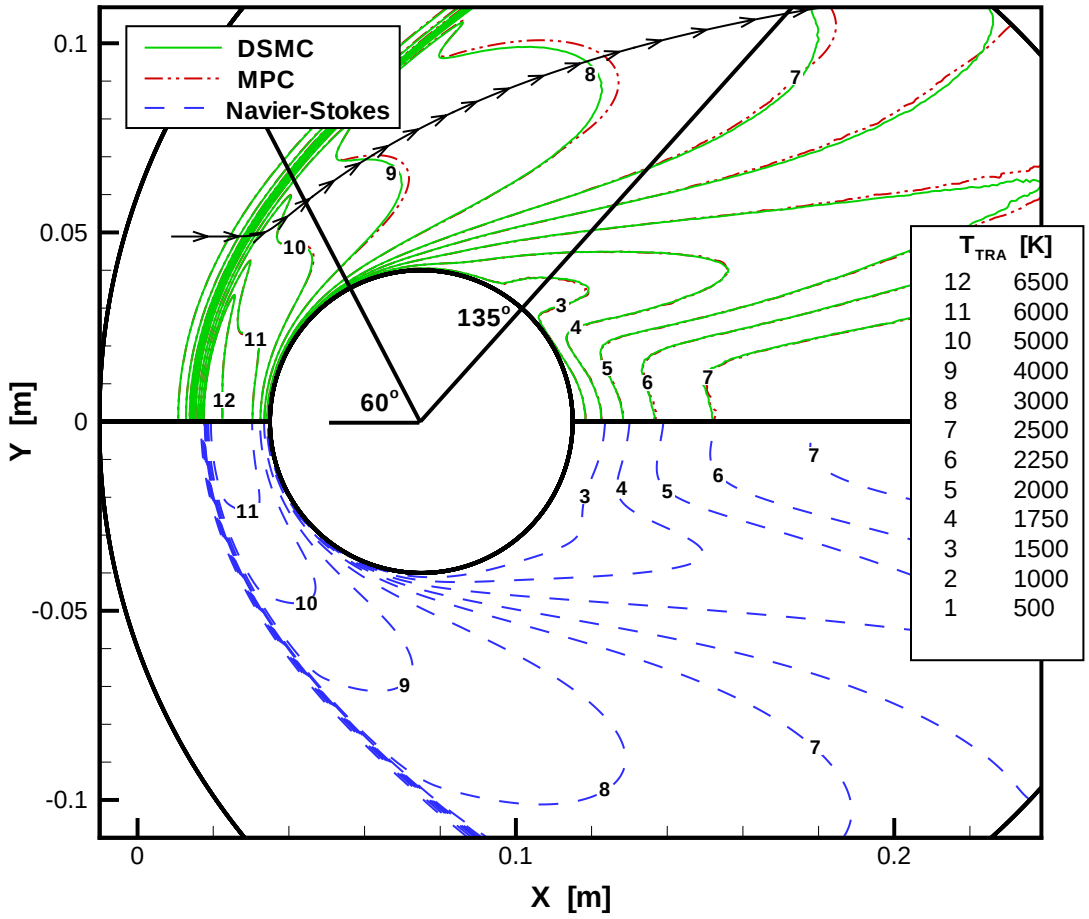
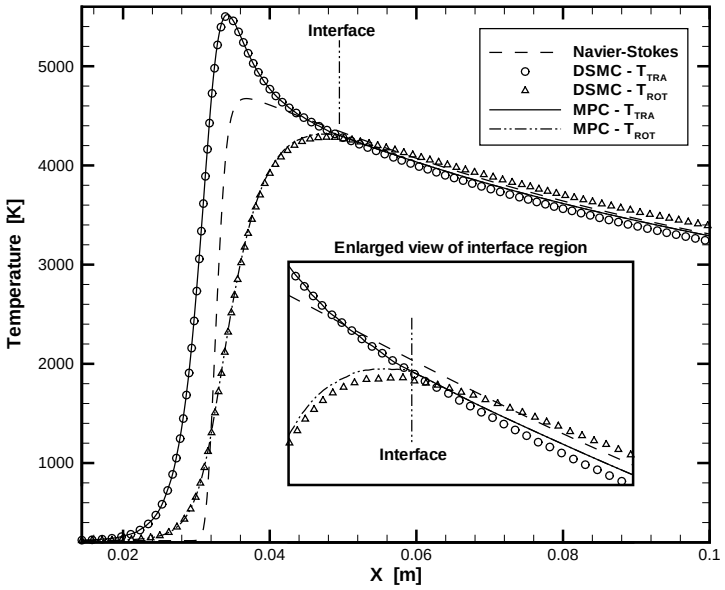
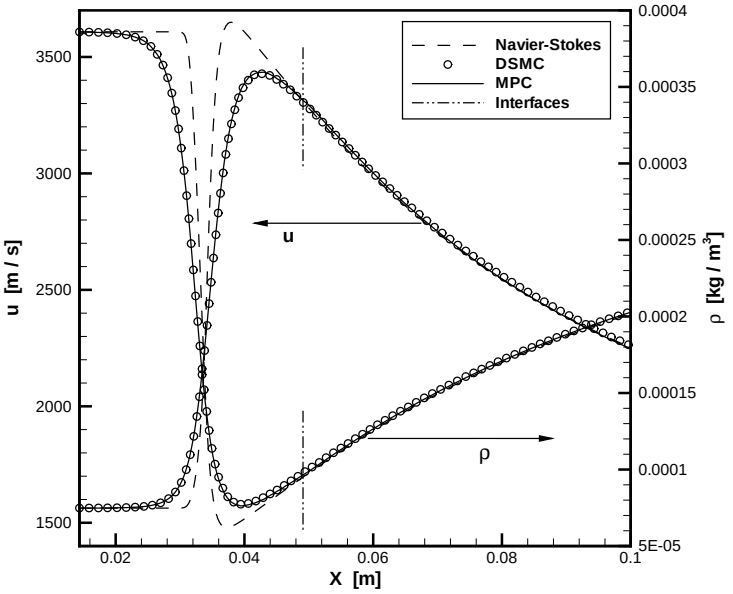


Figure 5.13: Validation for Mach 12 translational temperature field.

further upstream compared with the NS shock profile. In addition, the translational temperature overshoot is clearly visible in the DSMC solution. The wake region also shows that the temperature predicted by DSMC is approximately $10 - 15\%$ higher than that predicted by the NS solver. Qualitatively, very similar results are obtained for cases **M3** and **M6**. Figure 5.13 also shows how the MPC method successfully reproduces a full DSMC solution with little error. In fact the largest error at any location in the flow, for any flow variable, (see Eq. 5.1, replacing Q_{NS} with Q_{MPC}) is determined to be less than 3.0%, 2.5%, and 1.5% for cases **M12**, **M6**, and **M3** respectively. Moreover, the largest error is in the T_{TRA} field which is maximum in MPC-continuum regions (away from the surface) where the NS equations do not model thermal non-equilibrium. The flow properties along a streamline (that passes through this region of maximum error) are plotted in Figs. 5.14(a) and 5.14(b). The thermal non-equilibrium responsible for this error can be seen in Fig. 5.14(a). Inside the particle region encompassing the shock, the MPC method precisely reproduces full DSMC results for both T_{TRA} and T_{ROT}. These non-equilibrium profiles are seen to be significantly different from the single temperature shock profile predicted by the NS equations. As a result of the condition of thermal equilibrium added to the continuum breakdown parameter in Eq. 5.2, the final post-shock interface is positioned where $T_{TRA} = T_{ROT}$. On the other side of the interface, full DSMC results (symbols) show thermal non-equilibrium in the expanding continuum region. Since the rotational energy mode relaxes slower than the translational mode, T_{ROT} remains higher than T_{TRA}. The NS module of the MPC solver assumes a single temperature which can be seen to lie between T_{TRA} and T_{ROT} in the final MPC solution. This assumption of a single temperature is the source of the 3% error in the **M12** simulation. It is also important to note that in the continuum region, the MPC method



(a) Temperature profiles and thermal non-equilibrium.

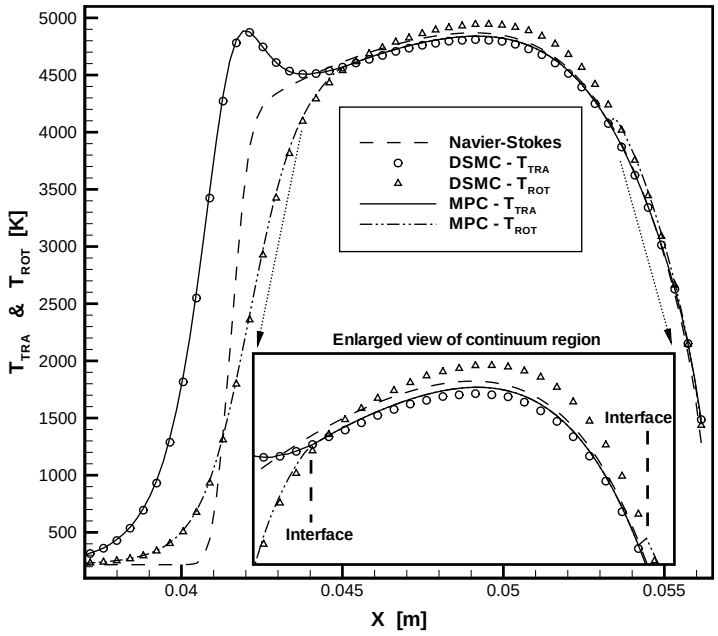


(b) Density and velocity profiles.

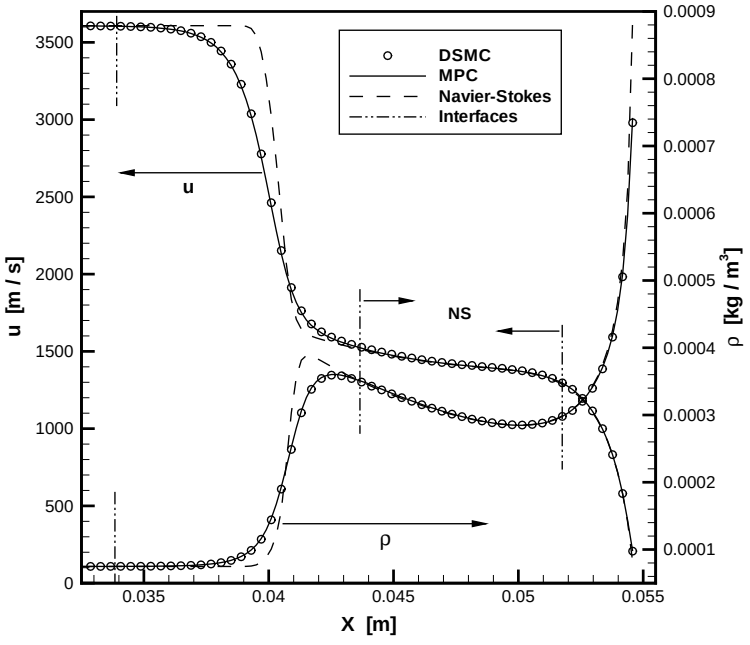
Figure 5.14: Properties along a streamline (see Fig. 5.13) for case **M12**.

has shifted the initial NS profile into better agreement with DSMC results. To verify that the error in other flow properties is much less than for T_{TRA}, the density and bulk x-velocity (u) along the streamline are plotted in Fig. 5.14(b). Here, in both particle and continuum regions, the MPC results are seen to exactly reproduce full DSMC results. It is important to note that along this streamline (away from the surface), DSMC results only differ from NS results inside the bow shock. It is the flow field properties closer to the cylinder surface that are of more interest.

Both the translational and rotational temperature profiles are plotted in Fig. 5.15(a) for each method along the 60^o line shown in Fig. 5.13 approaching the cylinder surface. Again, due to the flow expansion after passing through the shock wave, the rotational temperature is slightly higher than the translational temperature. Focusing only on the DSMC results, this behavior can be seen in Fig. 5.15(a). Moving focus to the MPC results, Fig. 5.15(a) shows how the MPC method predicts the correct profiles for both T_{TRA} and T_{ROT} in both the shock and boundary layer particle regions. However, in the MPC-continuum region between these interfaces, the NS equations are solved, using a single temperature that combines both translational and rotational modes. As expected, the MPC temperature profile in this region lies between the T_{TRA} and T_{ROT} profiles predicted by DSMC. Finally, focusing on the full NS solution in Fig. 5.15(a), it is important to notice that at the interface just after the shock, the MPC temperature is not equal to the initial NS temperature. This means that the MPC cycle has succeeded in providing more accurate boundary conditions to the MPC-continuum region. This mechanism was depicted previously in Fig. 5.9 at this specific flow location for the **M12** case. These new boundary conditions were set (or transferred) using the MPC-particle region and result in shifting the MPC-continuum profile to a lower, more accurate translational



(a) Thermal non-equilibrium effects.



(b) Density and velocity profiles.

Figure 5.15: Fore-body flow field profiles along a 60^o cut (case **M12**).

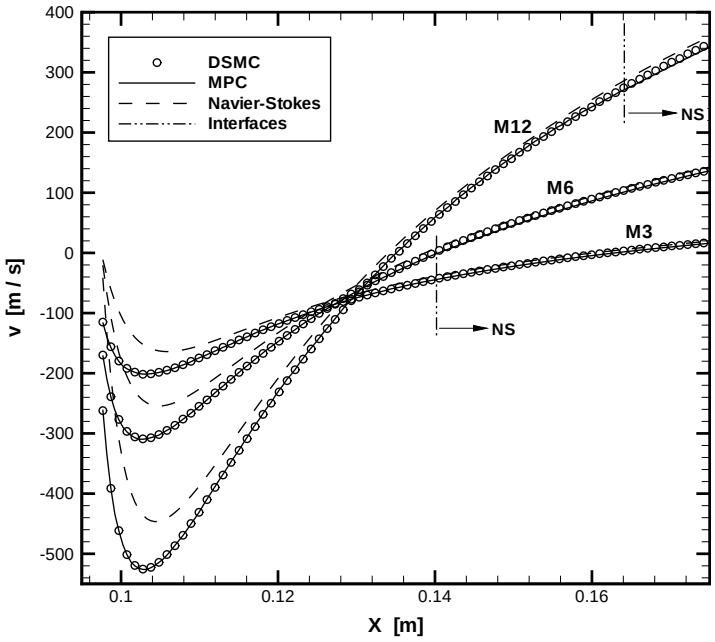


Figure 5.16: v along a 135^o cut (all cases).

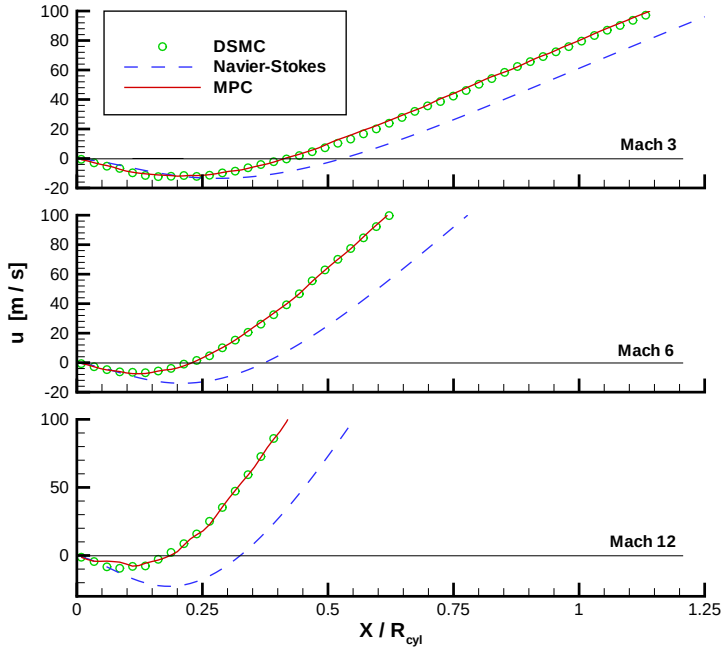
temperature. Again, to verify that the error in other flow properties is much less than for T_{TRA}, the density and velocity profiles (extracted from the 60^o line) and are presented in Fig. 5.15(b). The MPC solution for density and velocity is seen to agree exactly with full DSMC results. These profiles of ρ and u differ from the NS profiles inside the shock and also near the cylinder surface. Since the NS solver does not model temperature slip, the NS equations predict a higher density than DSMC at the cylinder surface.

In order to validate the MPC method for the flow in the near-wake region of the cylinder surface, the bulk y-velocity (v) is extracted from the 135^o cut shown in Fig. 5.13 and is presented in Fig. 5.16 for cases **M3**, **M6**, and **M12**. Figure 5.16 shows that significant slip in y-velocity is predicted by DSMC at a location 135^o around the cylinder surface for all three Mach number flows. The MPC method, which begins with the no-slip NS solution, successfully reproduces these full DSMC results in each

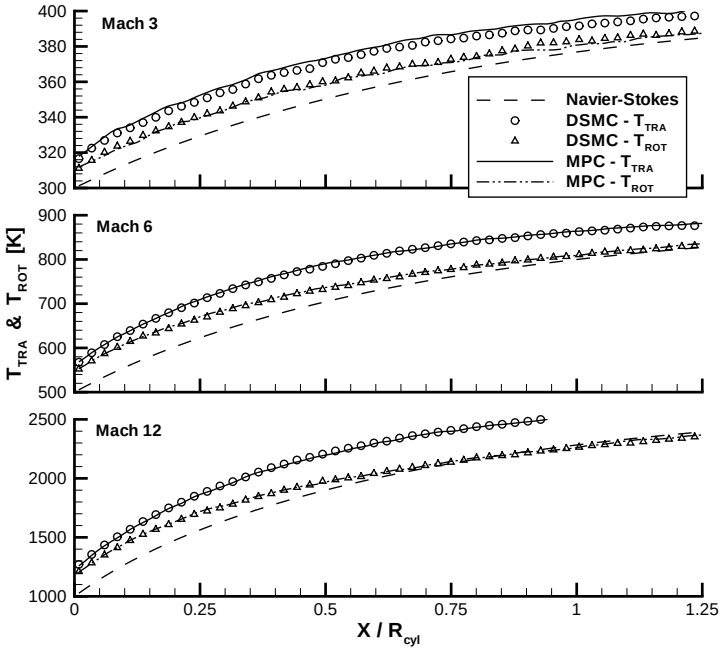
case.

Non-equilibrium effects directly in the wake of the cylinder are portrayed in Fig. 5.17. The x-velocity profiles in the cylinder wake, along the symmetry axis are plotted in Fig. 5.17(a) for the **M3**, **M6**, and **M12** cases. Here, the location at which the x-velocity becomes positive, indicates the location of "wake closure". For each free-stream Mach number, DSMC simulations predict wake closure to occur closer to the cylinder than predicted by a full NS simulation. In addition, higher Mach number flow, associated with larger non-equilibrium effects, is observed to result in a smaller wake closure length. Both of these results agree qualitatively with those presented in Ref. [61] which compares DSMC and NS results for a slightly more complicated geometry at higher flow speeds. The temperature profiles in the near-wake region are plotted in Fig. 5.17(b) for all three cases. The wake flow is seen to be slightly out of thermal equilibrium. In addition, as seen previously in Fig. 5.13 the translational temperature predicted by DSMC is higher than that predicted by a NS solution. Although not shown, the density in this region is correspondingly lower for DSMC, while the pressures predicted by both NS and DSMC remain similar. Finally, for each of the near-wake results plotted in Fig. 5.17, the MPC method is seen to reproduce the DSMC profiles with little error.

Finally, close inspection of the solution in post-shock regions (Figs. 5.9, 5.14, and 5.15) reveals that although information transfer has slightly shifted the boundary conditions for the post-shock continuum region, the shift is very small. This observation begs the question: if the post-shock state for both DSMC and NS solutions is virtually the same, is it even necessary to accurately predict the shock interior? In order to answer this question, two additional MPC simulations are tested. The first, denoted **M12a**, only models the boundary layer and wake as par-



(a) Wake closure.



(b) Near-wake temperature profile.

Figure 5.17: Non-equilibrium effects in the near-wake region.

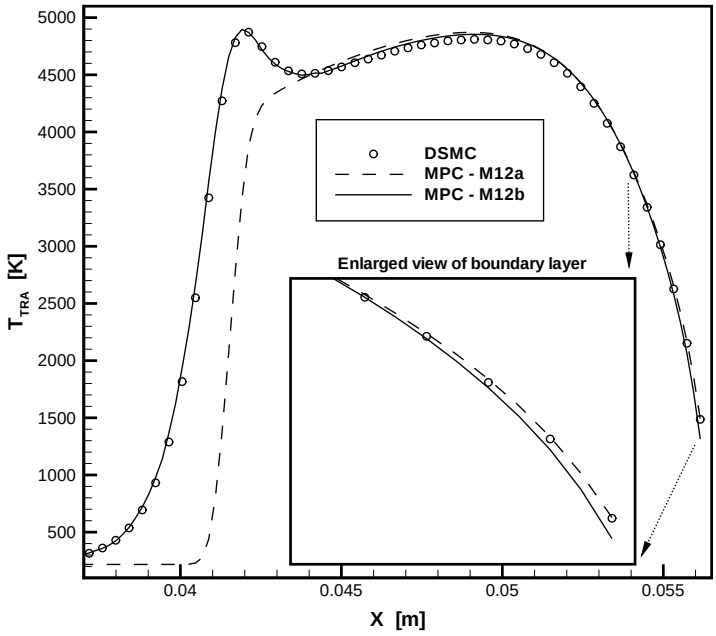


Figure 5.18: Effect of accurately solving the shock interior.

ticle regions. That is, the shock is simulated as a continuum region where the NS equations are solved. This approach may seem similar to the "zonally decoupled" solutions discussed earlier [61, 62, 63], however, the MPC method automatically adapts the interface locations and restricts particle regions only to that portion of the flow field where large non-equilibrium effects are felt. For the **M12** case these regions include the fore-body boundary layer and only a portion of the wake. The second simulation, denoted **M12b**, models only the shock with a particle region and solves the NS equations everywhere else. Both cases use the exact same flow conditions, mesh, and numerical parameters as the **M12** case and only differ in the final MPC interface locations. This is accomplished by evaluating the gradient-length Knudsen number based on pressure, Kn_{GL-p} (refer to Eq. 3.1). This parameter has a much higher value inside the shock compared with any other region of the flow. After setting up particle and continuum regions, cells in which $Kn_{GL-p} > 1.0$ are converted

back to continuum NS cells, resulting in case **M12a**. For case **M12b**, cells in which $Kn_{GL-p} < 1.0$ (everywhere except the shock) are converted to continuum NS cells. The resulting translational temperature profiles along the same 60^o cut are shown in Fig. 5.18. Although case **M12a** does not predict the correct shock profile, since it models the non-equilibrium boundary layer as a particle region, it agrees well with DSMC results near the surface. Conversely, although case **M12b** predicts the correct shock profile, since the post-shock state is only slightly different, the rest of the solution agrees very well with a full NS simulation. Near the cylinder surface, a continuum solution (such as case **M12b**) predicts steeper gradients that may influence the predicted heat transfer and momentum transfer to the spacecraft. Thus, based on the fore-body flow field, modelling the shock wave as a particle region may not be necessary. This will be verified in the next section by examining the predicted surface properties over the entire cylinder.

5.3.3 Surface properties

For the first time in the literature, this thesis also presents a detailed analysis of the surface properties predicted by a hybrid DSMC-NS method. Figures 5.19, 5.20, and 5.21 plot the coefficients of heat transfer (C_q) and shear-stress ($C_{\tau_{||}}$) around the cylinder surface for cases **M3**, **M6**, **M12**, **M12a**, and **M12b**. The coefficients are defined as

$$C_q = \frac{q}{\frac{1}{2}\rho_\infty u_\infty^3} \qquad \text{and} \qquad C_{\tau_{||}} = \frac{\tau_{||}}{\frac{1}{2}\rho_\infty u_\infty^2} \quad , \tag{5.3}$$

where q is the heat transferred to the cylinder per unit time per unit area and $\tau_{||}$ is the tangential momentum per unit time per unit area transferred to the cylinder at each point on the surface. In a NS simulation, both q and τ are determined using gradients of macroscopic flow properties (temperature and velocity). In a DSMC simulation,

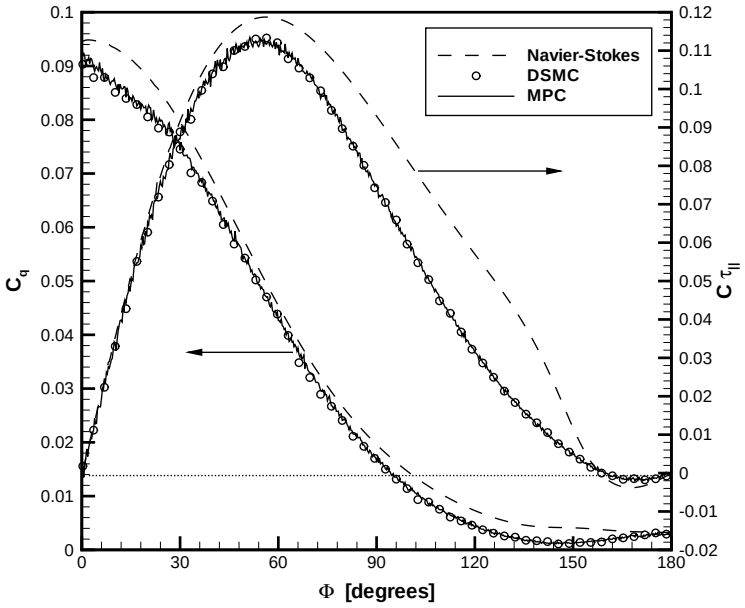


Figure 5.19: Mach 3 surface properties around the cylinder.

q and τ are determined by averaging changes in the kinetic energy and momentum of individual particles as they collide with the surface. In all MPC simulations, as depicted earlier in Fig. 5.10, a particle region envelops the entire cylinder surface. Thus heat and momentum transfer in the MPC simulations are determined in the same way as in a full DSMC simulation. It is evident that for each flow Mach number in Figs. 5.19-5.21, the heat and momentum transfer predicted by DSMC are less than that predicted by the NS solver around the entire cylinder. This is due in part to velocity slip and temperature jump at the surface that is not modelled in the NS simulations but is a natural result of the DSMC simulations. However, it is also due to the fact that in assuming more collisionality, the NS equations predict more rapid flow changes (steeper gradients), which correspond to higher rates of heat and momentum flux. The discrepancy in the peak heating rate is seen to be 4.4%, 10.0%, and 7.5% for cases **M3**, **M6**, and **M12** respectively. The largest difference between

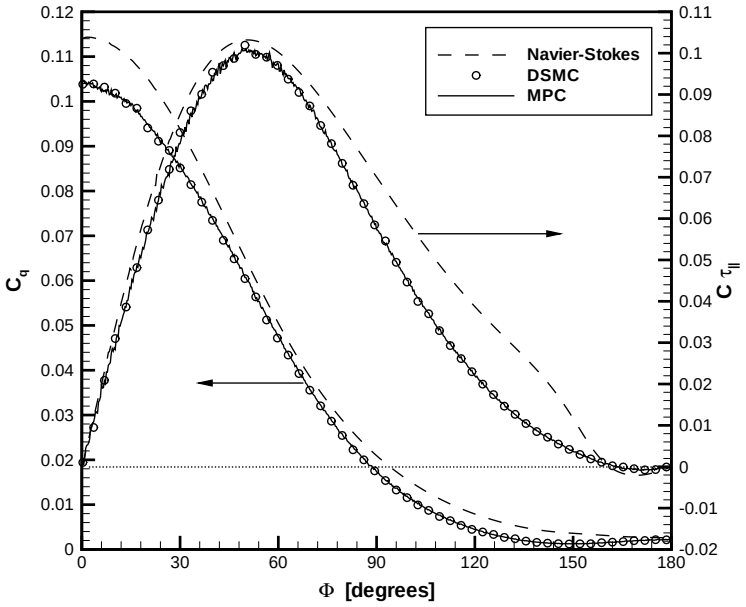


Figure 5.20: Mach 6 surface properties around the cylinder.

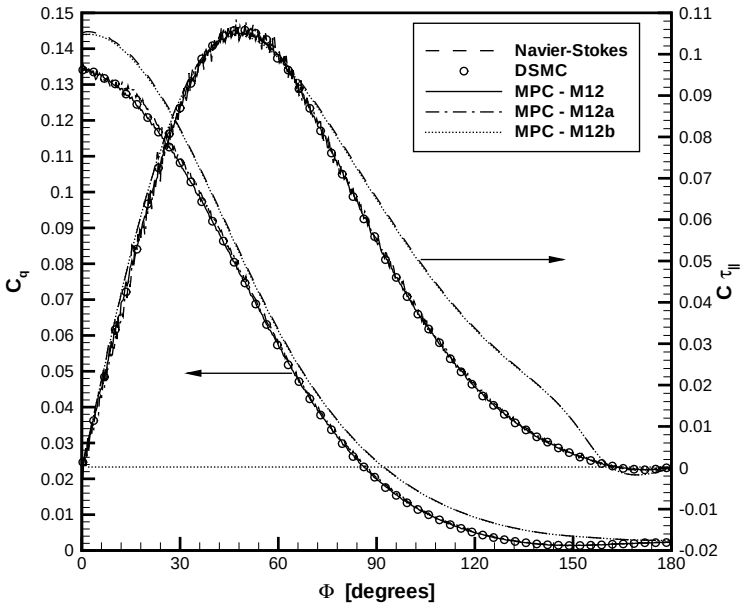


Figure 5.21: Mach 12 surface properties around the cylinder.

DSMC and NS is in the wake of the cylinder where the shear-stress is significantly less according to DSMC. These results are qualitatively and quantitatively similar to those presented in Ref. [60] for simulations of Mach 10 flow of argon around a cylinder. However, as shown in that study, the coefficient of pressure (C_p) predicted by both NS and DSMC is virtually the same. Since the majority of drag on the cylinder is due to pressure drag, despite differences in shear-stress, the total drag on the cylinder remains the same for both NS and DSMC. Again, Figs. 5.19-5.21 demonstrate how the MPC method successfully reproduces DSMC results. What is also evident in Fig. 5.21 is that when the shock is not modelled using a particle region (case **M12a**), the surface properties are still accurately predicted. In the fore-body, up to an angle of approximately 30^o, the heat transfer is seen to be slightly higher, but elsewhere the results are identical. Also, as expected, the **M12b** case, which captures the shock accurately with a particle region but uses the NS equations everywhere else, predicts surface properties consistent with a full NS solver.

5.4 Axi-Symmetric Planetary Probe Simulation

5.4.1 Flow conditions

The planetary probe geometry is detailed in the top of Fig. 5.22 where the diameter of the probe is 5 cm ($R_b = 0.025\ m$), the nose radius $R_n = 0.5 \times R_b$, and s is the distance around the surface. Experimental data was obtained in the SR3 wind tunnel in Meudon, France. Three test conditions were considered where the free-stream was nitrogen at a nominal Mach number of 20 with free-stream densities of 1.7×10^{-5}, 5.2×10^{-5}, and $46.6 \times 10^{-5}\ kg/m^3$, corresponding to Kn_{g} values of 0.03, 0.01, and 0.001 respectively. Experimental data and numerical simulation results for these cases are summarized in Ref. [2]. A "zonally decoupled" approach [61] has

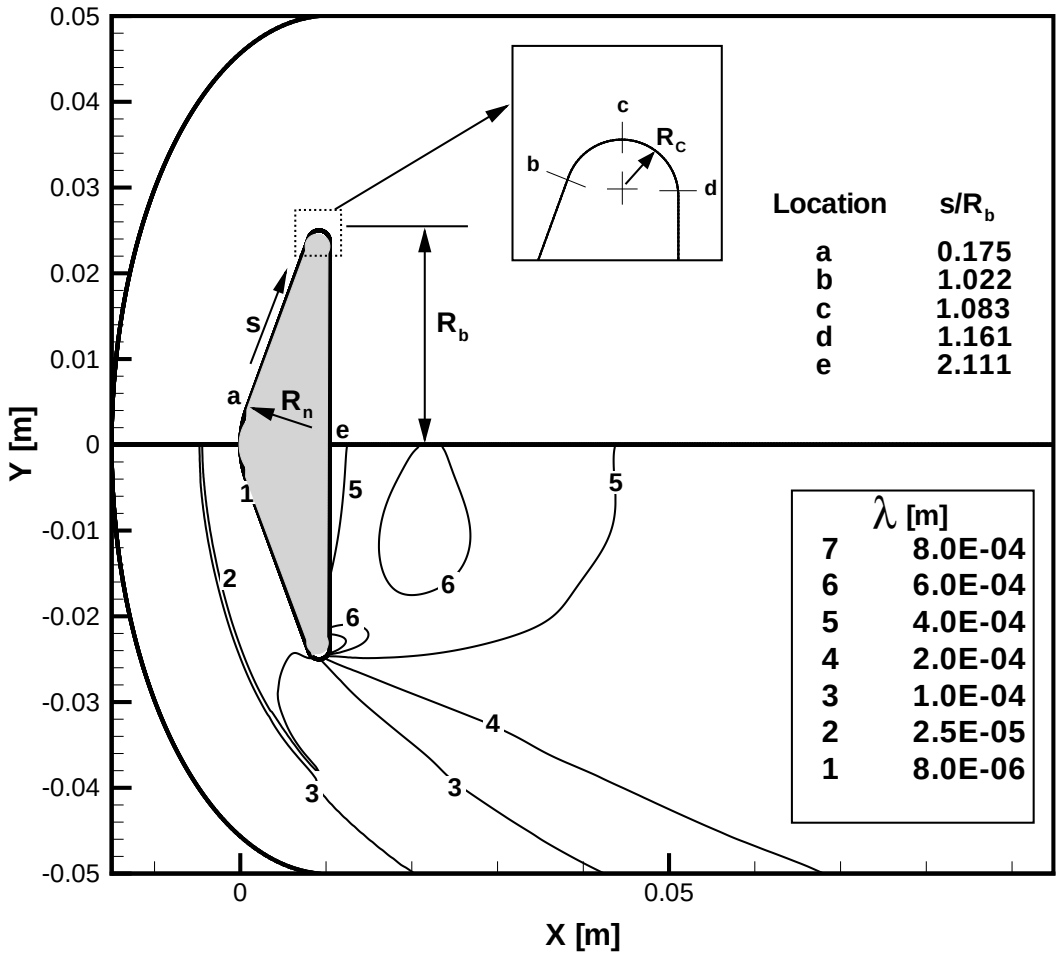


Figure 5.22: Planetary probe geometry (dimensions in mm) and variation of λ.

Table 5.1: Experimental flow conditions for the planetary probe problem.

Case	M_∞	u_∞ $[m/s]$	ρ_∞ $[kg/m^3]$	T_∞ $[K]$	Tw $[K]$	Kn_g^*
Planetary Probe - SR3 Case 3 -	20.5	1633.0	4.660×10^{-4}	15.3	300.0	0.001

$^*Kn_g$ is based on the diameter of the planetary probe, and the VHS value for λ

been applied to this same case where the dense fore-body flow is computed separately using a NS solver and the exit-plane solution is specified as inflow conditions for a decoupled DSMC simulation of the entire wake region. It should be noted that although the experiment supported the model with a sting, the zonally decoupled simulations did not include the sting. In order to investigate the size of the wake vortex and compare with the zonally decoupled results, the present study also does not include the sting. The low density (high Kn_g) cases are found to lie entirely in the rarefied (non-equilibrium) regime and can be solved efficiently using full DSMC simulation. In fact, application of the continuum breakdown parameter to these low density cases creates initial DSMC regions that cover the entire computational domain (with the exception of the free-stream flow before the bow shock). The current MPC simulation will focus on the highest density case whose initial conditions are listed in Table 5.1. As depicted in the bottom of Fig. 5.22, the hard-sphere value of λ varies by 2 orders of magnitude between the stagnation and near-wake regions. In a full DSMC simulation, since the local cell size (in each dimension!) must scale with λ, an enormous number of cells (and therefore simulation particles) are required to simulate the fore-body flow. In addition, the flow is axi-symmetric where regions of the computational domain include the axis of symmetry. Cells far away from the axis have orders-of-magnitude larger volumes than cells near the axis and as a result would require a prohibitive number of particles to model the same density. For this

reason, radial cell-weighting must be used for DSMC simulation. The effective weight W in each cell (defined as the number of real particles represented by each simulated particle) is increased by a factor of 4 for each factor of 4 increase in distance from the axis of symmetry. The simple algorithm used to set these radial weighting factors is detailed in Algorithm [17] Appendix A.

5.4.2 DSMC and NS simulation results

The computational mesh used to obtain a NS solution to the planetary probe problem consists of 76,500 quadrilateral cells. Cells are clustered towards the surface of the probe in order to ensure that a sufficient number of cells are located within the boundary layer. LeMANS solves the laminar, compressible, NS equations with the additional axi-symmetric source terms (detailed in Chapter II). A CFL value of 50 is used which results in constant NS timesteps of approximately 1.0×10^{-6} s. The computational mesh used to obtain the full DSMC solution consists of 1.6 million quadrilateral cells. The majority of cells are required to accurately simulate the fore-body region where, as seen in Fig. 5.22, λ is very small. This fine DSMC mesh is actually generated by starting with the NS mesh and solution (just described above) and applying the mesh refinement procedure used by the MPC method. This procedure was detailed in Chapter III and in order to generate a full DSMC mesh, the refinement is performed throughout the entire computational domain (instead of only in regions of continuum breakdown). The local cell size is verified to be less than the local value of λ in all regions of the flow. At the fore-body stagnation point the cell size is 0.9λ normal to the surface and approximately 3λ parallel to the surface. The MONACO code is used to perform a DSMC simulation of the planetary probe problem on this fine mesh. MONACO employs a constant timestep

in all flow regions for this study, which is verified to be less than 60% of the smallest mean-free-time found in the flow field. The mean-free-time is very small near the fore-body stagnation region, which significantly limits the timestep used by the DSMC method. The timestep used for DSMC simulation is 6.5×10^{-9} s, approximately 150 times smaller than the timestep used to solve the NS equations. In addition, the DSMC simulation requires a significant amount of iterations to reach steady-state. Portions of the large wake-vortex near the base of the planetary probe are highly rarefied. As a result, it takes many iterations (approximately 100,000) for simulation particles to fill this region and for the total number of simulation particles to reach a steady-state. After reaching steady-state, the solution is then sampled for another 100,000 iterations in order to reduce the statistical scatter in the surface properties (heat flux) to an acceptable level. The planetary probe problem is a computationally expensive simulation for the DSMC method and in this study the full NS simulation was obtained in less than 1% of the time required to obtain the full DSMC solution.

The translational temperature fields calculated using both MONACO (DSMC) and LeMANS (CFD) are shown in Fig. 5.23. Here both DSMC and CFD methods predict the same fore-body flow and only differ near the shoulder and in the near-wake region. Streamlines originating from identical points are plotted in Fig. 5.23 for both DSMC and CFD solutions, which show that the NS equations predict a larger vortex in the wake than DSMC. In addition, streamlines are seen to lie closer to the surface in the DSMC solution, signifying velocity slip in the shoulder and base regions of the planetary probe. A close look at the fore-body flow reveals a high level of agreement between NS and DSMC solutions. The temperature and density profiles along the stagnation line are plotted in Fig. 5.24(a). The sharp jump in temperature and density across the bow shock wave is seen to be followed by a large

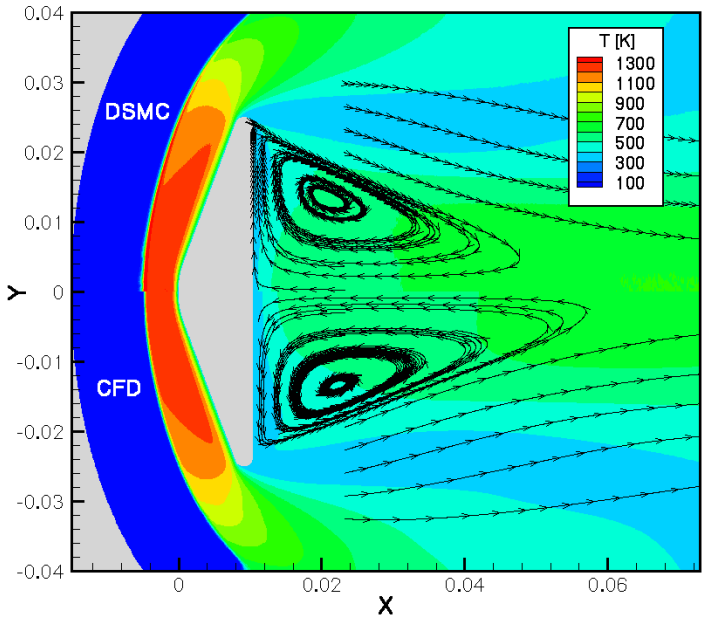
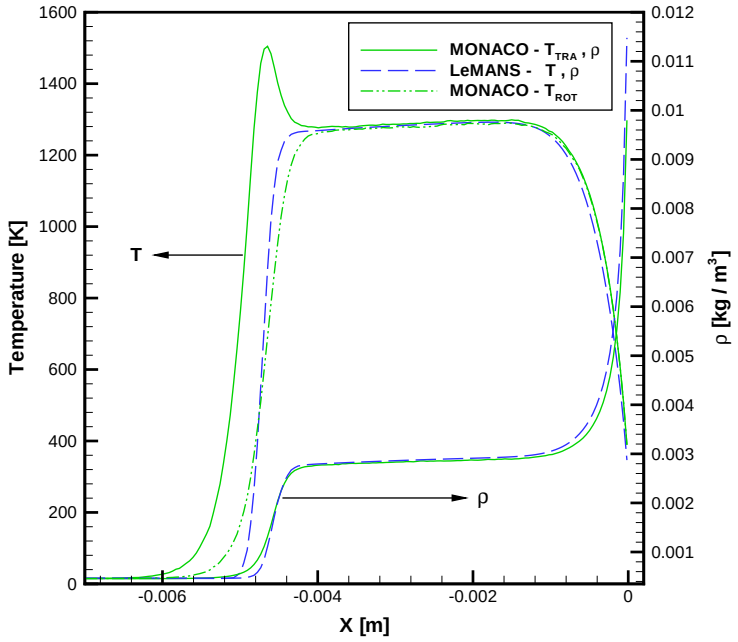
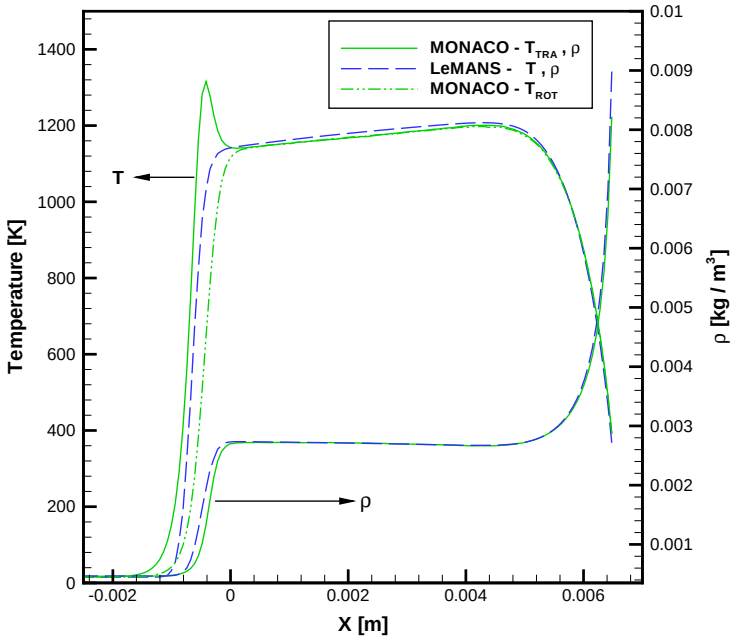


Figure 5.23: Particle and continuum simulation of the planetary probe configuration.

constant region of highly compressed, high temperature flow before the flow rapidly cools due to the cold wall of the probe. Except for the interior of the shock wave, both DSMC and NS solutions are seen to be in excellent agreement. The DSMC profiles show that the gas reaches thermal equilibrium ($T_{TRA} = T_{ROT}$) very quickly after the bow shock and remains in thermal equilibrium all the way to the probe surface. Thus the NS equations, which model the gas using a single translational-rotational temperature T, provide an accurate model along the stagnation line. It is important to notice that the temperature shock predicted by DSMC lies upstream of that predicted by the NS equations. This was also observed previously in Section 5.3 and is a well known prediction of the DSMC method. However, as expected, the density shock location (the shock stand-off distance) is seen to be identical for both cases. Figure 5.24(b) also plots the temperature and density profiles in the fore-body region, however further from the axis of symmetry. Specifically properties



(a) Temperature profiles along the stagnation line.



(b) Temperature profiles along cut $C1$ (see Fig. 5.25(a).)

Figure 5.24: DSMC and NS simulation results for the fore-body flow.

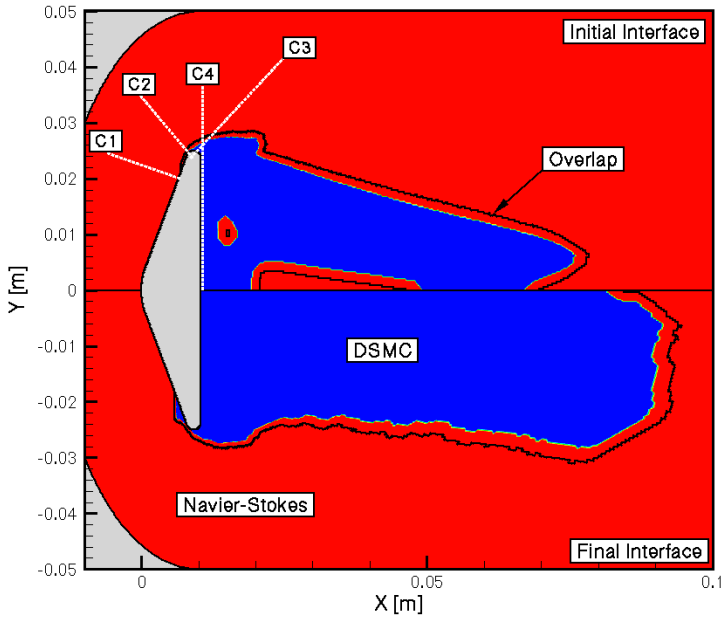
are extracted from a line normal to the surface originating at a point 2 cm from the axis on the surface of the probe (cut $C1$ in Fig. 5.25(a)). In this region of the fore-body, the trends and agreement between DSMC and NS solutions are identical to that observed along the stagnation line. This demonstrates that the entire fore-body region of the planetary probe configuration is accurately modelled by the NS equations. As previously observed in Section 5.3 the post-shock state predicted by both DSMC and NS methods is identical. Thus, if accurate prediction of the shock interior is not required, the shock wave may also be simulated using the NS equations. This would eliminate a significant number of particles from an MPC simulation and as determined in Section 5.3, will not effect the accuracy of predicted surface properties. For the MPC simulation of the planetary probe presented in this section, the interior of the bow shock wave is modelled using the NS equations.

5.4.3 Progression of the MPC solution

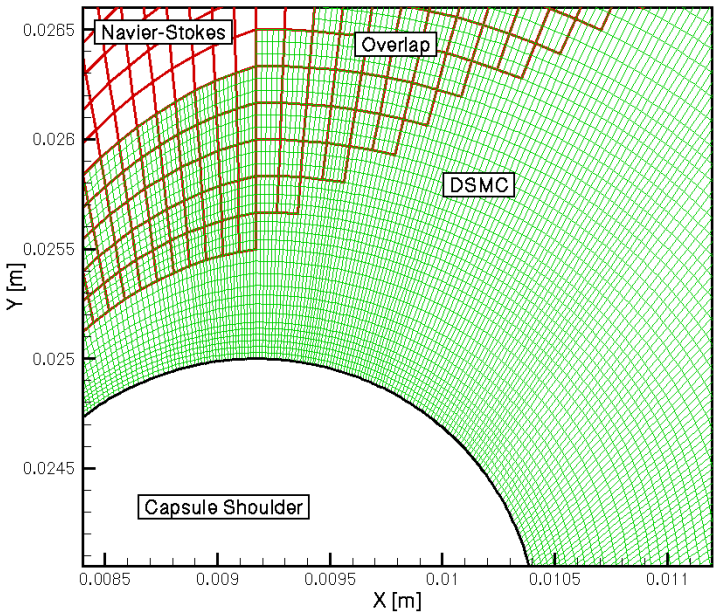
MPC simulation of the planetary probe begins with the initial NS solution obtained on the coarse continuum mesh described above. Application of the continuum breakdown parameter using a cutoff value of $Kn_{GL} = 0.05$, results in the initial particle region shown in the top of Fig. 5.25(a). The results presented in this section include flow properties extracted along cuts $C1$ to $C4$ for full DSMC, full NS, and MPC simulations. The precise location of cut $C1$ was outlined in the previous paragraph. Both cuts $C2$ and $C3$ extend normal to the capsule-shoulder surface and lie at a 45 degree angle from the axis of symmetry. Finally, cut $C4$ lies parallel to the base of the planetary probe, 0.15 mm off the surface (cutting through the center of the first layer of surface cells). After the initial DSMC regions are created, the MPC method carries out the mesh refinement procedure. A portion of the hybrid

mesh near the capsule shoulder is shown in Fig. 5.25(b). Since λ is very small in the fore-body region, significant mesh refinement is required. Since the same mesh refinement procedure was used to generate the mesh for the full DSMC simulation, Fig. 5.25(b) gives an indication of the mesh density required throughout the entire fore-body flow by the DSMC method. Initially particles are generated in all DSMC cells (using a Chapman-Enskog velocity distribution [56]) such that they are consistent with the NS solution. As the DSMC regions iterate and particles collide with each other and with surfaces, the DSMC solution proceeds away from the initial continuum NS solution towards the correct non-equilibrium solution. As this progression and relaxation occurs, DSMC and NS regions are loosely coupled and interface locations are periodically updated throughout the simulation according to the numerical cycle detailed in Section 5.2.3. As an example, the final steady-state interface location for the planetary probe problem is shown in the bottom of Fig. 5.25(a).

The progression and relaxation process within the DSMC region of the MPC simulation along cut $C3$ is plotted in Fig. 5.26. This process occurs during step 2 of the MPC numerical cycle outlined in Section 5.2.3. The vertical velocity component and the density are plotted at three different times (t_0, t_1, and t_2) during the MPC simulation in the top and bottom of Fig. 5.26 respectively. Within this highly non-equilibrium region of rapidly expanding flow, DSMC simulation (without coupling) allows the correct velocity and density profiles to quickly develop within the particle regions of the MPC simulation. These results demonstrate again how the overlap region provides a buffer that allows DSMC and NS solutions to differ. The more accurate DSMC information is then used to update the NS boundary conditions. This is evident in the bottom of Fig. 5.26 where although the DSMC density at



(a) Initial and final interface locations.



(b) Mesh refinement near the shoulder region.

Figure 5.25: Interface locations and mesh refinement for the planetary probe.

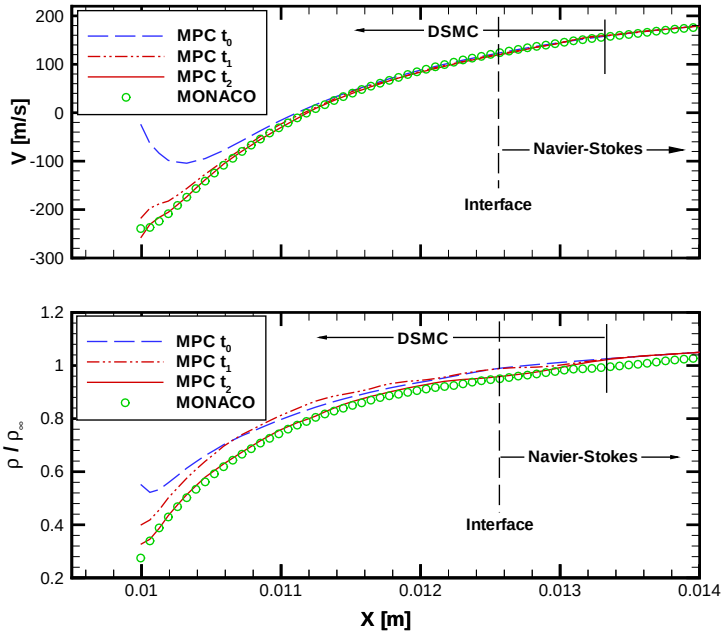


Figure 5.26: Progression of the hybrid solution for the planetary probe problem.

the edge of the overlap region is set by the NS solution, inside the overlap region the DSMC density differs from the NS result. When the DSMC solution is used to set new NS boundary conditions, the NS density profile will be shifted into better agreement with the full DSMC simulation (a lower density). The shifted NS solution will then provide more accurate boundary conditions for the DSMC region and the above MPC cycle will repeat itself until step 4 is reached.

5.4.4 Flow field results

The translational temperature profiles along cut $C2$, approaching the capsule shoulder in the fore-body flow, are plotted in Fig. 5.27. At this location in the flow field, the gas is beginning to expand around the capsule shoulder. As seen by the final particle-continuum interface locations in Fig. 5.25(a) (bottom), the NS equations are used to model the flow along this cut except for a thin region next to

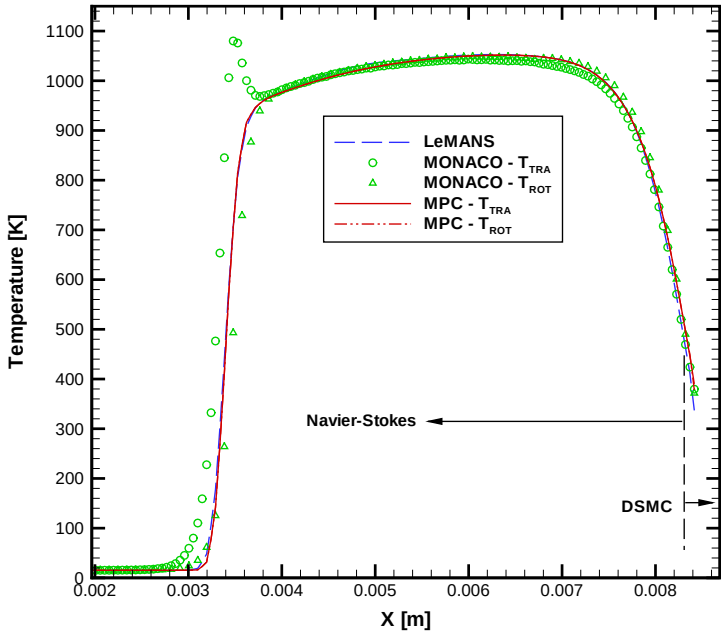
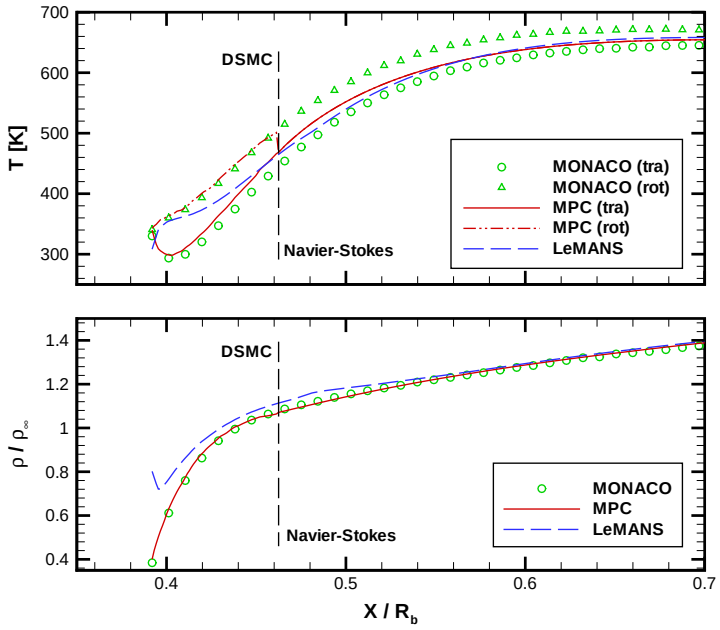


Figure 5.27: Fore-body temperature profile approaching the capsule shoulder.

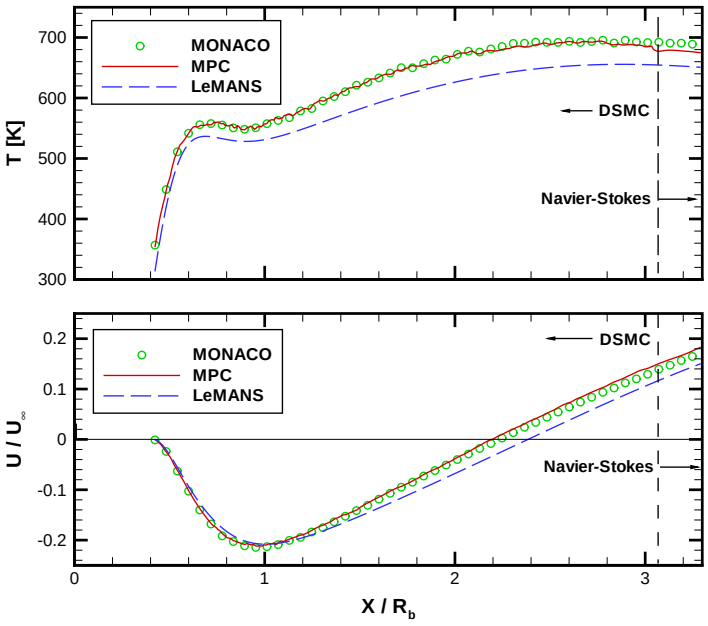
the surface. The rapid expansion about the capsule shoulder induces rarefaction that is detected by the continuum breakdown parameter which sets up the thin DSMC region next to the surface shown in Fig. 5.27. Except inside the shock wave and right next to the surface, DSMC, NS, and MPC simulations are in excellent agreement. A small temperature jump at the probe surface is predicted by DSMC which the MPC method successfully reproduces despite only using a thin layer of DSMC cells next to the surface.

The flow field properties extracted from cut $C3$, extending normal to the shoulder surface into the wake region, are displayed in Fig. 5.28(a). Both translational and rotational temperatures are plotted in the top of Fig. 5.28(a). DSMC predicts the gas to be in thermal equilibrium at the wall at a slightly higher temperature (a temperature jump of 7%) compared with the NS solution. It should be noted that a full thermal accommodation boundary condition is used in DSMC, which

assumes that particles colliding with the surface leave with both translational and rotational energy modes thermalized to the wall temperature. As seen in the top of Fig. 5.28(a), close to the surface the MPC method uses DSMC which individually models translational and rotational energy modes, however on the other side of the interface the MPC method solves the NS equations, which assume both modes are in equilibrium and are described by a single temperature. It is quite remarkable that despite particles being introduced to the DSMC region at the interface with incorrect translational and rotational energies, the hybrid-DSMC solution quickly recovers and reproduces full DSMC results accurately near the wall. This behavior has been observed in each MPC simulation studied in this thesis and demonstrates the fundamental mechanism that drives the MPC solution towards the correct non-equilibrium result. Of course the MPC method is unable to predict the thermal non-equilibrium that persists in the expansion region behind the probe where the hybrid-NS region assumes a single temperature. From a practical perspective however, accurate prediction of surface properties as well as the flow near the surface and within the vortex are more important than predicting this small degree of thermal non-equilibrium away from the probe surface. The density along cut $C3$ is plotted in the bottom of Fig. 5.28(a). At the surface, the density predicted by DSMC is 50% of that predicted by the NS equations. Again, the MPC method is seen to accurately reproduce full DSMC results, provide more accurate boundary conditions for the hybrid-NS region, and shift the NS portion of the MPC solution into better agreement with full DSMC results. The translational temperature and normalized x-velocity (u/u_∞) directly in the wake, along the axis of symmetry, are plotted for each method in the top and bottom of Fig. 5.28(b) respectively. DSMC is seen to predict a 17% temperature jump at the base of the probe and $10-13\%$ higher



(a) Wake temperature profile near the planetary probe shoulder.



(b) Flow properties in the planetary probe wake.

Figure 5.28: Flow field results in the wake of the planetary probe.

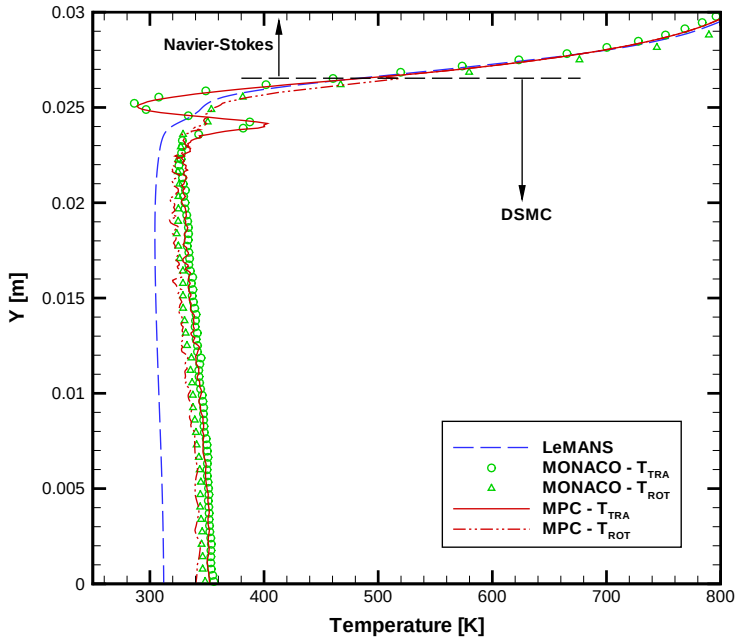


Figure 5.29: Temperature profile along the base of the planetary probe.

temperatures in the wake compared with that predicted by the NS equations. In the bottom of Fig. 5.28(b), DSMC is seen to predict the vortex to close 10% earlier than the NS simulation. The MPC method captures these flow field features very accurately. In the process, the variations in DSMC regions provide more accurate boundary conditions and succeed in shifting the NS portion of the solution into better agreement with full DSMC results.

Finally, the temperature profiles along the base of the planetary probe (cut $C4$) are plotted in Fig. 5.29. Here, due to rapid expansion of the gas, the temperature is seen to be significantly lower directly behind the probe ($Y < 0.025\ m$) than above it. Above the probe, the MPC method uses the NS equations to model the flow and the results from full DSMC, full NS, and MPC simulations are in close agreement. As seen in the upper portion of Fig. 5.29, in the rapid expansion region near the capsule shoulder, DSMC predicts first a rapid decrease in translational temperature followed

by a rapid increase. This decrease is due to expansion of the gas about the shoulder. The increase is due to a weak separation shockwave generated where the expanding gas hits the large wake-vortex. The rotational energy mode of the gas relaxes much slower than the translational mode. As a result, the rotational temperature profile predicted by DSMC does not follow such sharp flow features. Since the MPC method uses DSMC in this highly non-equilibrium region, as seen in Fig. 5.29, it is able to reproduce full DSMC results for both translational and rotational temperature with a high level of accuracy. Due to the no-slip boundary conditions employed by the NS solver, the NS temperature profile along the base of the probe corresponds to the wall temperature of 300 K. Since cut $C4$ is taken slightly off the surface, the NS temperature is seen to be slightly higher than 300 K. Full DSMC simulation predicts the gas to be close to thermal equilibrium with a temperature that is 40-60 K higher than the wall temperature. The results of the full DSMC simulation show that along the base of the probe the translational temperature is slightly higher than the rotational temperature. The MPC method reproduces these full DSMC results very accurately.

5.4.5 Surface property results

The final steady-state particle region for the planetary probe MPC simulation was shown previously in Fig. 5.25(a) and includes the shoulder region and a large portion of the wake. The heat transfer results for full DSMC, NS, and MPC simulations are displayed in Fig. 5.30 and compared with experimental measurements. Since the experiment was sting-supported and the numerical simulations were performed with no sting, experimental results are only plotted for $s/R_n < 3.0$ where the presence of a sting has little impact on the heating rate. Comparing full DSMC and NS results

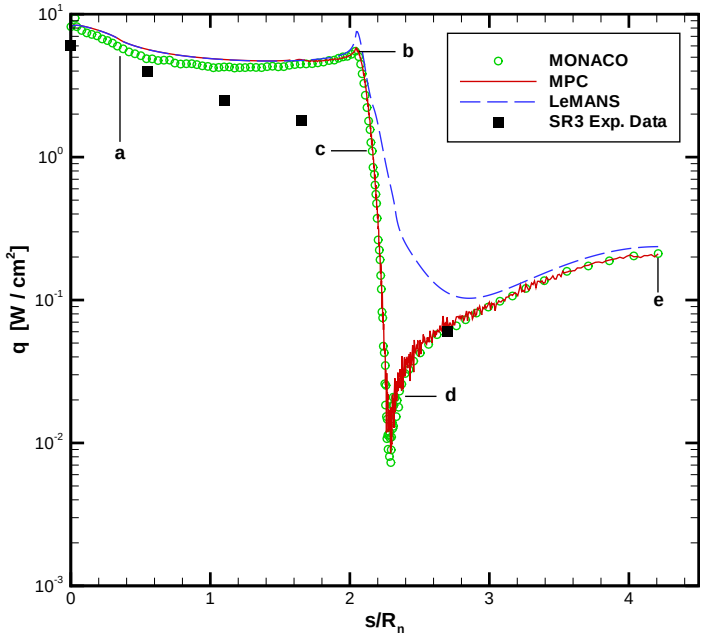


Figure 5.30: Experimental heating rates and numerical predictions.

in Fig. 5.30, it can be seen that both predict the same peak heating rate at the stagnation point. Along most of the fore-body (locations a-b), DSMC predicts a slightly lower heating rate than the NS equations, however both simulations predict heating rates ranging from 2-3 times larger than measured experimentally. While the reason for this remains unclear, both DSMC and NS results agree very well with simulations performed by other researchers [2]. Around the capsule shoulder and along the capsule base (locations b-e), DSMC is seen to predict a much lower heating rate than the NS equations, and DSMC is in better agreement with the experimental result. In Fig. 5.30, the MPC simulation is shown to reproduce DSMC results very accurately. In the highly compressed fore-body region where DSMC and NS simulations produce similar results, the MPC method successfully solves the NS equations. This is much more efficient than using the DSMC method in the fore-body and produces the same result for heat transfer. Just prior to the shoulder

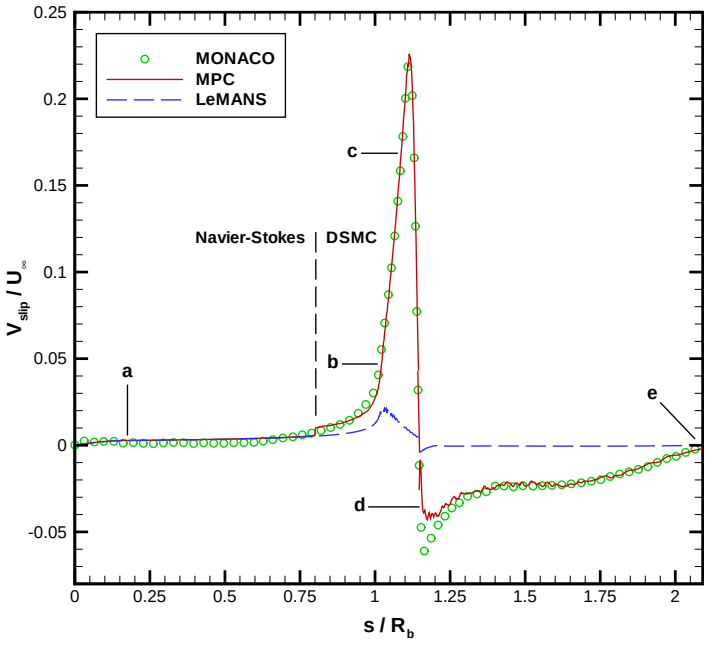


Figure 5.31: Velocity slip along the planetary probe surface.

($s/R_n = 1.6$), where the MPC method switches to DSMC, we see the heating rate transition from the NS result to the DSMC result. Thereafter, for the entire shoulder and base region, the MPC method is seen to reproduce full DSMC results with a high degree of accuracy.

The velocity magnitude around the surface of the planetary probe (just 10 μm off the surface) is extracted and plotted in Fig. 5.31 after being normalized by the free-stream velocity. The extraction is performed using Tecplot© where a curve is created parallel to the probe surface at a distance 10 μm normal to the surface. Data from the flow field is then extracted along this curve which involves Tecplot© linearly interpolating data from the mesh nodes to the curve. In the fore-body region both full DSMC and NS simulations predict virtually no slip velocity at the probe surface. However, DSMC shows very large velocity slip near the shoulder region that persists throughout the entire base region ($s/R_b > 0.8$), whereas the NS simulation

assumes no slip conditions around the entire surface. It should be noted that since the velocity magnitude is extracted slightly off the surface that even the NS result will show a velocity magnitude greater than zero in some regions. Furthermore, the results in Fig. 5.31 agree both qualitatively and quantitatively with those of other researchers [2], accounting for the fact that the exact distance from the surface directly influences the precise magnitude of the velocity. Finally, the MPC method is seen to accurately reproduce full DSMC results along the entire planetary probe surface while successfully solving the NS equations in the fore-body region where both DSMC and NS predict no velocity slip. Both MPC results plotted in Figs. 5.30 and 5.31 lend further confidence in the use of the gradient-length Knudsen number (Kn_{GL}) and a value of $Br_{\text{cutoff}} = 0.05$ for predicting continuum breakdown in blunt-body flows.

5.5 Computational Performance of the MPC Method

5.5.1 Comparing DSMC and MPC simulations

The goal of the modular particle-continuum numerical method is to accurately reproduce the results of full DSMC simulations in less time using less memory. Since the main advantages of the MPC method come from spatial and temporal scale decoupling (different sized cells and timesteps), it is not meaningful to make efficiency comparisons per cell or per iteration. Instead, the most meaningful and useful comparisons involve the total time required by each method to generate an accurate solution. Both DSMC and MPC simulations have two distinct stages: the time required for the flow to reach a *steady-state*, followed by the time required to *sample* molecular information to obtain low-scatter macroscopic flowfield and surface properties. For DSMC simulations, steady-state is reached when the particles have filled

the entire computational domain and all flow features have developed to steady-state. Steady-state in a DSMC simulation can be determined when the total number of simulation particles levels off and becomes constant. For the MPC simulations, the time required to reach steady-state involves the initial NS simulation plus the hybrid coupling procedure, which is complete when both DSMC and NS solutions stop changing and the interfaces stop moving. Thus an MPC simulation is determined to have reached steady state when it reaches step 4 of the numerical cycle (refer to Section 5.2.3). A fair comparison then requires both DSMC and MPC simulations to be sampled for the same number of DSMC timesteps. Sampling the MPC solution involves fewer particles than for full DSMC, but also involves occasional coupling to the NS solver and NS iterations of the continuum regions. The total simulation time for both DSMC and MPC methods is then the sum of these two stages. The *speedup factor* for each case is defined as the time required for full DSMC simulation divided by the time required for MPC simulation. The speedup factors achieved for the 2D cylinder flows with and without mesh refinement are plotted as solid symbols in Fig. 5.32 and are seen to range from 1.4 to 3.3.

5.5.2 Ideal speedup factor

There are certainly many factors and numerical parameters that influence the efficiency of a given DSMC or NS code for a given flow problem. However, the underlying principle of the MPC method is that by simulating continuum regions using far fewer cells for far fewer timesteps, the time spent simulating these continuum regions should be negligible compared with the time required to simulate the same region using a particle method. The time required for a DSMC simulation scales nearly linearly with the total number of simulated particles. Thus if the MPC

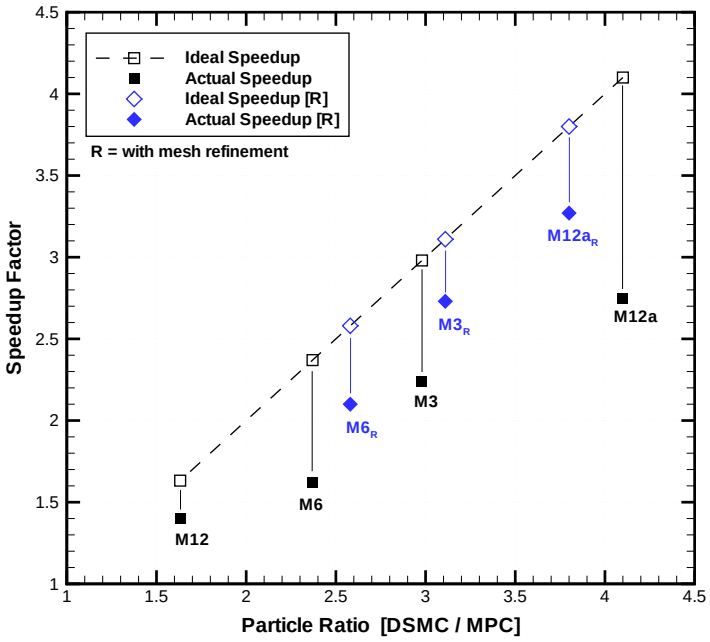


Figure 5.32: Computational efficiency of the MPC method.

method was able to use 4 times fewer particles and all hybrid NS operations were free, ideally it could reach the answer 4 times faster. The ideal speedup factor is thus defined as the ratio of particles in a full DSMC simulation to that required by the MPC method. The ideal speedup factor is shown as the dashed line in Fig. 5.32 and the hollow symbols represent the ideal speedup factor (equal to the particle ratio) for each case. If NS operations and updates are indeed negligible, then even significant variations in the efficiency of the NS solver would have a minimal effect on the speedup factor. Also, since the MPC method uses existing DSMC and NS codes as modules, any acceleration techniques added to individual DSMC and NS methods, will also be used with the hybrid simulation. Thus although absolute simulation times may certainly vary, the speedup factors will still be an overwhelming function of the number of particles eliminated (replaced by a continuum description) by the MPC method.

5.5.3 MPC overhead

Of course the time required for the initial NS solution as well as the time required for NS updates within the MPC simulation is not entirely negligible. As seen in Fig. 5.32, MPC simulations of the 2D cylinder flows achieve actual speedup factors that are lower than the ideal speedup factors for each case. The difference between actual and ideal speedup factors (normalized by the ideal speedup factor) is termed the *MPC overhead*. This overhead is depicted by the vertical lines in Fig. 5.32 and is listed in Table 5.2 for each case. In addition to NS operations, the time required to generate particles in DSMC boundary cells, compute the sub-relaxation averages, and apply the continuum breakdown parameter also contributes to the overhead. A final source of overhead may come from the loose coupling between DSMC and NS regions. Although fewer particle-cells are required in an MPC simulation, due to the alternating simulation of DSMC and NS regions, more iterations of these particle-cells may be required to reach steady-state. For example, the numerical cycle employed by Wu *et al.* [40] simulates DSMC regions to steady-state before updating the interface locations. Therefore DSMC regions are simulated to steady-state as many times as the interface locations are updated. This may be the reason that they report their hybrid simulations as taking longer than full DSMC simulations. As detailed in Section 5.2.3, the MPC method utilizes the sub-relaxation average to update interface locations *while* DSMC regions are proceeding towards steady-state. This is a very important aspect of the MPC numerical cycle. As a general result, an MPC simulation typically requires a similar number of iterations to reach steady-state as required by a full DSMC simulation. Thus the major source of MPC overhead comes from the time required for the initial NS solution as well as for NS updates during the MPC cycle.

Table 5.2: Computational time and memory requirements of the MPC method.

case	actual speedup	ideal speedup	MPC overhead	memory usage
M12	1.40	1.65	15.2%	158%
M6	1.62	2.37	31.6%	137%
$\mathbf{M6_R}$	2.10	2.58	18.6%	69%
M3	2.24	2.98	24.8%	134%
$\mathbf{M3_R}$	2.73	3.11	12.2%	51%
M12a	2.75	4.10	33.0%	97%
$\mathbf{M12a_R}$	3.27	3.80	14.0%	32%

5.5.4 MPC efficiency for 2D cylinder flows

The actual speedup factors for each case are shown in Fig. 5.32 as the solid symbols and are also listed in Table 5.2. They are seen to be proportional to the number of particles saved by the MPC method (the particle ratio), which is proportional to the amount of the flow field which can be assumed as near-equilibrium flow. For cases **M3**, **M6**, and **M12**, the MPC method uses 3.0, 2.4, and 1.65 times fewer particles and reaches the solution 2.2, 1.6, and 1.4 times faster, respectively, than full DSMC simulations. Analysis of the additional **M12a** case reveals that by not modelling the shock wave as a particle region, a large number of particles are eliminated because the shock is a dense region of the flow. The speedup achieved for the Mach 12 flow conditions is determined to increase from 1.4 to 2.8 times and, as shown earlier, aside from the internal shock structure, the MPC results for case **M12a** accurately reproduce full DSMC flowfield and surface properties. It should be noted that the initial NS solution is determined to account for roughly 50% of the MPC overhead for all cases that do not utilize mesh refinement. This is due to the fact that the NS solver uses the same mesh as DSMC, which is refined to the mean-free-path. In addition,

NS iterations on this fine mesh during the MPC coupling cycle also contribute to the overhead. Thus, by allowing the NS solver to operate on a coarser mesh, a large portion of the MPC overhead should be eliminated.

The addition of mesh refinement to the MPC method reduces its memory requirements and enables the NS module to operate efficiently on a coarser mesh. The decrease in both overhead and memory requirements of the MPC method achieved by mesh refinement are shown in Fig. 5.32 and detailed in Table 5.2. The added use of mesh refinement for cases $\mathbf{M3_R}$, $\mathbf{M6_R}$, and $\mathbf{M12a_R}$, is verified not to affect the level of accuracy achieved by the MPC method without mesh refinement (cases **M3**, **M6**, and **M12a**). Since the final hybrid mesh is different when mesh refinement is utilized, the final interface locations are not precisely the same as without mesh refinement. In general, cases with and without mesh refinement then have a slightly different total number of simulation particles, which results in the slightly different particle ratios seen in Fig. 5.32 and in Table 5.2. The memory usage is the ratio of memory used by an MPC simulation to that required by a full DSMC simulation. Table 5.2 shows that added mesh refinement has cut the MPC overhead roughly in half for each case. This is entirely due to the decrease in time required for both the initial NS solution and for NS updates of the continuum regions during the MPC numerical cycle. For the flow conditions used in cases $\mathbf{M3_R}$, $\mathbf{M6_R}$, and $\mathbf{M12a_R}$, the speedup factors achieved with mesh refinement are now 2.73, 2.10, and 3.27 respectively. Since previous results without mesh refinement stored the same mesh (refined to the mean-free-path) in both NS and DSMC modules, as Table 5.2 shows, the MPC simulations actually required more memory than full DSMC simulations despite using fewer particles. The addition of mesh refinement has substantially lowered the memory requirements for the NS module. As a result, MPC simulations

now require less memory than full DSMC simulations. In general, as the number of particles increases in an MPC simulation, the overhead is expected to diminish (especially that associated with the initial NS solution and NS updates).

5.5.5 MPC efficiency for the planetary probe flow

The planetary probe problem is very well suited for a hybrid method. Due to the large variation in λ, mesh refinement within the MPC method is able to eliminate many computational cells and many particles. The MPC simulation uses 7.1 times fewer particles and implicit NS timesteps are 150 times larger than the timestep used by DSMC in the fore-body region. As a result, hybrid and NS operations within the MPC cycle are indeed negligible compared with the time spent simulating DSMC regions. The MPC solution is obtained approximately 12.5 times faster and uses only 20% of the memory compared with full DSMC simulation. For this case, full DSMC simulation requires a long time for particles to fill the wake region and reach steady-state. Since the MPC method begins with a NS solution, particles generated in the wake region are much closer to steady-state immediately. As a result, the time required for the MPC method to reach steady-state and begin sampling is much less than the corresponding time required by full DSMC simulation. This results in a larger actual speedup (12.5x) than would be achieved by the elimination of particles alone (7.1x).

Table 5.3: Computational efficiency of the MPC method for the planetary probe.

	speedup	memory	particles	initial NS	steady-state	sampling
DSMC	100%	100%	100%	N/A	100%	100%
MPC	8% (12.5x)	20%	14% (7.1x)	.75%	30%	100%

5.6 Conclusions Drawn from Blunt-Body Simulations

The MPC algorithm is demonstrated to be capable of simulating 2D and axi-symmetric flows. The final MPC numerical cycle is developed within this chapter through application to hypersonic blunt-body flow problems. In summary, six major conclusions are drawn.

1) For hypersonic blunt-body flows, the gradient-length Knudsen number accurately predicts regions of continuum breakdown where a continuum NS solution begins to deviate significantly from DSMC results. The parameter is also shown to predict the magnitude of this discrepancy quite well. Thus using the Kn_{GL} parameter with a cutoff of 0.05 within the MPC method performs very well for blunt-body flows.

2) When initialized to be consistent with a continuum NS solution, DSMC simulation in particle regions progresses the hybrid solution towards the correct non-equilibrium result. It is important that particle-continuum interfaces are located within regions of near equilibrium flow before any information is transferred to continuum regions. A loosely-coupled, modular particle-continuum numerical cycle based on these conclusions is completely detailed. The numerical cycle ensures that the hybrid solution progresses towards the correct physical result in a robust and predictable manner.

3) The modular particle-continuum method applied to 2D baseline cylinder flows reproduces full DSMC results with a high level of accuracy. Excellent results are obtained for flow field and surface properties, as well as for local velocity distribution functions. Application to a more complex axi-symmetric flow of a planetary probe geometry also produce excellent results compared with full DSMC simulation.

4) Analysis of the local velocity distributions reveal that there are localized regions of non-equilibrium flow as well as large regions of continuum flow within typical blunt-body problems. Non-equilibrium flow produces macroscopic effects such as slip flow and temperature jump near surfaces, thermal non-equilibrium, variations in wake-vortex size, and higher temperatures within the wake-vortex. The heat and momentum transfer predicted by DSMC and the MPC method are generally lower than that predicted by the NS equations. In addition, accurate computation of the interior bow shock profile is not necessary for adequate prediction of surface properties.

5) Mesh refinement and implicit time-marching within the MPC method allow for complete spatial and temporal scale decoupling. Simulating fewer computational cells for fewer timesteps (than required by the DSMC method) allows for very efficient simulation of continuum regions. Comparison of simulation times with and without mesh refinement demonstrates the computational efficiency gained by spatial scale decoupling. For near continuum flows such as the planetary probe experiment, large efficiency gains are achieved by the MPC method compared with full DSMC simulation.

6) The MPC method using mesh refinement is shown to accurately reproduce full DSMC results in simulation times which are 2.7, 2.1, and 3.3 times faster for Mach 3, 6, and 12 cylinder flows respectively, where the global Knudsen number is 0.01. For a near continuum flow of the planetary probe, the MPC method reproduces full DSMC flowfield and surface property results, 12.5 times faster using 20% of the memory.

CHAPTER VI

Hypersonic Interaction Flows

The modular particle-continuum (MPC) method completely described in previous chapters is now applied to a hypersonic interaction flow. Hypersonic interaction flows involve the interactions between shock waves and boundary layers, producing complex and highly non-linear flow fields. Shock-boundary layer interactions occur in inlets of supersonic and hypersonic propulsion devices and in the vicinity of control surfaces on hypersonic vehicles. In such flows, shock-shock interactions often produce reflected shock waves that impinge on the surface of the vehicle. The interaction between a strong shock wave and a boundary layer often causes the flow to separate and form a region of high-pressure, recirculating gas next to the surface. High-speed flow hitting such a separation region significantly alters shock structures and interaction regions which, in turn affect the extent of flow separation. Peak aerothermal loads are observed at the location of shock impingement on the vehicle surface and therefore accurate prediction of this phenomenon is important in the design of a hypersonic vehicle. In addition, the heating and frictional loads measured within a separated region are drastically different than when the flow is purely attached. Such differences may reduce the effectiveness of a control surface and thus accurately predicting the extent of any flow separation is also a very important as-

pect in the design of hypersonic vehicles. For blunt-body flows, the strong bow shock does not interact with the vehicle surface. However, various blunt-body deceleration techniques that use supersonic parachutes or ballutes involve highly complex flow fields where many regions of shock-shock and shock-boundary layer interactions are present [10, 11].

In order to better understand hypersonic interaction flows, detailed experiments have been performed on two baseline geometries in particular. The first is an axi-symmetric double-cone geometry and the second is an axi-symmetric hollow cylinder-flare geometry. Both flows involve a shock-shock interaction producing a reflected shock that impinges on the boundary layer. This interaction results in a flow separation region, the extent of which depends on the free-stream flow conditions. The experiments were performed in the Large Energy National Shock (LENS) facility at the Calspan - University of Buffalo Research Center (CUBRC). The experimental results were made public after a blind code validation exercise organized by Holden at the 2001 AIAA Aerospace Sciences Meeting in Reno, NV. [64, 65] During the conference, invited numerical solutions using both CFD and DSMC methods were presented by various researchers including, Candler *et al.* [66], Gnoffo [67], Moss [68], Kato and Tannehill [69], Boyd and Wang [70], and Roy *et al.* [71].

This chapter focuses on one particular result of these studies. For the hollow cylinder-flare geometry under certain free-stream conditions, NS simulations over-predicted the size of flow separation. The free-stream conditions for this particular case were such that rarefied effects (such as slip flow and temperature jump) along the hollow cylinder were suspected. Indeed, the DSMC simulations accurately predicted the extent of flow separation measured by experiment. This chapter applies the MPC method to the hollow cylinder-flare geometry under these flow conditions

where DSMC simulation was seen to provide more accurate results than NS simulation. First, full DSMC and NS simulation results are presented for the hollow cylinder-flare configuration and are compared with experimental results. Next, MPC simulation results are presented and the progression of the MPC solution is analyzed. Flowfield quantities and surface properties predicted by each simulation method are compared with experimental results and the computational efficiency gained by the MPC method is discussed.

6.1 The Hollow Cylinder-Flare

6.1.1 Particle and Continuum Simulation

The hollow cylinder-flare geometry is detailed in the top of Fig. 6.1. The viscous interaction at the sharp leading-edge generates a boundary layer and weak shock wave that then interact with the stronger shock generated by boundary layer separation at the flare junction. This shock-shock interaction, which occurs near the surface of the flare after the junction, in-turn affects the size of the separation region. This strongly coupled and highly nonlinear flow provides a challenging case for the validation of both CFD and DSMC methods. The flow conditions investigated are those of Run 11 performed in the Large Energy National Shock (LENS) facility at the Calspan - University of Buffalo Research Center (CUBRC). Again, the experimental results were made public after a blind code validation exercise [64, 65]. For the Run 11 conditions, Navier-Stokes simulations [66] predicted a larger separation region than measured experimentally. The same simulations also highlighted the sensitivity of the solution to grid resolution, numerical flux function, and limiters used in the CFD code. Further investigation by Candler *et al.* [72] determined that the free-stream conditions had a significant amount of thermal non-equilibrium due to frozen vibra-

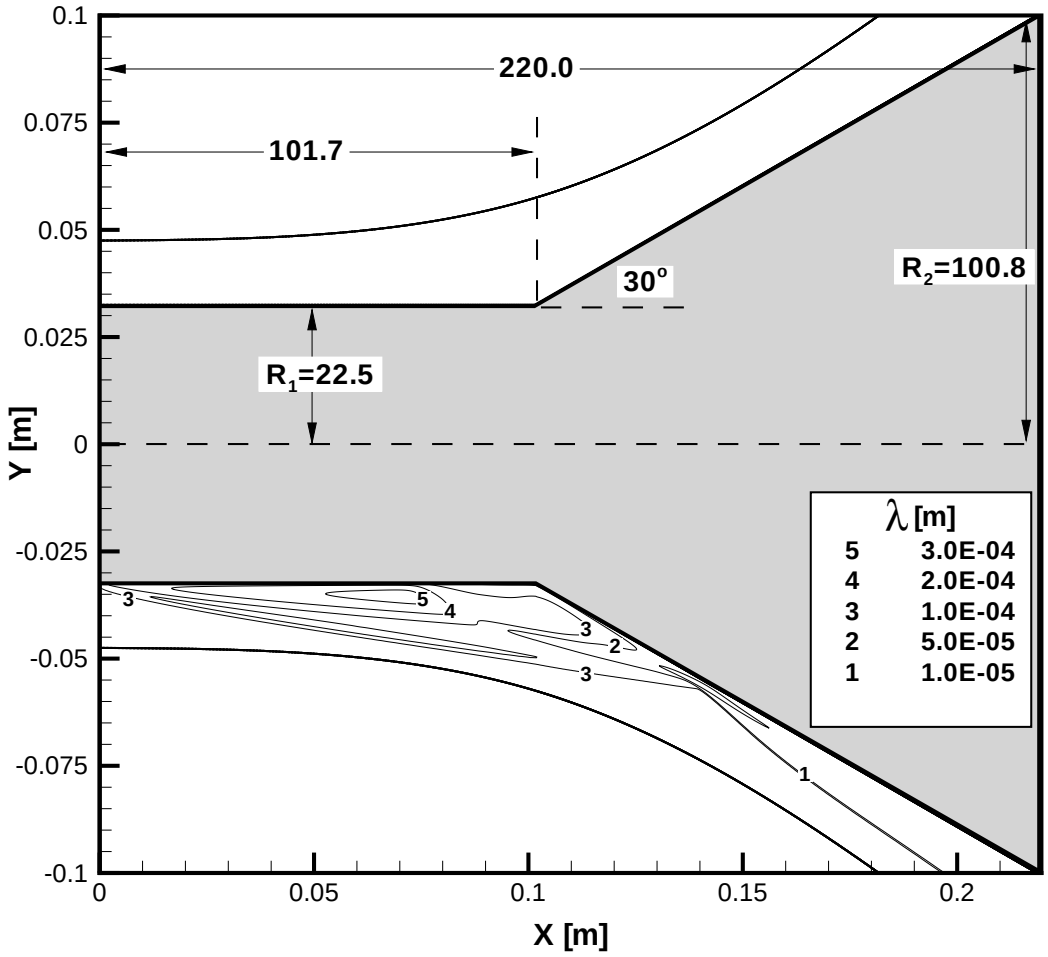


Figure 6.1: Hollow cylinder-flare geometry (dimensions in mm) and variation of λ.

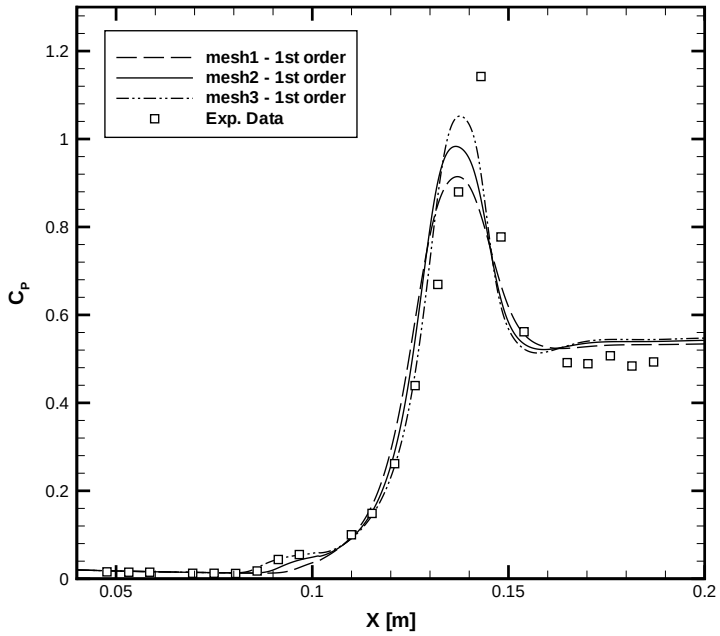
tional relaxation during the rapid nozzle expansion in the LENS facility. As a result, new revised free-stream conditions were defined for this case. The current study uses these revised flow conditions, which are listed in Table 6.1. It should be noted that the vibrational energy mode is assumed to be frozen throughout the flow and the vibrational temperature is therefore not used in any NS, DSMC, or hybrid simulations presented in this study. The LeMANS code as described in Chapter III is used to solve the NS equations for the hollow cylinder-flare geometry detailed in Fig. 6.1

Table 6.1: Experimental flow conditions for hollow cylinder-flare problem.

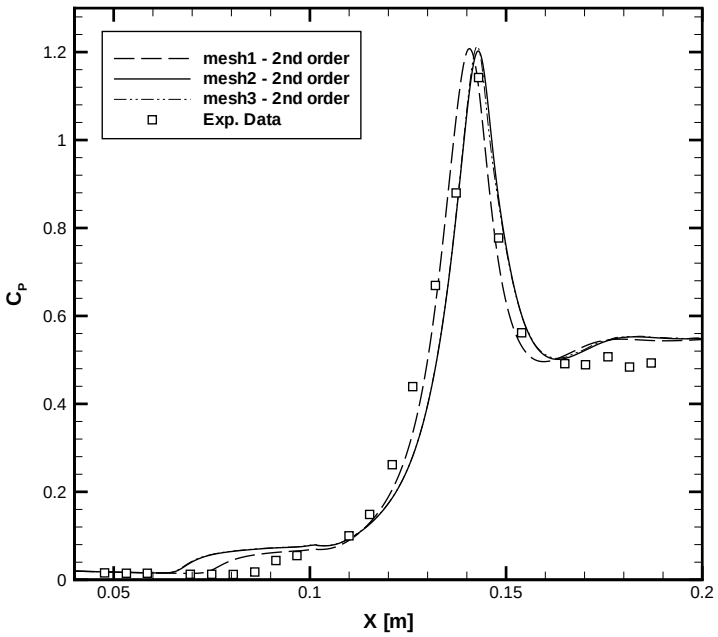
Case	M_∞	u_∞ $[m/s]$	ρ_∞ $[kg/m^3]$	T_∞ $[K]$	Tw $[K]$	Kn_g^*
Hollow Cyl. Flare - CUBRC Run 11 -	12.4	2484.1	5.566×10^{-4}	95.6	297.2	0.0008

$^*Kn_g$ is based on the length of the cylinder and the VHS value for λ

and for the flow conditions listed in Table 6.1. Solutions are obtained on three mesh densities of 550x125, 1100x250, and 2200x500 cells for mesh1, mesh2, and mesh3, respectively. Results for the coefficient of pressure $[C_P = (p - p_\infty)/(0.5)\rho_\infty u_\infty^2]$ are displayed in Fig. 6.2 in order to verify grid-independence and compare the NS solution with experimental results. In Fig. 6.2(a), first-order reconstruction of the solution is used by LeMANS. Here, the numerical results appear to predict the experimentally measured separation size well. However, closer inspection of the solution on all three mesh densities reveals that the size of the separation region becomes larger as the mesh refinement is increased. More importantly the variation in solution between each mesh is not seen to decrease, signifying that a grid-independent solution has not been obtained. Figure 6.2(b) demonstrates the importance of using second-order reconstruction when solving the NS equations for highly non-linear flows. Here, the predicted size of separation on even the coarsest mesh (mesh1) is seen to be much larger than that measured experimentally. The solutions obtained using both mesh2 and mesh3 are virtually identical, signifying that mesh2 has adequate resolution to provide a grid-independent solution to the problem. Thus for Run 11, the NS equations predict a separation region significantly larger than measured experimentally. As a result, the separation shock forms earlier which causes the strong interaction region, and thus the peak surface pressure, to occur further downstream than observed



(a) 1st-order Navier-Stokes results.



(b) 2nd-order Navier-Stokes results.

Figure 6.2: Navier-Stokes (LeMANS) grid convergence studies for the hollow cylinder-flare geometry.

experimentally. Although not shown, the NS equations (using no-slip boundary conditions) significantly over-predict surface heating and shear stress right at the leading edge of the hollow cylinder. The surface pressure distribution predicted by LeMANS in Fig. 6.2(b) is in excellent agreement with simulations performed by Candler *et al.* [72] with respect to the precise location of flow separation, the peak surface pressure, and the over-prediction of surface pressure along the flare.

In the blind code validation exercise, the DSMC method predicted leading edge surface properties and the size of the separation region well [65]. The largest discrepancy with experimental results was found downstream of the flare junction where λ becomes small. This can be seen in the bottom of Fig. 6.1 where the hard-sphere value of λ is seen to decrease by an order of magnitude after the flare junction. For this reason, fine resolution (cell size and timestep) is required for the flare region in a DSMC simulation. Subsequent DSMC simulations [73] have improved the resolution and produced excellent results compared with the experimental data. The MONACO code as described above is used to calculate the flow field and surface properties for the Run 11 flow conditions listed in Table 6.1 using both mesh2 and mesh3. The least resolved region in mesh2 and mesh3 is at the surface along the flare where cell dimensions are 4λ and 2λ respectively, in the direction normal to the wall. The mesh3 simulation uses twice as many simulated particles as the mesh2 simulation and both contain at least 15 particles per cell. The flow is axi-symmetric where cell-volumes in DSMC become larger as distance from the axis increases and therefore contain more particles. However, since the entire computational domain is located away from the axis of symmetry, the effect is minimal and no cell-weighting is used for DSMC simulation of the hollow cylinder-flare. Constant DSMC timesteps of 5×10^{-9} s are used, which are verified to be less than $0.6 \times \tau_c$ everywhere.

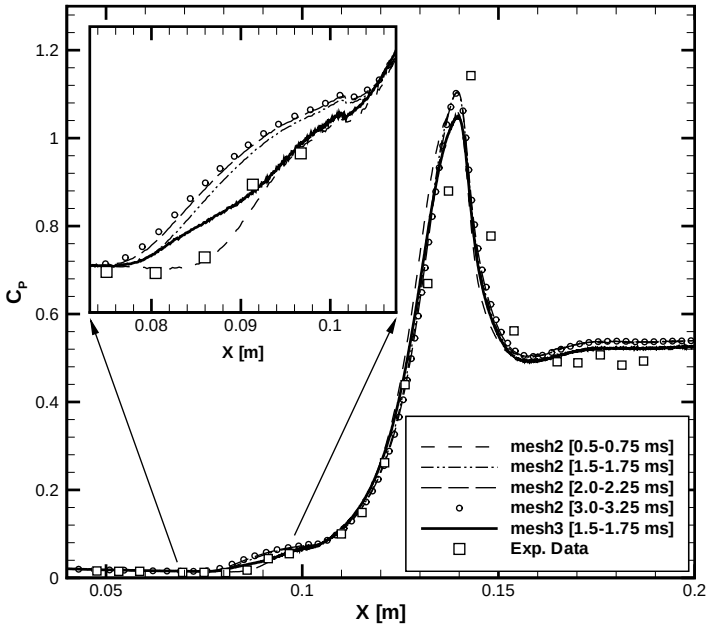


Figure 6.3: DSMC (MONACO) grid convergence and steady-state for the hollow cylinder-flare geometry.

In addition to cell size and timestep restrictions, a DSMC simulation must be allowed to reach steady-state before sampling of the solution begins. In order to demonstrate this, Fig. 6.3 shows the MONACO solution on mesh2 sampled at various times. A close-up view of the separation region clearly demonstrates that the size of the separation region increases significantly between the sampling periods of 0.5-0.75 ms and 1.5-1.75 ms and remains relatively constant thereafter. The precise location of flow separation is best determined as the location on the surface where the shear-stress becomes negative. Although profiles of shear-stress are not shown here (and were not measured experimentally), the point of separation (x_{sep}) predicted by MONACO for each sampling interval is listed in Table 6.2. By noting the percentage change in x_{sep} per millisecond, it is evident that the size of the separation region has effectively reached steady-state after 1.5 ms. DSMC simulations of this same case

Table 6.2: DSMC prediction for the location of flow separation and approach to steady-state.

sampling interval [ms]	x_{sep} $[m]$	% change / ms
0.5 - 0.75 ms	0.09316	– –
1.5 - 1.75 ms	0.08936	4.0%
2.0 - 2.25 ms	0.08856	1.8%
3.0 - 3.25 ms	0.08806	0.6%

performed by Moss and Bird [73] draw the same conclusion and predict a similar level of agreement with experimental data as the MONACO results in Fig. 6.3. Compared with the NS results in Fig. 6.2(b), DSMC is seen to better predict the extent of flow separation for the Run 11 conditions.

Finally, comparison of the solutions obtained on mesh2 and mesh3 (for the same sampling interval) demonstrates that mesh2 provides adequate resolution for a DSMC simulation. Although the mesh2 solution predicts a slightly higher surface pressure inside the separated region, the location of flow separation and the remainder of the profile agree very well with the mesh3 solution. It is interesting to note that for the Run 11 conditions, the mesh density required to obtain a grid-independent NS solution (mesh2) is the same as that required by a corresponding DSMC simulation.

6.2 Hollow Cylinder-Flare Simulation Results

6.2.1 Problem Setup and Progression of hybrid solution

The MPC simulation begins with a NS solution obtained on a mesh designed to solve the NS equations for the hollow cylinder-flare problem. An estimate of local continuum breakdown is provided by the gradient-length Knudsen number, Kn_{GL} and for the hollow cylinder-flare configuration, the cutoff value for continuum break-

down is specified as $Br_{\text{cutoff}} = 0.03$. Next, DSMC regions are extended further into the continuum region by a number of cells ($n_{\text{overlap}} = 4$) in order to create an overlap region where both methods are used. Extended DSMC regions are further surrounded by a layer of DSMC boundary cells and NS regions are surrounded by a layer of NS boundary cells. After all particle and continuum regions and boundaries have been initialized, the mesh inside DSMC regions is refined to the local value of λ. This procedure (described previously in Chapter III) uses flow gradient information from the NS solution to set more stringent refinement in the direction of flow gradients and relax the refinement in other directions. The initial particle region (including the overlap) for the hollow cylinder-flare is shown in the top of Fig. 6.4 and a portion of the mesh is depicted in the bottom of Fig. 6.4. Initially particles are generated in all DSMC cells (using a Chapman-Enskog velocity distribution [56]) such that they are consistent with the NS solution. As the DSMC regions iterate and particles collide with each other and with surfaces, the DSMC solution proceeds away from the initial continuum NS solution towards the correct non-equilibrium solution. As this progression and relaxation occurs, DSMC and NS regions are loosely coupled and interface locations are periodically updated throughout the simulation. As an example, the final steady-state interface locations for the hollow cylinder-flare simulation is shown in the top of Fig. 6.4.

The progression of the MPC solution to the hollow cylinder-flare problem is investigated at various stages during step 2 of the MPC numerical cycle (see Section 5.2.3). In Fig. 6.5 the translational temperature profile along cut C2 (shown previously in Fig. 6.4) is plotted at four intervals; t_0, t_1, t_2, and t_3. Since the initial boundary layer development (at the leading edge) is enclosed entirely by a DSMC region (see top of Fig. 6.4), both temperature jump and velocity slip at the surface are captured. As

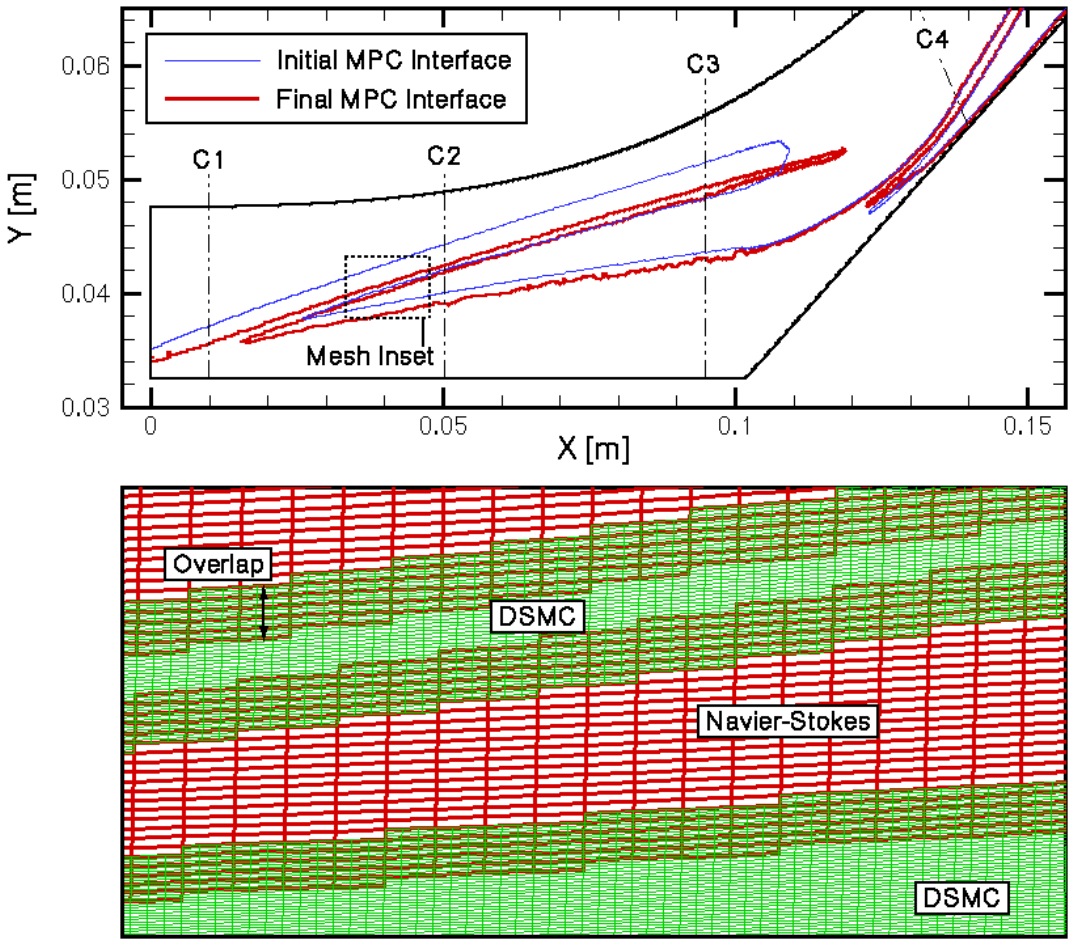


Figure 6.4: Interface locations (top) and mesh refinement (bottom) for the hollow cylinder-flare.

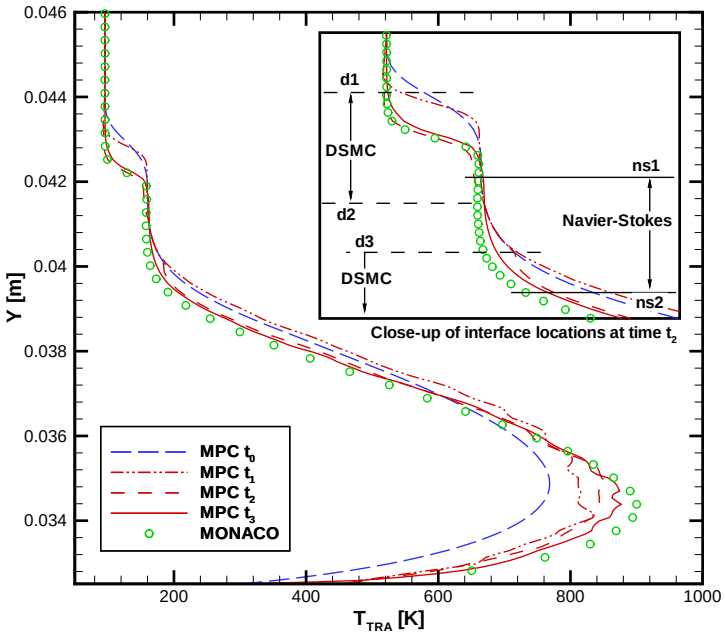


Figure 6.5: Hollow cylinder-flare MPC simulation.

a result, the weak leading edge shock predicted by DSMC lies closer to the surface than predicted by the NS equations employing no-slip conditions. As DSMC regions are allowed to iterate (from time t_0 through t_3), the solution progresses significantly towards the final correct solution predicted by a full MONACO (DSMC) simulation including velocity slip, temperature jump, and thermal non-equilibrium. As this progression occurs the interface locations are updated, which as seen in Fig. 6.4, actually follow the shock movement towards the lower position predicted by full DSMC. The movement of the hybrid interfaces is entirely facilitated by the overlap region, which allows for local differences between DSMC and NS solutions to develop and be detected by the continuum breakdown parameter. A close up look at the shock region in Fig. 6.5 shows three overlapping regions; DSMC inside the shock (lines $d1 - d2$), NS in the continuum region between the shock and boundary layer (lines $ns1 - ns2$), and DSMC in the boundary layer (below line $d3$). The MPC solution is

initialized to the NS solution (the profile at time t_0) and after 2000 DSMC timesteps has progressed to the profile labelled as t_1, which at this point has not clearly progressed in the proper direction. However, in step 2 of the MPC cycle, as long as the DSMC regions and interfaces continue to change, no information is transferred to the NS regions and DSMC iterations continue. After 4000 more DSMC timesteps, the solution reaches the profile labelled as time t_2. At this point, variations inside the DSMC regions have ceased and repeated implementation of step 2 no longer results in movement of the interfaces. The MPC cycle has reached step 3 and information is now transferred to the NS regions. It is important to notice the role of the overlap region between lines $d3$ and $ns2$. Although the t_2 profile is fixed by incorrect boundary conditions at line $d3$, within this overlap region it recovers and by line $ns2$, is very close to the final DSMC result. Thus, during step 3 of the MPC cycle, when information is transferred to the NS regions, the new NS boundary condition is taken on line $ns2$. After convergence of the NS regions the profile labelled t_3 is obtained, which shows significant progression towards the correct DSMC profile. At this point, step 3 is complete and the cycle returns to step 2 with new (and more accurate) DSMC boundary conditions supplied from the NS regions at lines $d1$, $d2$, and $d3$.

6.2.2 Flow Field Quantities

The current implementation of the MPC method is restricted to serial (single CPU) processes. Due to computational time and memory restrictions, the initial NS solution used is that obtained on mesh1. DSMC regions are further refined to the local value of λ, which results in a mesh density slightly higher than that of mesh2. A portion of this hybrid mesh was shown earlier in Fig. 6.4. As seen in

Fig. 6.2(b) the NS solution on mesh1 is very close to the grid-converged result on mesh2 except that the separation region is slightly smaller. The size of the separation region on mesh1 is still sufficiently larger than that predicted by DSMC and thus still provides a challenge for the MPC method. Beginning the MPC simulation using the NS solution on mesh1 also allows testing of the mesh refinement procedures for the hollow cylinder-flare.

In order to provide detailed comparisons between full NS, DSMC, and MPC simulations, flow properties are extracted along various cuts normal to the cylinder-flare surface. The results along cuts $C1$, $C2$, $C3$, and $C4$ (from Fig. 6.4) are shown in Figs. 6.6, 6.7, 6.8, and 6.9, respectively. Near the leading edge, the breakdown parameter sets up a DSMC region that completely envelopes the initial boundary layer growth and leading edge shock. As seen in Fig. 6.6, the shock/boundary layer height predicted at location $C1$ by full DSMC is lower than that predicted by the initial NS solution. In addition, DSMC is seen to predict significant velocity slip, temperature jump, and thermal non-equilibrium at this location. The MPC method is seen to very accurately reproduce full DSMC results near the leading edge and at the same time has lowered the interface location from its initial position of approximately $Y = 0.0364\,m$ to a final position of $Y = 0.0355\,m$. At location $C2$, the shock and boundary layer become more distinct and Fig. 6.7 shows that the MPC method now solves the NS equations in a continuum region between the shock and boundary layer. Again, the MPC method captures the new shock location as well as the thermal non-equilibrium and temperature jump predicted by full DSMC very accurately. The MPC profile in Fig. 6.7 also demonstrates how the changing DSMC regions have successfully set new NS boundary conditions, which significantly shift the continuum solution between the shock and boundary layer. The progression

of the MPC solution at cut $C2$ was detailed in Fig. 6.5. It is important to note in the continuum region between the shock and boundary layer that the flow is in thermal equilibrium and is adequately modelled by the NS equations. This adds further credibility to the use of the gradient-length Knudsen number [50, 51] to detect continuum breakdown. Location $C3$ cuts through the separation region which is evident in Fig. 6.8 where the x-velocity (U) is negative. Here, DSMC is seen to predict a smaller separation region than the initial NS solution. Similar to a full DSMC simulation, an MPC simulation also requires significant time for the separation region to transition from the initial NS solution and reach a steady-state. However, since the MPC interface envelops the entire separation region, the interface locations do not move at all during this lengthy time. This causes a problem for the MPC cycle outlined in the previous section since little or no interface movement signifies steady-state at which point the MPC cycle wants to lock the interfaces and begin sampling. In order to allow the MPC cycle to handle slow recirculating flows a user-defined parameter, Δt_{ss}, is introduced which specifies the minimum number of iterations (or minimum physical time) before sampling begins. This parameter is analogous to the user-defined parameter specifying the sampling interval employed in full DSMC simulations. The MPC solution along cut $C3$ is plotted for various values of Δt_{ss} and Fig. 6.8 shows that a time of 0.3 ms is sufficient for the MPC simulation to reach steady-state. This is significantly less than the time required for a full DSMC simulation to reach steady-state (see Fig. 6.3) and is entirely attributed to an MPC simulation being initialized with particles corresponding to a full NS solution. Finally, Fig. 6.9 depicts the temperature profiles at location $C4$, which lies after re-attachment and just downstream of the shock-shock interaction region. Here, DSMC predicts the translational temperature increase due to the strong shock to

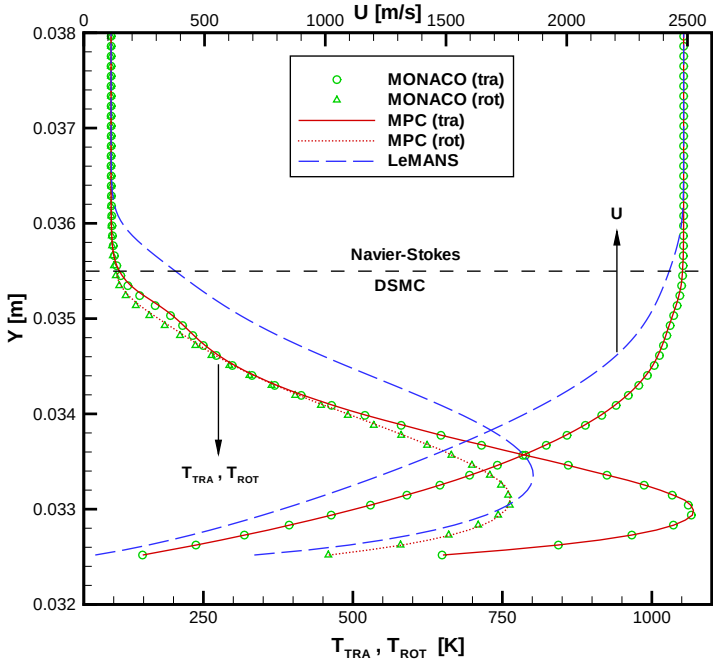


Figure 6.6: Temperature and velocity profiles along C1.

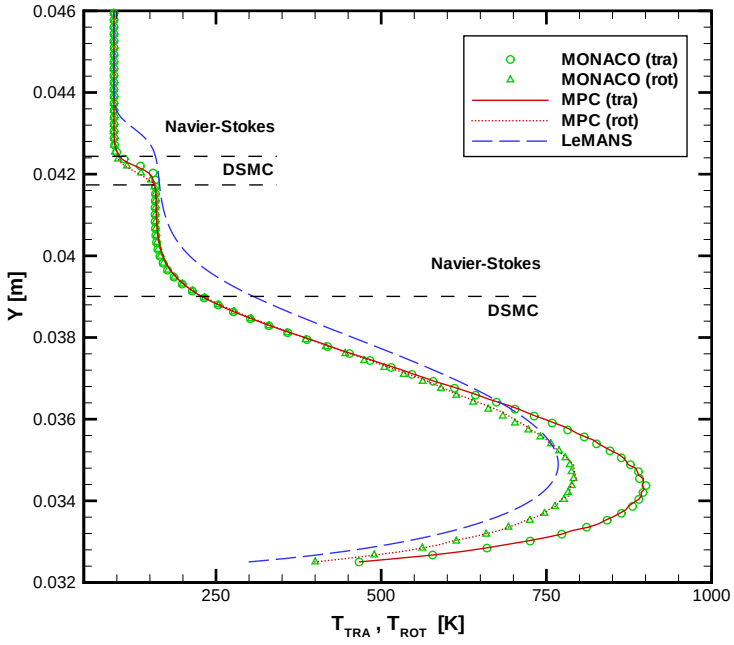


Figure 6.7: Temperature profiles along C2.

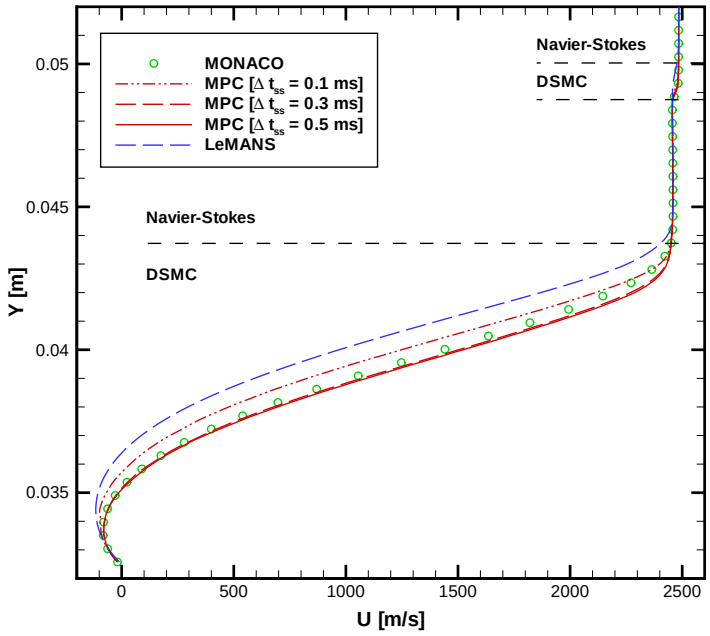


Figure 6.8: Velocity profile along C3.

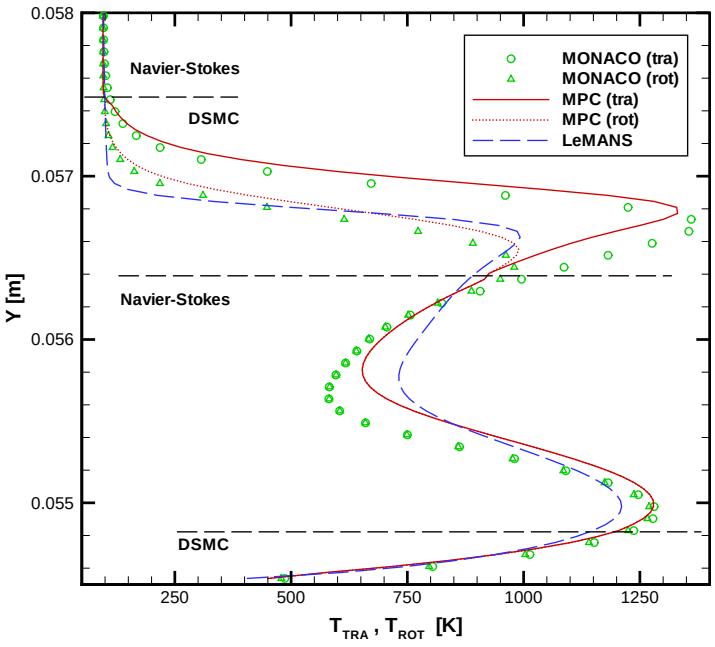


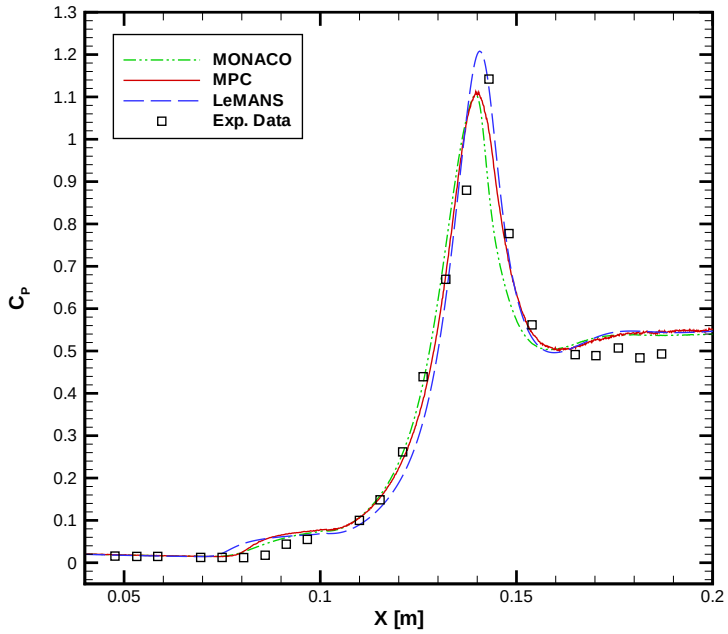
Figure 6.9: Temperature profiles along C4.

occur further upstream (or higher) than predicted by the NS equations. In addition, DSMC shows a high degree of thermal non-equilibrium inside the shock region. Both of these effects are captured well by the MPC method. In addition to using DSMC for the strong shock, the continuum breakdown parameter sets up a very thin DSMC region inside the hypersonic boundary layer next to the surface (only 6-10 DSMC cells thick). This causes problems with the heat transfer predicted by the MPC method and will be discussed in the next paragraph. Other than the shock, it is important to note that the remainder of the flow at location $C4$ is seen to be in thermal equilibrium and the MPC method is seen to move the initial NS solution into better agreement with the full DSMC solution. However, it should also be noted that at this location, full DSMC and full NS solutions agree quite well, especially close to the surface, indicating that at this point on the flare, the flow may lie in the continuum regime.

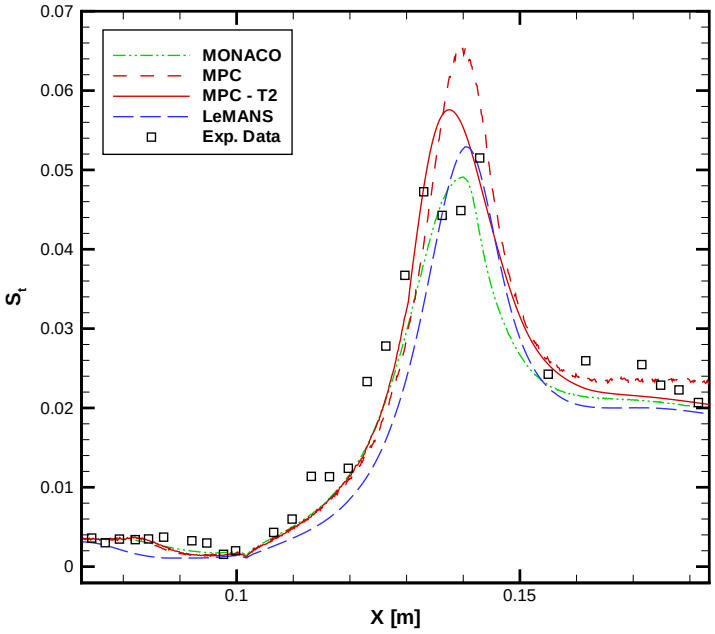
6.2.3 Surface Properties

The coefficient of pressure (C_P) predicted by full DSMC, NS, and MPC simulations is plotted in Fig. 6.10(a). The size of the separation region has been successfully reduced by the MPC method and agrees well with that predicted by full DSMC. The coefficient of heat transfer, also referred to as the Stanton number [$S_t = q/(0.5)\rho_\infty u_\infty^3$] where q represents the amount of energy transferred to the surface per unit time per unit area, is plotted in Fig. 6.10(b) for each simulation method. Although not shown, the MPC simulation reproduces exactly the results for heat transfer predicted by full DSMC near the leading edge. This is expected, since the MPC method was shown in Fig. 6.6 to accurately capture velocity slip, temperature jump, and thermal non-equilibrium near the leading edge. In Fig. 6.10(b),

the MPC method is seen to improve the prediction of heat transfer in the separation region over the initial NS prediction. However, downstream of the flare junction the MPC results for heat transfer begin to differ from full DSMC results and over-predict the heat transfer by 20-30% for $X > 0.135\ m$. As mentioned above, and shown previously in Figs. 6.4 and 6.9 the DSMC region created by the MPC method next to the flare surface is very thin. A similar, thin DSMC region is found near the wedge surface for the hybrid simulations of Wu *et al.* [40]. Although surface property results are not analyzed in the article by Wu *et al.*, close analysis of the thin DSMC region next to the wedge surface [40] shows similarities with Fig. 6.9 of this chapter. Specifically, small differences in the steep temperature gradient compared with the gradients predicted by either pure DSMC or NS are visible, which may alter the calculated heat transfer significantly. In order to alleviate this problem, the Br_{cutoff} parameter could be lowered in order to enlarge the DSMC regions. However, such a change is found to adversely affect other regions of the flow. Instead, DSMC regions could be enlarged near the surface only. However this approach may be highly dependent on the geometry and flow conditions and would be difficult to implement in a general manner. Another approach comes from realizing that aside from the strong shock wave, the re-attached flow near the flare surface is well within the continuum regime and DSMC may not be necessary at all. Comparing the initial NS solution to the full DSMC solution in Fig. 6.9 reveals no significant difference between DSMC and NS solutions (except inside the shock wave). Additionally, in Fig. 6.2, the NS simulation is seen to accurately predict the experimental data downstream of the flare junction. In order to address this, a second MPC simulation (MPC - T2) is run where DSMC regions are only allowed to develop prior to $X = 0.13\ m$ and the NS equations are solved for the remainder of the flare. As seen in Fig. 6.10(b), this



(a) Coefficient of pressure.



(b) Coefficient of heat transfer (Stanton number).

Figure 6.10: DSMC, NS, and MPC surface properties for the hollow cylinder-flare.

improves the heat transfer result over that predicted by the original MPC simulation.

Although the heat transfer prediction is improved by eliminating the thin DSMC region next to the surface, it does not explain exactly why such a thin DSMC region produces inaccurate results. If the region next to the surface is in fact a near-equilibrium region, the DSMC method should still provide an accurate model. One possible limitation of the current MPC implementation is that particles are generated uniformly within DSMC boundary cells. As described in Chapter III, particles are randomly placed within each DSMC boundary cell and the properties used to generate the Chapman-Enskog velocity distributions are those of the larger NS cell. Thus, both the spatial and velocity distributions of generated particles is the same for all refined DSMC cells contained within the larger NS cell. These distributions may not be physically correct if substantial mesh refinement is used and/or the boundary cell is located in a high-gradient region of the flow. In order to remedy this situation, instead of being randomly placed within each cell, particles could be placed along density gradients. Furthermore, the macroscopic properties used to sample from a Chapman-Enskog distribution could be interpolated across the larger NS cell such that a more accurate value is used in each refined DSMC cell. The difference could be quite substantial in the high-gradient region see in Fig. 6.9. A future study implementing these new procedures and investigating any improvement for this case is required. It should be noted that the hollow cylinder-flare simulation is the only simulation presented in this thesis in which interfaces are located in such high-gradient regions. Thus any error associated with the current particle generation procedure may not have produced noticeable errors for the other simulations presented.

6.3 Computational Performance

The speed-up factor achieved by the MPC method for the hollow cylinder-flare problem is determined in the same manner used for the blunt-body simulations presented in Section 5.5 of Chapter V. The performance parameters of most interest are listed in Table 6.3 for pure DSMC, MPC, and MPC-T2 simulations. MPC simulation of the hollow cylinder-flare is seen to use 53% of the particles and 80% of the memory required by a full DSMC simulation. Similar to the planetary probe simulation discussed in the previous Chapter, the time required for initial NS simulation plus the time required for the MPC simulation to reach steady-state is less than that required by a full DSMC simulation to reach steady-state. The ratio of iterations required for the MPC method to reach steady-state (after obtaining the initial NS solution) to that required by full DSMC simulation is seen to be 20% in Table 6.3. However, despite reaching steady-state much faster than full DSMC, the MPC method still requires 71% of the time needed to obtain a full DSMC solution (a speedup factor of only 1.4). This signifies that the time required for the initial NS solution and NS operations within the MPC cycle are not negligible compared with the time spent simulating DSMC regions. Since the flare region has a high density and is located further away from the axis (recall no cell-weighting is used for the hollow cylinder-flare), a large portion of the particles in a full DSMC simulation are

Table 6.3: Computational efficiency of the MPC method for the Hollow Cyl. Flare.

HCF	speedup	memory	particles	init. NS	steady-state	sampling
DSMC	100%	100%	100%	N/A	100%	100%
MPC	71% (1.4x)	80%	53% (1.9x)	4%	20%	100%
MPC-T2	10% (10x)	15%	7.3% (13.7x)	4%	20%	100%

located in this flare region. As seen in Table 6.3, when the flare region is simulated entirely using the NS equations (case MPC - T2), a significant number of particles are eliminated within the MPC simulation, which increases the speedup factor proportionally. Specifically a full DSMC simulation requires 13.7 times more particles than the MPC - T2 simulation, which obtains an accurate solution approximately 10 times faster. However, as mentioned previously in section 6.1.1, NS simulation of the hollow cylinder-flare problem (Run 11 conditions) requires a similar mesh resolution as DSMC. As seen in Table 6.3, the initial NS simulation takes 4% of the time required by a full DSMC simulation. If variable DSMC timestepping and subcell procedures are used in the simulation of the hollow cylinder-flare, it may be possible to reduce the number of simulated particles and increase the speed of full DSMC simulation by an order of magnitude [73]. The time required to obtain a full DSMC solution then becomes comparable to the time required to solve the NS equations and hybrid simulation of this shock-interaction flow is no longer practical or beneficial. A more practical hybrid simulation may involve the simulation of an entire hypersonic vehicle (such as NASA's X-43A vehicle) for which the majority of the flow is continuum. The MPC method could then be used to embed a full DSMC simulation of the shock interaction region (the entire hollow cylinder-flare problem) within this much larger continuum flow.

The hollow cylinder-flare configuration (CUBRC-Run 11 conditions) has previously been simulated by Wang and Boyd using a hybrid DSMC-IP-NS numerical method [74]. In this work, DSMC and NS regions are strongly coupled and therefore information is transferred at each simulation timestep. The hybrid method uses the Information Preservation (IP) method [37] to reduce the statistical scatter inherent when evaluating macroscopic properties at each timestep within particle regions.

The IP method reduces statistical scatter by preserving and updating macroscopic information for each DSMC simulation particle in addition to microscopic particle information. This method was developed for low-speed simulation of micro-flows and its application to hypersonic flow introduced computational stability limitations. For simulation of the hollow cylinder-flare, the timestep required to maintain stability of the IP method was reported to be 2.5×10^{-10} s. This timestep is 20 times smaller than the DSMC timestep shown suitable for the hollow cylinder-flare problem in the results of this chapter. In addition, the DSMC-IP-NS hybrid method used an explicit NS solver (with no time-scale decoupling) in continuum regions so that the timestep used in both particle and continuum regions was 2.5×10^{-10} s. In contrast, the timestep used by the MPC method in DSMC regions is 20 times larger, and by using an implicit NS solver, the timestep in continuum regions is 400 times larger than that used by the DSMC-IP-NS method. In addition, within the DSMC-IP-NS hybrid method, the same mesh resolution was used for both particle and continuum regions (1000x200 cells). The timestep limitation imposed by the IP method for this flow combined with the lack of spatial scale decoupling resulted in the DSMC-IP-NS method being at least two times slower than a full DSMC simulation while requiring more memory [74]. Indeed, the time and spatial scale decoupling employed by the MPC method allow continuum regions to be solved with negligible computational resources compared to that required for particle regions, thereby resulting in significant speedup factors for near-continuum flows.

6.4 Conclusions of the Hollow Cylinder-Flare Simulations

The MPC algorithm developed in previous chapters and previously applied to blunt-body flows is here applied to a hypersonic shock interaction flow. In applying

the MPC method to the hollow cylinder-flare geometry and the Run 11 experimental flow conditions, four major conclusions are drawn.

1) In comparison with simulations of blunt-body flows, application of the MPC method to the hollow cylinder-flare problem demonstrates new essential capabilities. Such capabilities include the ability of MPC interfaces to track moving flow features, and the ability of the MPC cycle to significantly alter NS boundary conditions which then shift the solution in NS-regions of the simulation. An important result is that these capabilities are demonstrated for a loosely coupled approach that is able to maintain spatial and temporal scale decoupling throughout the simulation.

2) For flow over the hollow cylinder-flare geometry, an MPC simulation initialized with a NS solution is able to transition away from the no-slip NS solution and reproduce the velocity slip, temperature jump, thermal non-equilibrium, and surface properties predicted by full DSMC near the leading edge. MPC simulation also successfully reduces the over-sized separation region predicted by the NS equations to the size predicted by DSMC and experimental results. With the exception of the flare region, MPC simulation accurately reproduces DSMC results approximately 1.4 times faster than full DSMC simulation while using 80% of the memory. When the NS equations are used to simulate the entire flare region downstream of re-attachment, the resulting MPC simulation achieves the same level of accuracy approximately 10 times faster using only 15% of the memory.

3) The gradient-length Knudsen number predicts continuum breakdown very well for this shock-interaction flow with the exception of the dense flare region of the hollow cylinder-flare problem. The parameter has difficulty in very dense regions with large gradients where it predicts continuum breakdown near the surface even though the NS equations are shown to be valid in this region. Furthermore, the transfer

of information across such thin DSMC regions is shown to lead to a 20-30% over-prediction in heat transfer along the flare. Two possible solutions to this problem are suggested. The first recommends that such thin DSMC regions be avoided in a loosely-coupled hybrid approach. This may require further investigation into the failure of the continuum breakdown parameter in this region. The second possible solution involves generating particles along the density gradient within DSMC boundary cells instead of random placement. When interfaces are located in high-gradient regions, initializing particles with more physically accurate spatial and velocity distributions may lead to better heat transfer results.

4) The fine resolution required to solve the NS equations for the hollow cylinder-flare problem combined with the possibility of efficient full DSMC simulation (utilizing variable timestepping and subcell procedures) lead to the conclusion that the hollow cylinder-flare problem is not practically suited for a hybrid simulation. Rather, the entire shock-interaction flow is better simulated using pure DSMC and embedded as a single particle region (perhaps around a control surface) in a hybrid simulation of an entire vehicle.

CHAPTER VII

Modular Implementation

The level of sophistication and complexity of state-of-the-art engineering simulation codes is constantly increasing as advancements in computer performance continue. Most incorporate advanced physical models and complex convergence acceleration techniques. As a result, the use of modules in simulation codes is now routine. In hypersonics, there are mesh generation packages, equilibrium chemistry modules, radiation modules, ablation modules, and advanced matrix solving packages. Due to the complexity of such engineering codes, attention must be focused on carefully validating and documenting such packages of physical and numerical models [75, 76]. Once a complex piece of code has been debugged and validated over many years, it is highly beneficial to keep the source code completely intact and incorporate it within a larger engineering code as a module. Such an approach holds much potential for multi-scale and multi-disciplinary engineering simulations. A modular implementation first allows for rapid development of a multi-scale method such as the hybrid particle-continuum method developed in this thesis. In addition, such an implementation allows for rapid incorporation of new updates in any of its individual modules.

7.1 Source Code Organization

Both MONACO and LeMANS source codes consist of a single main function as well as a list of many sub-functions that are compiled and linked together using a makefile. Each main function (`monaco.c` and `lemans.c`) is then responsible for calling all sub-functions in a certain sequence that logically implements the respective numerical method. In order to implement the modular particle-continuum method, only 15 additional hybrid functions are required. This organization is portrayed in Fig. 7.1. Here, DSMC, NS, and hybrid functions are linked together by editing the makefile, such that hybrid functions can call any function from either the DSMC or the NS module. The original main functions from both DSMC and NS modules (`monaco.c` and `lemans.c`) are incorporated into a single main function (`hybrid.c`). The original DSMC and NS algorithms are kept available using compiler flags so that a user may run the simulation in either pure DSMC (MONACO) or pure NS (LeMANS) mode. When running in hybrid (MONACO-LeMANS) mode, the function `hybrid.c` calls a mixture of DSMC, NS, and new hybrid functions that logically implement the MPC numerical cycle detailed in Section 5.2.3. The general structure of the new, hybrid main function is shown in Algorithm [1]. Since high-level DSMC and NS functions can be used as-is, currently only 15 new hybrid functions are required, 4 of which are solely devoted to input and output of simulation data. As portrayed in Fig. 7.1, with such an organization of source code it is clear which functions are responsible for pure NS operations, pure DSMC operations, and additional hybrid operations. This is very useful when further developing such a large hybrid DSMC-NS code.

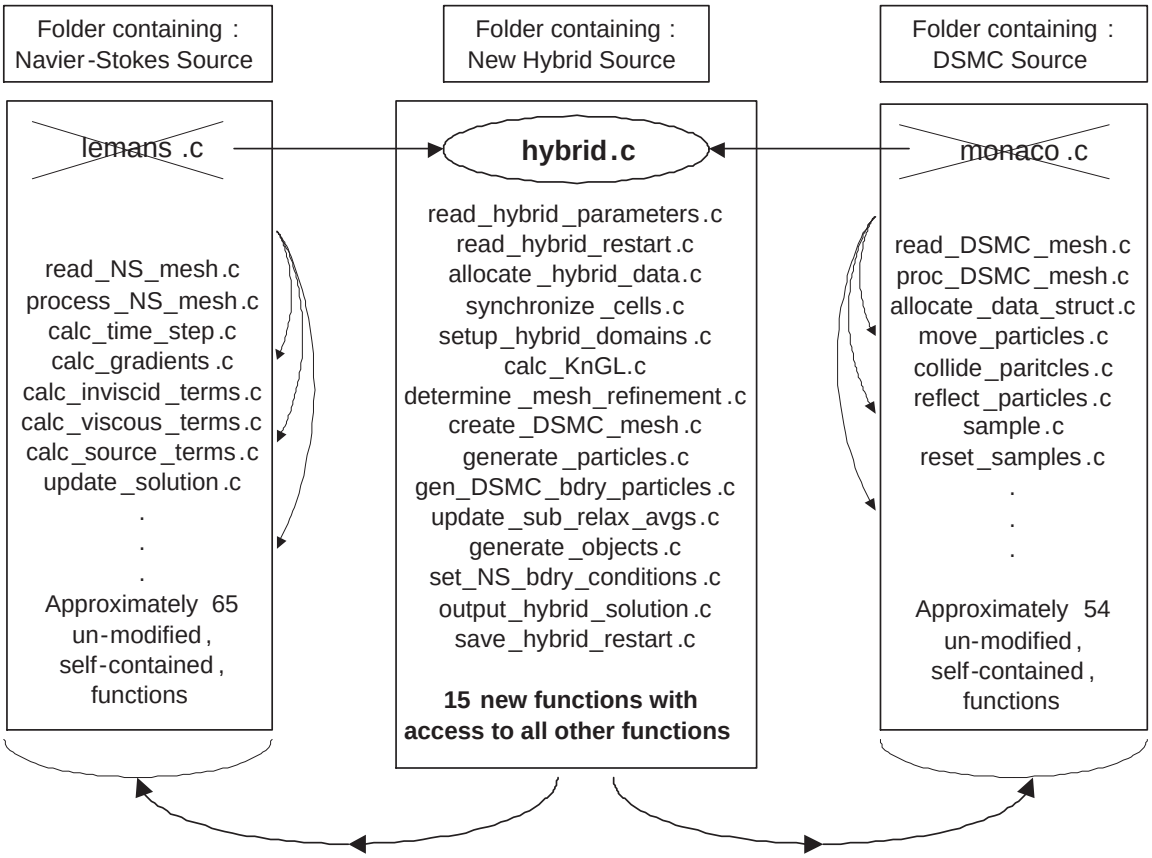


Figure 7.1: Modular organization of source code.

Algorithm 1 General structure of `hybrid.c`

function MAIN( )
  if (compiler flag = –PART) **then**
    `monaco.c` source code

  else if (compiler flag = –CONT) **then**
    `lemans.c` source code

  else if (compiler flag = –HYBRID) **then**
    ▷ A combination of MONCACO, LeMANS, and Hybrid function commands
    ▷ See Algorithms [2,3]
  end if
end function

7.2 Data Structure Organization

In order to utilize the existing, un-modified source code for both DSMC and NS modules, the exact data structures used by each module must not be modified in any way. Thus, any new information required by the hybrid functions must be stored in a separate, newly created hybrid data structure. The modular organization of all source code and data structures is portrayed in Fig. 7.2. Both NS and DSMC modules contain their original, un-modified source code (library of functions) written to operate on their own existing data structures. For example, LeMANS operates on a data structure of cells containing cell-centered information as well as on data structures containing face and node information. The information stored in this data structure is referred to as **NSdata**, which contains the macroscopic flow quantities (Q) for each cell (NSdata[*cell*].Q[*variable*]) as well as the geometry and neighboring cells (cell connectivity) for each cell in the continuum NS mesh. As shown in Fig. 7.2, MONACO operates on a data structure of cells that each contain a linked-list of particles. The information contained within this data structure is referred to as **DSMCdata**, which contains the microscopic properties of each particle within the cell (DSMCdata[*cell*].particles) and also contains the geometry and cell connectivity for each cell in the refined DSMC mesh.

In order to exactly retain these existing data structures, a new *hybrid* data structure is created to store the additional, hybrid information required by the MPC method. All hybrid information is handled at the continuum level and so for each cell in the NS mesh, a corresponding hybrid structure contains the cell's hybrid information. As seen in Fig. 7.2, the information stored in this data structure is referred to as **HCdata**. Such information includes the cell-type, the refinement criteria, the

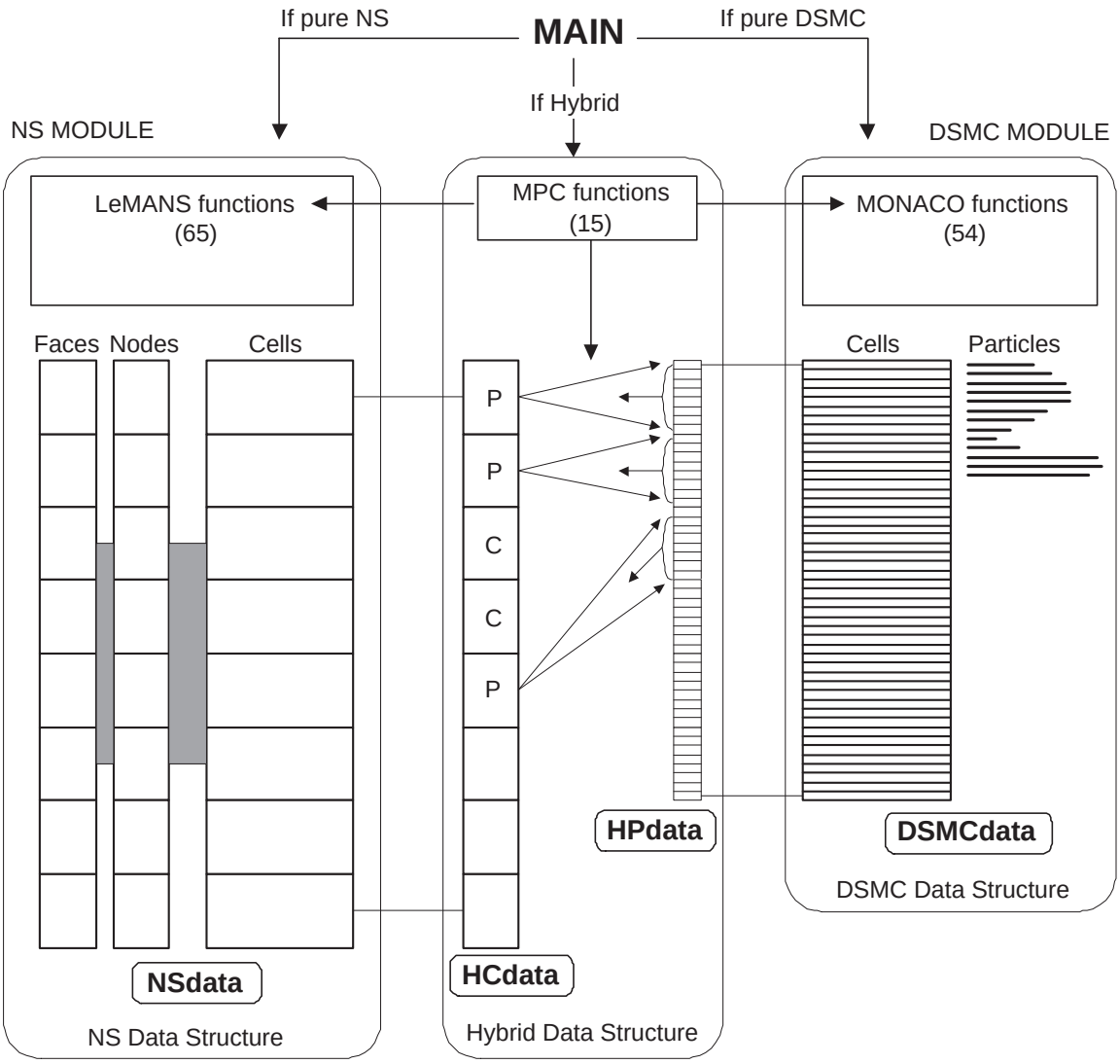


Figure 7.2: Modular organization of data structures.

indices of the first and last refined DSMC cell, as well as the sub-relaxation averages taken over all refined DSMC cells and particles within them. As portrayed in Fig. 7.2, the indices of HCdata correspond exactly to the indices of NSdata. The HCdata structure is detailed in Algorithm [4] of Appendix A. For convenience, the hybrid data structure also contains an array that matches the particle cell array in the DSMC module. As portrayed in Fig. 7.2, this array is referred to as **HPdata** and has indices corresponding exactly to the indices of DSMCdata. The only information contained in HPdata is a single integer for each DSMC cell representing the indices of the larger NS cell in which it is located. Such a modular implementation then requires only a few hybrid functions to both retrieve and modify information in either of the NSdata, DSMCdata, HCdata, or HPdata structures. For example, one hybrid function accesses the initial NS solution (from NSdata[*cell*].Q[*variable*]), calculates the continuum breakdown parameter in each cell, and sets both the cell types and the refinement criteria for each cell in HCdata. Additional hybrid functions handle information transfer by simply retrieving data from one data structure, processing it, and inserting it into the other data structure at appropriate times during the simulation. The way in which both DSMC and NS modules operate on their own data (contained in NSdata and DSMCdata) remains completely un-changed.

It is certainly possible to develop a hybrid particle-continuum code that uses a single data structure. For example, each continuum NS cell structure could contain a list of refined DSMC cell structures, which in turn contain a list of particle structures. Maintaining separate data structures involves some duplication of information; however, such additional storage is found to contribute little to the overall memory requirements of the simulation. Certainly, a single data structure would eliminate duplicate information and contain all data in a single, perhaps more “elegant” array.

However, all DSMC and NS functions would have to be modified to operate on this new hybrid data structure. Such modifications may not be major but they are numerous. The above modular organization of source code and data structures holds several important advantages over developing source code to operate on a single data structure. Since no modification is required for the numerous DSMC and NS functions, development time of the hybrid code is significantly reduced. One must only concentrate on writing the relatively few hybrid functions required to interact with existing DSMC and NS data structures. Second, if new versions of either DSMC or NS modules are released (involving even major updates) the source code of these new module versions can be inserted as-is within the modular framework. One must only ensure that all hybrid functions are modified to remain consistent with the changes in either DSMC or NS module. Although these modifications may be major, they are restricted to the relatively few hybrid functions only. Whereas, if a single data structure is used, the modifications involved in the new module versions must be re-implemented for the DSMC and NS functions within the hybrid code. Another important advantage of maintaining modular data structures is that the optimal data structure for the NS module is not necessarily suited for the DSMC method. For example, in order to move all particles for one timestep, MONACO loops through all cells. However, in order to update all cell quantities in the NS module, LeMANS loops over all faces. The flux quantities computed at each face are then added to both left and right cells. As seen in Fig. 7.2, MONACO and LeMANS data structures have significant differences. A single hybrid data structure accommodating both update schemes would arguably be as large as using them both separately. The advantage of a modular implementation is that once the boundary conditions are set, both NS and DSMC modules are free to update their information in a manner best

suited for each numerical method. Finally, the existing method of parallelization employed by each module (which is heavily dependent on data structure) would require significant modification before being employed on a single particle-continuum data structure. The implications of using a modular implementation when developing a parallel hybrid code will be discussed in Section 7.4.

7.3 Hybrid Functions

As outlined in Algorithm [1], when run in hybrid mode (compiler flag = -HYBRID) the source code consists of a mixture of existing functions from DSMC and NS modules as well as newly created hybrid functions. The main function, `hybrid.c`, is responsible for exactly implementing the MPC numerical cycle previously outlined in Section 5.2.3. This function is detailed in Algorithms [2] and [3]. Algorithms [1] through [15] in Appendix A then represent the complete list of additional hybrid functions called in `hybrid.c` in order to implement the MPC method. All other function calls appearing in any hybrid function are existing high-level DSMC and NS functions and are therefore not detailed. Just as these high-level functions are called within `monaco.c` and `lemans.c`, they can be called within `hybrid.c` or any of its sub-functions as well.

In Algorithm [2], the function `hybrid.c` begins by initializing all variables required by the DSMC, NS, and hybrid numerical methods. Initialization of DSMC and NS variables is identical to that required in each module's original source code (`monaco.c` and `lemans.c`). The first step in the MPC method is to load in an initial NS solution on an initial NS mesh. The same high-level functions called in `lemans.c` are also used here. Thus by line 6, the NS data structure (NSdata in Fig. 7.2) is allocated and filled with an initial NS solution. Next, memory is allocated for the

hybrid data structure (HCdata in Fig. 7.2) whose indices exactly correspond to those of NSdata. Memory is allocated to store the cell types, refinement criteria, indices of the first and last refined DSMC cells, and memory for the sub-relaxation averages within each NS cell. Initial interface locations are then determined by the function `setup_hybrid_domains()`. This function uses the information from NSdata to calculate the continuum breakdown parameter. Hybrid domains (including particle, continuum, and overlap regions) are set up and the resulting cell types are stored within HCdata. Next, at line 10 within function `determine_mesh_refinement()` the information in NSdata is used to determine the refinement required in each NS cell, which is also stored in HCdata. Finally, this refinement criterion along with the geometry from NSdata is used to create a refined DSMC mesh in function `create_DSMC_mesh()`. The mesh is then written in the format required by the DSMC module. In the process, the array HPdata is created to correspond to the indices of the DSMC mesh and the links between HCdata and HPdata are created and stored within the hybrid data structure. Following this, at line 13 the DSMC mesh is used to completely initialize the DSMC data structure (DSMCdata) using the same high-level functions that accomplish this in `monaco.c`. At this point DSMCdata does not yet contain any simulation particles. The hybrid function, `generate_particles()`, then uses information from NSdata and HCdata to generate particles within each cell of DSMCdata and the number of cells containing particles is recorded. Therefore, by line 15, all NS, DSMC, and hybrid data structures have been created and filled with initial data and particles corresponding to the initial NS solution. Thus the hybrid problem is completely initialized and step 1 of the MPC cycle (see Section 5.2.3) is complete.

The main hybrid loop that begins on line 17 of Algorithm [2] closely resembles

the main loop used by the DSMC method (`monaco.c`), which involves the continual movement and collision of simulation particles. The first difference is that the boundary conditions for DSMC regions are not fixed, rather they are set using NS information. Thus before each DSMC iteration, all particles within DSMC boundary cells are first deleted and then re-generated based on the current NS solution. This is performed in function `generate_DSMC_bdry_particles()` on line 22. The second difference is that during the hybrid simulation, before the hybrid interfaces are locked in position, the solution within DSMC regions is monitored at each timestep. The function `monitor_update_hybrid_solution()` is detailed in Algorithm [3]. Within this function, the sub-relaxation averages are calculated and occasionally interface locations are updated causing particles to be added or removed from the simulation. The final difference compared to a standard DSMC cycle is that the hybrid simulation will pause occasionally to update NS regions of the hybrid solution. The function `update_NS_regions()` is also detailed in Algorithm [3]. The hybrid simulation then continues with the movement and collision of simulation particles using the new NS solution to set the boundary conditions for DSMC regions.

Algorithm [3] completely details how the sub-relaxation averages are used to monitor the hybrid solution and how interface adaptation is performed. The first step is to update the sub-relaxation averages after completion of the previous DSMC iteration. The equations used to compute the averages are listed in Chapter III (Eqns. 3.8 to 3.12) and are computed within the function `update_sub_relax_avgs()`. This function also determines the percentage change in the sub-relaxation average within any NS boundary cell (any cell touching the interface) compared with the NS solution stored in NSdata. If this change exceeds a certain amount then a flag is returned (update_interfaces=1) that signifies that the interfaces should be

Algorithm 2 Implementation of the MPC numerical cycle within `hybrid.c`

```
Initialize MONACO variables
Initialize LeMANS variables
Initialize new Hybrid variables

▷ Load initial NS solution into NS data structure
read_process_NS_mesh()              ▷ allocates and sets up NS data structure
read_NS_restart_file()        ▷ fill NS data structure with initial NS solution

▷ Create and fill Hybrid data structure
allocate_hybrid_data()                                    ▷ see Algorithm [4]
setup_hybrid_domains()                                  ▷ see Algorithms [5-7]
determine_mesh_refinement()                               ▷ see Algorithm [8]
create_DSMC_mesh()                                        ▷ see Algorithm [9]

▷ Create and fill DSMC data structure
read_process_DSMC_mesh()        ▷ allocates and sets up DSMC data structure
new_dsmc_cells = generate_particles()                  ▷ see Algorithms [10,12]
total_dsmc_cells += new_dsmc_cells
▷ All NS, DSMC, and Hybrid data structures are now created and filled

▷ Main Hybrid loop begins
locked=0,      global_update=0,      cycle_counter=0,      NS_flag=0
while (iter < max_iterations) do
    iter++
    cycle_counter++
    generate_DSMC_bdry_particles()                      ▷ see Algorithms [11,12]
    move_collide_all_particles()
    collect_dsmc_samples()

    update_interfaces=0
    if (!locked) then
        ▷ Update sub-relax avgs, possibly move interfaces or set NS_flag=1
        monitor_update_hybrid_solution()              ▷ see Algorithm [3]
        reset_dsmc_sample_values()
    end if

    if (NS_flag=1 OR (locked AND cycle_counter=dsmc_cycle)) then
        ▷ Update NS regions using boundary conditions from DSMC regions
        update_NS_regions()                           ▷ see Algorithm [3]
    end if

end while    ▷ Continue with Main Hybrid loop
```

Algorithm 3 Two major sub-functions with `hybrid.c`

```
1: function MONITOR_UPDATE_HYBRID_SOULUTION( )
2:     update_interfaces = update_sub_relax_avgs()          ▷ see Algorithm [13]
3:     if ( (cycle_counter % 1/theta)=0 ) then
4:         update_interfaces = 1                          ▷ force an interface update
5:     end if
6:     if (update_interfaces) then
7:         synchronize_cells()                             ▷ see Algorithm [14]
8:         setup_hybrid_domains()                          ▷ see Algorithms [5-7]
9:         new_dsmc_cells = generate_particles()           ▷ see Algorithms [10,12]
10:        total_dsmc_cells += new_dsmc_cells
11:        if (new_dsmc_cells / total_dsmc_cells < PctNewCells) then
12:            if (!global_update OR iter < unsteady) then
13:                ▷ Interfaces temporarily within near-equilibrium regions
14:                NS_flag=1
15:            else if (global_update AND iter > unsteady) then
16:                ▷ Final interface locations found, remove overlap regions and lock
17:                setup_hybrid_domains(n_overlap = 1)      ▷ see Algorithms [5-7]
18:                locked=1
19:                reset_dsmc_sample_values()  ▷ prepare for cumulative sampling
20:            end if
21:        end if
22:        global_update=0
23:    end if
24: end function

25: function UPDATE_NS_REGIONS( )
26:    NS_iter=0
27:    while (NS_iter < ns_cycle) do
28:        NS_iter++
29:        set_NS_bdry_conditions()                        ▷ see Algorithm [15]
30:
31:        ▷ Perform a standard NS update using lemans.c functions
32:        calc_time_step()
33:        calc_inviscid_terms()
34:        calc_viscous_terms()
35:        update_solution() ... etc.
36:    end while
37:    if (!locked) then
38:        global_update=1
39:    end if
40:    cycle_counter=0
41:    NS_flag=0
42: end function
```

updated. This level of percentage change is set by the user to a typical value of **percent_change**=20%. Or, if the hybrid simulation has proceeded for a sufficient number (1/**theta**) of iterations without the interfaces having been updated, the flag is also turned on in line 4 of Algorithm [3]. If the flag remains off, then the `monitor_update_hybrid_solution()` function is complete and the hybrid simulation continues with the next DSMC movement-collision process before the solution is monitored again. However, with the flag turned on, the new solution in DSMC regions (stored using the sub-relaxation average) is used to re-evaluate the interface locations. The first step involves the function `synchronize_cells()` on line 7. Within this function all macroscopic sub-relaxation averages in DSMC regions are copied into the NS data structure (NSdata). In this way, the current hybrid solution (even in DSMC regions) is specified at the continuum level on the NS mesh. The function `setup_hybrid_domains()` is then able to operate on only the NS mesh (NSdata) in order to calculate the continuum breakdown parameter and update the cell types in HCdata. As interface locations are updated and new particles are generated in function `generate_particles()`, the number of new DSMC cells created is recorded on lines 9 and 10. If the interfaces move significantly and introduce many new DSMC cells, then DSMC simulation of the particle regions continues. However, if the interfaces move by a negligible amount (new_dsmc_cells / total_dsmc_cells $<$ **PctNewCells**), a flag is turned on (NS_flag=1) on line 14. This signifies that the interfaces are temporarily located in near-equilibrium regions and information is now to be transferred to the continuum domain.

The function `update_NS_regions()` in Algorithm [3] details the steps involved in updating the solution in NS regions. First, the function `set_NS_bdry_conditions()` sets the complete state in all NS boundary cells using the current sub-relaxation av-

erages stored in HCdata. Each NS update then uses the identical high-level function calls (lines 32 to 35 in Algorithm [3]) that are used in the NS module's original source code (`lemans.c`). Large implicit timesteps are used for a set number (**ns_cycle**) of NS timesteps. Immediately after, a flag is set (global_update=1) signifying that the NS solution has just been obtained specifically using the boundary conditions from the current DSMC solution. The main hybrid loop then resumes and DSMC regions continue to iterate using this new NS solution for boundary conditions.

If the new NS solution significantly changes the DSMC boundary conditions then the DSMC solution will continue to change and the hybrid cycle will continue. If so, monitoring of the sub-relaxation average in lines 1-10 of Algorithm [3] will cause further significant adaptation of the interfaces. However, at some point during the hybrid cycle, the new NS solution may not significantly modify the DSMC boundary conditions and therefore the interfaces may be adjusted by a negligible amount. That is, the flag global_update is turned on *and* the change in the number of DSMC cells is negligible (new_dsmc_cells / total_dsmc_cells < **PctNewCells**). In this case, the solution in both NS and DSMC regions is no longer varying and has reached a steady-state with the interfaces located in near-equilibrium regions. At this point (lines 17 to 19 of Algorithm [3]), the overlap regions are removed and the interface locations and cell types are locked. The simulation has now reached step 4 of the MPC numerical cycle (see Section 5.2.3). With the flag, locked=1, the main hybrid loop proceeds to iterate the DSMC method for **dsmc_cycle** timesteps and cumulative samples are collected. This sampled information is then transferred to the NS regions, which are converged for a number (**ns_cycle**) of large implicit timesteps. The result is a scatter-free DSMC solution in non-equilibrium regions that transitions to a converged NS solution in continuum regions.

The MPC method is thus entirely implemented by Algorithms [1-15]. This general implementation was used to produce all MPC results shown in this thesis. The syntax of these hybrid functions is specific to both DSMC and NS module source codes and data structures. Thus the developer of such a modular particle-continuum code must be familiar with the original source codes and understand how to access information from, and update information within, each data structure.

7.4 Parallel Implementation

Although the MPC method has not yet been implemented for parallel computations, the modular implementation is expected to be advantageous when parallelizing the MPC code. In the same way that the data structures of the DSMC and NS modules differed, the optimal parallel implementation of a DSMC code is also quite different compared to that for a NS solver. MONACO partitions the mesh such that the number of particles is the same on each processor. However, for LeMANS, since the computational time required to update each cell is identical, the NS mesh is partitioned such that the number of cells is the same on each processor. At the same time, it is very important to realize that in the loosely-coupled MPC numerical cycle, information exchange between DSMC and NS meshes occurs infrequently. This means that there is little reason why a NS cell must be stored on the same processor as the corresponding refined DSMC cells contained within it. Conversely, if information was exchanged at each timestep then it would be very important that corresponding cells be located on the same processor. Without this restriction, both meshes are effectively able to be partitioned in the most efficient way for their respective numerical methods. Thus both hybrid DSMC and NS meshes could be partitioned in the same manner used by each module's original source code.

The modular implementation then has the significant advantage that the majority of the parallel implementation used in both MONACO and LeMANS can be used un-modified.

The DSMC mesh would thus be partitioned such that the number of particles on each processor is the same, which should automatically account for near-equilibrium regions of the DSMC mesh that contain no particles. As interface locations evolve, the DSMC mesh will have to be re-partitioned periodically. However, this is already a standard procedure in MONACO and within most DSMC codes. LeMANS (along with most parallel NS solvers), partitions the mesh just once at the beginning of each simulation and would not account for the fact that non-equilibrium regions are not updated by the solver. While the parallel implementation of the NS module could certainly be modified to account for this, since the time required to solve NS regions in a hybrid simulation contributes little to the overall simulation time, this may not even be necessary. The new hybrid data structures (HCdata and HPdata) would have to be partitioned, however they could simply reflect the partitioning of each module's data structure. That is, HCdata would be partitioned exactly as the NS mesh (NSdata) and HPdata would be partitioned exactly as the DSMC mesh (DSMCdata). The hybrid data structures would have to be slightly modified to contain both the processor number and index number of the corresponding cells. Only when information is transferred between DSMC and NS regions, would additional communication between processors be required. This additional communication should be negligible compared with the communication between processors required during each timestep of both DSMC and NS portions of the hybrid simulation.

Thus, although the MPC method is presently capable of serial computations only, it is expected that the modular implementation of the MPC method will be

highly beneficial when implementing a parallel version of the code. Certainly there are many subtle aspects of such a parallel implementation. However, the general implementation described above should provide parallel efficiencies similar to those obtained by MONACO and LeMANS individually. After implementing the MPC method to run on large parallel architectures, the actual parallel efficiency achieved by the hybrid code should be investigated and compared with the parallel efficiency of DSMC and NS modules individually.

CHAPTER VIII

Conclusions

In this concluding chapter, a summary of the dissertation including specific contributions to the field as well as recommendations for future research are presented.

8.1 Summary

Chapter I described the molecular nature of dilute gases and introduced fundamental spatial and temporal scales; the mean-free-path λ and the mean-free-time τ_c, and the Knudsen number. The nature of thermochemical non-equilibrium inherent in hypersonic flows was described and the success of accurately simulating a wide range of such flows using a continuum formulation was highlighted. However, localized regions of translational non-equilibrium within hypersonic flows are not accurately modelled with a continuum description. Chapter I cited numerous examples from the literature where a particle description is necessary to accurately predict the aerothermodynamic environment experienced by modern hypersonic atmospheric entry vehicles. Examples included the high-altitude portion of various spacecraft trajectories, the after-body flowfields of blunt-body vehicles, and vehicle configurations such as large ballutes being towed by thin tethers which contain a wide range of spatial and temporal scales.

Chapter II first detailed how gases are modelled from a kinetic/particle perspective according to the Boltzmann equation. The difficulties involved in obtaining a deterministic solution to this equation, namely the high dimensionality of the problem, were outlined. The theory used to compute moments of the Boltzmann equation that effectively average over molecular properties and result in a macroscopic description of a gas was briefly outlined. Following this, the continuum Navier-Stokes (NS) equations and inherent continuum assumptions were detailed and the direct simulation Monte Carlo (DSMC) particle method was described. Chapter II then discussed the limitations of both DSMC and NS methods. The NS equations are inaccurate for regions where the local Knudsen number is high. Although the DSMC method is valid for all Knudsen numbers, the method becomes computationally prohibitive as the flow nears continuum conditions. After describing both approaches, Chapter II then provided a detailed literature review of hybrid kinetic-continuum research conducted over the past decade divided into the two main categories, "unified" methods and "coupled" particle-continuum methods.

In Chapter III, the various components required by the hybrid particle-continuum method described in this thesis were completely detailed. Such components included calculation of the gradient length Knudsen number (Kn_{GL}) as an indicator of continuum breakdown and mesh refinement procedures for particle regions. Chapter III also described state-based information transfer including particle generation (using a Chapman-Enskog distribution) and the sub-relaxation averaging technique, which combines both spatial and temporal averaging to control statistical scatter.

In Chapter IV, the above-mentioned components were coupled using a simple tightly-coupled hybrid numerical cycle. The resulting modular particle-continuum (MPC) method was applied to 1D normal shock waves in argon and diatomic nitro-

gen gas. This 1D study first revealed that the hybrid solution within DSMC regions (when initialized to a continuum solution) does in fact progress towards the correct non-equilibrium solution in a robust manner. This progression was found to occur despite the influence from moderately incorrect states and velocity distributions prescribed within DSMC boundary cells by the continuum NS regions. Quantitative measures of the statistical scatter in macroscopic properties computed in DSMC regions were presented. The sub-relaxation average proved a very effective tool capable of reducing the statistical scatter while minimizing the temporal lag in the average. Using these important findings, hybrid particle-continuum simulation results of 1D normal shock waves were presented in Chapter IV for a range of shock Mach numbers in both argon and diatomic nitrogen gas. Hybrid solutions improved upon initial NS solutions and were found to be in excellent agreement with full DSMC solutions and with experimental measurements. Since, in the hybrid simulations, particles were eliminated in both pre and post-shock regions, the hybrid solutions of the shock waves were obtained approximately two times faster than full DSMC simulations.

In Chapter V the MPC algorithm was re-evaluated and extended to simulate 2D and axi-symmetric flows. The final MPC numerical cycle was completely detailed within Chapter V through its application to hypersonic blunt-body flow problems. Baseline MPC simulations of Mach 3, 6, and 12 flow of nitrogen about a 2D cylinder geometry (where the global Knudsen number was 0.01) were performed. In addition, an axi-symmetric flow about a planetary probe geometry at a nominal Mach number of 20 and a global Knudsen number of 0.001 was performed using the MPC method. For these hypersonic blunt-body flows, the gradient-length Knudsen number (using a cutoff value of 0.05) was shown to accurately predict regions of continuum breakdown. This parameter was also shown to provide an accurate prediction for the

magnitude of the error in a continuum solution compared to a DSMC solution. Close monitoring of the progression of hybrid solutions in Chapter V revealed that particle-continuum interfaces must be located within regions of near equilibrium flow before any information is transferred to continuum regions. A loosely-coupled, numerical cycle based on this conclusion was able to ensure that hybrid solutions progressed towards the correct physical result in a robust and predictable manner. Analysis of the local velocity distributions revealed that there are localized regions of non-equilibrium flow as well as large regions of continuum flow within these blunt-body flows. Non-equilibrium effects such as slip flow and temperature jump near surfaces, thermal non-equilibrium, variations in wake-vortex size, and higher temperatures within the wake-vortex were all captured with a high level of accuracy by the MPC method. The surface properties and local velocity distributions predicted by the MPC method were found to agree with full DSMC results with a high level of accuracy for the baseline cylinder flows. Application to the more complex axi-symmetric flow of a planetary probe geometry also produced excellent results compared with full DSMC simulation. Finally, the mesh refinement and implicit time-marching utilized by the MPC method allowed for complete spatial and temporal scale decoupling. Simulating fewer computational cells for fewer timesteps (than required by the DSMC method) allowed for very efficient simulation of continuum regions. As a result, the MPC method was shown to accurately reproduce full DSMC results in simulation times that are 2.7, 2.1, and 3.3 times faster for Mach 3, 6, and 12 cylinder flows respectively. For a near continuum flow of the planetary probe, the MPC method reproduced full DSMC flow field and surface property results approximately 12.5 times faster while requiring only 20% of the memory.

In Chapter VI the MPC algorithm was applied to flow over an axi-symmetric

hollow cylinder-flare geometry at a nominal Mach number of 12 and a global Knudsen number of 0.0008. Several new essential capabilities of the loosely-coupled MPC method were demonstrated by hybrid simulations of this hypersonic interaction flow. Such capabilities included the ability of MPC interfaces to track moving flow features, and the ability of the MPC cycle to significantly alter NS boundary conditions and shift the solution in NS-regions of the simulation. Chapter VI showed that MPC simulations reproduced the velocity slip, temperature jump, thermal non-equilibrium, and surface properties predicted by full DSMC near the leading edge. MPC simulation also successfully reduced the over-sized separation region predicted by the NS equations to the size predicted by DSMC and experimental results. With the exception of the flare region, MPC simulation accurately reproduced DSMC results approximately 1.4 times faster than full DSMC simulation while using 80% of the memory. Along the flare, the gradient-length Knudsen number predicted a thin layer of continuum breakdown next to the surface. The transfer of information across such a thin DSMC region within a hypersonic boundary layer was shown to lead to a 20-30% over-prediction in heat transfer along the flare. When the NS equations were used to simulate the entire flare region downstream of re-attachment, the accuracy of the resulting MPC heat transfer profile improves and the hybrid solution is obtained approximately 10 times faster using only 15% of the memory. Finally, in Chapter VI it was concluded that the hollow cylinder-flare problem is not practically suited for a hybrid simulation due to the the fine resolution required to solve the NS equations and the possibility of efficient full DSMC simulation (utilizing variable timesteps and subcells).

In Chapter VII, all aspects of the modular implementation used for the loosely-coupled MPC numerical cycle were completely detailed. The original source code

of both DSMC and NS modules is incorporated within the hybrid code virtually un-modified. As a result, only 15 new hybrid functions are required to implement the MPC method. Additional hybrid information is stored in a separate hybrid data structure enabling both DSMC and NS modules' data structures to remain un-altered. Chapter VII discussed how such a modular implementation reduces the development time of a hybrid particle-continuum code, allows for updates in either module to be rapidly incorporated within the hybrid code, and ensures the most accurate and efficient numerical tools are used in both non-equilibrium and continuum regions of the simulation.

8.2 Contributions

In general, the various components utilized by the MPC numerical method (detailed in Chapter III) were originally developed by previous researchers as part of their own efforts towards developing a hybrid particle-continuum method for a particular application. Depending on whether the flow is high speed or low speed, steady or unsteady, procedures used for state-based and flux-based coupling, particle generation, mesh refinement, computing continuum breakdown, and various averaging procedures have all been combined in various ways. The MPC method has adopted these existing components from the work of previous researchers with the exception of the mesh refinement algorithm and the use of the sub-relaxation average, which are unique to the MPC method. Although many of these hybrid components are based on prior work, this dissertation adds five distinct contributions to the field of hybrid particle-continuum numerical simulation of hypersonic flows. These contributions are contained in Refs. [77, 78, 79, 80, 81, 82, 83]

1. This dissertation clearly demonstrates that a loosely-coupled hybrid DSMC-NS

algorithm is very effective specifically for the simulation of hypersonic steady-state flows. Loose-coupling allows for complete spatial and temporal scale decoupling, which accounts for significant gains in computational efficiency. At the same time, a loosely-coupled algorithm allows a sub-relaxation average (a mixture of spatial and temporal averaging) to effectively control statistical fluctuations in particle regions. For the first time in the literature, this dissertation demonstrates a significant (order-of-magnitude) increase in computational efficiency achieved by hybrid particle-continuum simulation of mainly-continuum hypersonic flows that contain localized regions of non-equilibrium.

2. Unique to this dissertation are the detailed investigations of the progression of each hybrid solution, which clarify how such a loosely-coupled algorithm successfully reproduces full DSMC results. It is found that hybrid particle-continuum interfaces must be located in regions of near-equilibrium before any information is transferred into the continuum domains. Otherwise, a non-equilibrium boundary condition will introduce error into the continuum regions, which may accumulate within the hybrid simulation.

3. This dissertation presents the most detailed, quantitative comparisons between full NS, full DSMC, and hybrid solutions currently seen in the literature. Comparisons are made for a variety of flow variables in all regions of interest for each simulation. Results are overlayed in all Figures such that the relative discrepancy between all simulations is readily determined.

4. For the first time in the literature, this dissertation investigates the surface properties (heat transfer, shear-stress, and surface pressure) as well as the local velocity distribution functions predicted by a hybrid particle-continuum

method. Despite using particles to simulate only certain regions of the flowfield, the MPC method successfully reproduces full DSMC velocity distribution functions as well as full DSMC and experimental results for surface properties.

5. This dissertation presents a novel modular implementation procedure for the development of a hybrid particle-continuum method. The modular implementation is able to incorporate existing state-of-the-art DSMC and NS codes as modules (un-modified) within the hybrid code. Such an implementation is shown to require minimal additional source code to implement the hybrid procedures and numerical cycle. The modular implementation has the potential to reduce the development time of a hybrid particle-continuum code and ensure the most accurate and efficient models are utilized in both particle and continuum regions.

In conclusion, application of the modular particle-continuum (MPC) method to baseline 2D cylinder flows, a hollow cylinder-flare, and a planetary probe problem, clearly show that a loosely coupled hybrid DSMC-NS approach is very promising for the solution of hypersonic steady-state flows where large variations in local Knudsen number are seen. The MPC method is shown to be particulary promising for blunt-body and leading edge flows. Complex shock interaction flows are recommended for hybrid simulation only if the flow is part of a much larger, full vehicle simulation.

8.3 Future Direction

The MPC algorithm along with its modular implementation provide a solid basis for further development of the hybrid particle-continuum numerical method. The best applications for a hybrid method are very high-speed, mainly continuum, multi-dimensional flows where DSMC simulation of localized non-equilibrium regions is

necessary however full DSMC simulation becomes computationally prohibitive. By their very nature, such flows involve vibrational energy mode excitation, chemical reactions, and complex 3D geometries; all of which must be addressed in future development of the MPC method.

8.3.1 Vibrational energy and chemical reactions

To begin DSMC and NS simulations including both vibrational energy exchange and chemical reactions for near-equilibrium flows (where both methods are expected to produce consistent solutions) must be compared. DSMC and NS methods model these physical processes in substantially different ways and may not produce equivalent results as expected [84]. As examples, both the vibrational energy exchange and backwards chemical reaction rates are modelled in different ways in DSMC and NS simulations. Subtleties such as this must be analyzed and remedied in order to ensure DSMC and NS produce not only equivalent results in near-equilibrium regions, but the correct physical result. If both methods do not agree in overlap regions, a hybrid method that attempts to use NS in near-equilibrium regions cannot possibly reproduce full DSMC results.

The next step is to actually implement the physical routines for chemical reactions and vibrational energy transfer within the hybrid code. It is important to realize that the modular implementation greatly simplifies this task. Both DSMC and NS modules (MONACO and LeMANS) already contain these physical models. Thus only the averaging of these new quantities and generating different particle species with vibrational energy requires further research and testing. The largest obstacle besides ensuring consistency between DSMC and NS methods, will likely be the averaging of trace species in a chemically reacting flow. The scatter in trace species

may be far too large to pass into the continuum domain as a NS boundary condition and the scatter may be difficult to control using simple averaging techniques. Trace species particle-weighting or advanced averaging procedures may be required.

8.3.2 Mesh refinement

The mesh refinement algorithm utilized by the MPC method successfully demonstrates the efficiency gained by spatial scale separation, however the current algorithm is quite limited. Adaptive mesh refinement and highly variable cell sizes can provide substantial efficiency gains for a DSMC code. Since a DSMC mesh is only used to organize particle locations and is not used to calculate derivatives or interpolate data, an unstructured DSMC mesh can be locally refined to the mean-free-path for each cell. The mesh refinement algorithm used by the MPC method uses a structured mesh for both continuum and particle regions, which is not optimal in particle regions. Ideally, both DSMC and NS regions of the mesh could be completely optimized for their respective numerical method and have no geometrical alignment at all. This would require slightly more sophisticated averaging procedures and particle generation procedures since corresponding DSMC and NS cells would no longer occupy the same physical space. However, this added complexity is likely to be negligible compared with the efficiency gained by using optimized meshes in both DSMC and NS regions.

8.3.3 Parallelization and extension for 3D simulations

For hypersonic non-equilibrium flows that involve large variations in λ, since the local cell size must scale with λ in each dimension, 3D DSMC simulations can require a prohibitive number of cells and enormous computational resources. Such flows provide the best application for a hybrid method, but require that the numerical

algorithm be parallelized and extended for 3D flows. The MPC method has not yet been implemented for use on parallel computer systems, however a discussion of possible implementation techniques has been presented previously in Chapter VII in context with the modular implementation. The modular implementation of the MPC method will again prove advantageous when extending the method for 3D flows. Both modules, MONACO and LeMANS, are already implemented for 3D flows. Thus only the few additional hybrid functions require modification. The most difficult modifications will likely involve the mesh refinement routines and the fact that interfaces will form more complex surfaces in a 3D flow. These modifications may be tedious to implement however no significant physical issues are expected to be introduced by extending the MPC method for 3D flows.

8.3.4 Continuum breakdown parameter

The accurate detection of continuum breakdown remains one of the fundamental limitations of any hybrid method. It is arguable that further research into continuum breakdown may be best performed through use of a hybrid method. Continuum regions of a flowfield may be entirely shifted within a full DSMC simulation due to velocity slip, temperature jump, or frozen thermal non-equilibrium. Thus, comparison of full DSMC and full NS solutions may indicate large error in the majority of the flow due to this shift and thus indicate that the majority of the flow is in non-equilibrium. For this reason, investigation of continuum breakdown using only DSMC and NS methods is often difficult. Within a hybrid simulation, variation of the continuum breakdown parameter is able to directly determine its effectiveness for predicting continuum breakdown by comparison of the hybrid solution to a full DSMC solution. In addition, since the MPC method contains an overlap region

where both DSMC and NS methods are used and are expected to agree, the potential exists to utilize this overlap region to properly position the interfaces. That is, instead of relying on the continuum breakdown parameter, if DSMC and NS solutions are in agreement within the overlap region then the interface is properly positioned. In contrast, if there is a noticeable discrepancy between the solutions in the overlap region, this would indicate that the interface locations need to be adjusted. However, in regard to the use of a continuum breakdown parameter, it is important to note that as long as hybrid particle-continuum interfaces are placed conservatively (well within near-equilibrium regions), their exact placement will only affect the efficiency of the hybrid method and not the accuracy.

APPENDIX

APPENDIX A

Algorithms for Hybrid Functions

Algorithm 4 `allocate_hybrid_data()`

function ALLOCATE_HYBRID_DATA(NSdata)
▷ Cell types
HCdata[cells].dsmc
HCdata[cells].cfd
HCdata[cells].bc
▷ Overlap cells have (HCdata[i].dsmc=1 AND HCdata[i].cfd=1)
▷ DSMC boundary cells have (HCdata[i].cfd=1 AND HCdata[i].bc=1)
▷ NS boundary cells have (HCdata[i].dsmc=1 AND HCdata[i].bc=1)

▷ Refinement criteria
HCdata[cells].refine[1]
HCdata[cells].refine[2]

▷ Links to DSMC mesh
HCdata[cells].Pindex_B = pointer to first (Beginning) refined cell
HCdata[cells].Pindex_E = pointer to last (End) refined cell

▷ Sub-relaxation variables
HCdata[cells].avgQ[new][variables]
HCdata[cells].avgQ[prev][variables]
HCdata[cells].avgQ[old][variables]
▷ All sub-relax avgs. are initialized to NSdata[cells].Q[variables]
end function

Algorithm 5 `setup_hybrid_domains()`

function SETUP_HYBRID_DOMAINS(NSdata, HCdata, $\mathbf{n}_{\text{overlap}}$, $\mathbf{n}_{\text{DSMC\_BCs}}$, $\mathbf{Br}_{\text{cutoff}}$)

KnGL[NScells][variables] = `calculate_KnGL(NSdata)` ▷ see Algorithm [7]

for all (NScells, i) **do**
KnGL_max = max(KnGL[i][variables]) ▷ See Eq. 3.3

if (**ignore_shock_flag** AND KnGL[i][pressure]>1.0) **then**
KnGL_max=0.0 ▷ Ignore continuum breakdown inside shocks
end if

if (KnGL_max > $\mathbf{Br}_{\text{cutoff}}$) **then** ▷ Set cell types
HCdata[i].dsmc = 1
HCdata[i].cfd = 0
else
HCdata[i].dsmc = 0
HCdata[i].cfd = 1
end if
end for

`smooth_interfaces(HCdata)` ▷ see Algorithm [16]

▷ Set layer of NS boundary cells and first layer of overlap cells
for all (NS cells, i) **do**
for all (neighbor cells, n) **do**
▷ If a DSMC cell and neighbor is a NS cell
if (HCdata[i].dsmc=1 AND HCdata[n].cfd=1) **then**
HCdata[i].bc=1 ▷ Create a boundary cell
HCdata[n].dsmc=1 ▷ Create first overlap cell
end if
end for
end for
▷ Set the remaining overlap cells
for ($\mathbf{n}_{\text{overlap}} - 1$ layers) **do**
for all (existing overlap cells, i) **do**
for all (neighbor cells, n) **do**
if (HCdata[n].cfd=1 AND HCdata[n].dsmc=0) **then**
HCdata[n].dsmc=1
end if
end for
end for
end for

▷ Continued in Algorithm [6]

Algorithm 6 `setup_hybrid_domains()` ...continued from Algorithm [5]

```
    ▷ Set first layer of DSMC boundary cells
    for all (overlap cells, i) do
        for all (neighbor cells, n) do
            if (HCdata[n].cfd=1 AND HCdata[n].dsmc=0) then
                HCdata[n].bc=1
            end if
        end for
    end for
    ▷ Set the remaining DSMC boundary cells
    for (n_DSMC_BCs − 1 layers) do
        for all (existing DSMC boundary cells, i) do
            for all (neighbor cells, n) do
                if (HCdata[n].cfd=1 AND HCdata[n].dsmc=0) then
                    HCdata[n].bc=1
                end if
            end for
        end for                                   ▷ now complete next layer
    end for
    ▷ HCdata[i].dsmc, HCdata[i].cfd, and HCdata[i].bc, are now set for all cells
end function
```

Algorithm 7 `calculate_KnGL()`

```
function CALCULATE_KNGL(NSdata)
    for all (NS cells, i) do
        ▷ Calculating gradients is a standard NS function
        dQ[x, y][variable] = calculate_gradients()
        λ = VHS value from Eq. 3.2 using NSdata[i]

        for all (variables, v) do
            ▷ Calculate KnGL using Eq. 3.1
            KnGL[i][v] = | λ√(dQ[x][v]² + dQ[y][v]²) / NSdata[i].Q[v] |
        end for
    end for
    return KnGL[NScells][variables]
end function
```

Algorithm 8 `determine_mesh_refinement()`

function `DETERMINE_MESH_REFINEMENT()`(NSdata)

KnGL[NScells][variables] = `calculate_KnGL(NSdata)` ▷ See Algorithm [7]
for all (NS cells, i) **do**
▷ Record variable of maximum gradient
determine which variable v gives max KnGL$[i][v]$
$max\_var[i] = v$
end for

for all (NS cells, i) **do**
▷ Calculating gradients is a standard NS function
$dQ[x, y][variable]$ = `calculate_gradients()`
λ = VHS value from Eq. 3.2 using NSdata$[i]$

▷ Calculate unit gradient vector
$dQ\_x = dQ[x][max\_var[i]]$
$dQ\_y = dQ[y][max\_var[i]]$
$|dQ| = \sqrt{dQ\_x^2 + dQ\_y^2}$
$dQ\_\hat{x} = dQ\_x \ / \ |dQ|$
$dQ\_\hat{y} = dQ\_y \ / \ |dQ|$

for (2 cell faces, f) **do**
face normal unit vectors $\hat{n}_x$ and $\hat{n}_y$ available from NSdata$[i]$
factor$[f]$ = $\mathbf{F}[\ 3 \times |\hat{n}_x dQ\_\hat{x} + \hat{n}_y dQ\_\hat{y}| + 1\ ]$

▷ The face length l_f is available from NSdata$[i]$
▷ Calculate required refinement using Eq. 3.4
HCdata$[i]$.refine$[f]$ = round[max$(l_f/(\lambda \times$ factor$[f])$, 1)]
end for
end for
▷ Required refinement now stored for each NScell i and face f
end function

Algorithm 9 `create_DSMC_mesh()`

```
function CREATE_DSMC_MESH()(NSdata, HCdata)
    ▷ Determine required number of refined DSMC cells
    for all (NScells, i) do
        DSMCcells += HCdata[i].refine[1] × HCdata[i].refine[2]
    end for

    ▷ Allocate for DSMC data structure, DSMCdata[DSMCcells]
    ▷ Create hybrid links between NSdata and DSMCdata
    index = 0
    for all (NScells, i) do
        n_subcells = HCdata[i].refine[1] × HCdata[i].refine[2]
        ▷ Link NS cell (i) to (B)eginning and (E)nding DSMC subcell
        HCdata[i].Pindex_B = index
        HCdata[i].Pindex_E = index + n_subcells − 1
        while (index < HCdata[i].Pindex_E) do
            ▷ Link DSMC cell (index) to NS cell (i), by creating HPdata[DSMCcells]
            HPdata[index]=i
            ▷ Allocate memory for one DSMCcell
            DSMCdata[index]=empty DSMCcell
            index++
        end while
    end for

    for all (NScells, i) do
        ▷ Node co-ordinates for each NScell are available from NSdata[i]
        ▷ Determine the node co-ordinates for all DSMC subcells
        index=HCdata[i].Pindex_B
        while (index < HCdata[i].Pindex_E) do
            DSMCdata[index].nodes = assign node co-ordinates
            index++
        end while
    end for
    for all (NScells, i) do
        ▷ Determine neighbors (cell-connectivity) for DSMCdata[DSMCcells]
        index=HCdata[i].Pindex_B
        while (index < HCdata[i].Pindex_E) do
            DSMCdata[index].neighbors = assign indices of neighbors
            index++
        end while
    end for
    ▷ Now have list of DSMCcells, with node co-ordinates, and neighbors
    ▷ Use this to generate the mesh format required by the DSMC module
end function
```

Algorithm 10 `generate_particles()`

```
function GENERATE_PARTICLES(NSdata, DSMCdata, HCdata)
    new_cell_count = 0
    for all (NS cells, i) do
        if (HCdata[i].dsmc=1) then
            ▷ Check if it is a newly created and empty DSMC cell
            empty = 1
            index = HCdata[i].Pindex_B
            while (index < HCdata[i].Pindex_E) do
                if (DSMCdata[index].particles> 0) then
                    empty = 0
                end if
            end while

            if (empty) then
                index = HCdata[i].Pindex_B
                while (index < HCdata[i].Pindex_E) do
                    DSMCdata[index].particles
                    = generate_objects(DSMCdata[index],NSdata[i])
                                                        ▷ see Algorithm [12]
                    new_cell_count ++
                end while
            end if
        end if
    end for
    return new_cell_count
end function
```

Algorithm 11 `generate_DSMC_bdry_particles()`

```
function GENERATE_DSMC_BDRY_PARTICLES(NSdata, DSMCdata, HCdata)
    for all (DSMC cells, index) do
        ▷ Determine corresponding NScell i
        i = HPdata[index]

        ▷ If it is a DSMC boundary cell
        if (HCdata[i].cfd=1 AND HCdata[i].bc=1) then
            Delete all particles from DSMCdata[index].particles
            DSMCdata[index].particles
            = generate_objects(DSMCdata[index],NSdata[i])
                                                        ▷ see Algorithm [12]
        end if
    end for
end function
```

Algorithm 12 `generate_objects()`

function GENERATE_OBJECTS(NSdata[i], DSMCdata[$index$])
▷ Determine the number of particles n
based on NSdata[i].Q[ρ] and DSMCdata[$index$].volume
▷ Compute q_i and $\tau_{i,j}$ for current cell from NSdata[i]

▷ Select thermal velocities C_x, C_y, and C_z
for these n particles using the acceptance-rejection
algorithm of Garcia and Alder (Ref. [56])

▷ Set particle velocities as
v_x = NSdata[i].Q[u] + C_x
v_y = NSdata[i].Q[v] + C_y
$v_z = C_z$

▷ Assign these velocities to DSMCdata[$index$].particles
end function

Algorithm 13 `update_sub_relax_avgs()`

```
function UPDATE_SUB_RELAX_AVGS(NSdata, DSMCdata, HCdata)
    for all (NS cells, i) do
        ▷ If an interior or DSMC boundary cell
        if (HCdata[i].dsmc=1 OR HCdata[i].bc=1) then
            index = HCdata[i].Pindex_B
            while (index < HCdata[i].Pindex_E) do
                ▷ Using DSMCdata[index].particles and Eqs. 3.8 to 3.12,
                evaluate the summations needed for spatial averages
            end while
            ▷ Determine spatial averages for this NS cell: A[variables]

            ▷ Update all the sub-relaxation average in this NS cell
            for all (variables, v) do
                HCdata[i].avgQ[new][v] =
                (1-theta)HCdata[i].avgQ[prev][v] + theta× A[v]
                ▷ Apply a correction every (1/theta) timesteps (Eqs. 3.13 and 3.14)
            end for
        end if

        ▷ If a NS boundary cell
        if (HCdata[i].dsmc=1 AND HCdata[i].bc=1) then
            ▷ Record the max change (over all variables) in the sub-relax avg
            change = max[(HCdata[i].avgQ[new][v]-NSdata[i].Q[v])/NSdata[i].Q[v]]
            if (change > PctChange) then
                interface_update_flag = 1
            end if
        end if
    end for
    ▷ All sub-relax avgs have been updated

    if (interface_update_flag=1) then
        synchronize_cells()                          ▷ see Algorithm [14]
    else
        ▷ Sub-relax avgs have been updated, but leave interface_update_flag=0
    end if

    return interface_update_flag
end function
```

Algorithm 14 `synchronize_cells()`

```
function SYNCHRONIZE_CELLS(NSdata, HCdata)
    ▷ Copy macroscopic DSMC solution to current NS mesh
    for all (NS cells, i) do
        if (HCdata[i].dsmc=1) then
            for all (variables, v) do
                NSdata[i].Q[v] = HCdata[i].avgQ[new][v]
            end for
        end if
    end for
end function
```

Algorithm 15 `set_NS_bdry_conditions()`

```
function SET_NS_BDRY_CONDITIONS(NSdata, HCdata)
    for all (NS cells, i) do
        ▷ If it is a NS boundary cell
        if (HCdata[i].dsmc=1 AND HCdata[i].bc=1) then
            for all (variables, v) do
                if (!locked) then
                    NSdata[i].Q[v] = HCdata[i].avgQ[new][v]
                else
                    NSdata[i].Q[v] = cumulative sampled averages from DSMCdata
                end if
            end for
        end if
    end for
end function
```

Algorithm 16 `smooth_interfaces()`

```
function SMOOTH_INTERFACES(HCdata)
    for all (NS cells, i) do
        value[i] = HCdata[i].dsmc
    end for

    ▷ Diffusion - smoothing function
    for (4 iterations) do
        for all (NS cells, i) do
            new_value[i] = (value[i]+∑_neighbors value[neighbor]) / (#neighbors+1)
        end for
        for all (NS cells, i) do
            value[i]=new_value[i]
        end for
    end for

    ▷ Re-assign cell types based on smooth profile
    for all (NS cells, i) do
        if (value[i]>0.5) then
            HCdata[i].dsmc=1
            HCdata[i].cfd=0
        else
            HCdata[i].dsmc=0
            HCdata[i].cfd=1
        end if
    end for
end function
```

Algorithm 17 `set_DSMC_cell_weights()`

```
function SET_DSMC_CELL_WEIGHTS(DSMCdata)
    for all (DSMC cells, index) do
        r = distance from cell centroid to axis of symmetry
        r_min = record minimum value of r in the mesh
        r_max = record maximum value of r in the mesh
    end for
    ▷ User defined cloning factor, CF = desired factor increase in cell weights
    CF = 2.5                                                    ▷ User defined value
    max_ratio = r_max / r_min
    num_bands = (max_ratio) ^ 1/CF
    for all (DSMC cells, index) do
        ratio = round_to_integer( DSMCdata[index].y-centroid / r_min )
        weight[index]=1
        for (b=1 to num_bands) do
            if (ratio> CF^b) then
                weight[index] = CF^b
            end if
        end for
    end for
    return weight[cells]
end function
```

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Bird, G. A., *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Oxford University Press, New York, 1994.

[2] Moss, J. N. and Price, J. M., "Survey of Blunt Body Flows Including Wakes at Hypersonic Low-Density Conditions," *Journal of Thermophysics and Heat Transfer*, Vol. 11, No. 3, 1997, pp. 321–329.

[3] Wright, M. J., Milos, F. S., and Tran, P., "Afterbody Aeroheating Flight Data for Planetary Probe Thermal Protection System Design," *Journal of Spacecraft and Rockets*, Vol. 43, No. 5, 2006, pp. 929–943.

[4] Wright, M. J., Prabhu, D. K., and Martinez, E. R., "Analysis of Apollo Command Module Afterbody Heating Part I: AS-202," *Journal of Thermophysics and Heat Transfer*, Vol. 20, No. 1, 2006, pp. 16–30.

[5] Edquist, K. T., "Afterbody Heating Predictions for a Mars Science Laboratory Entry Vehicle," *AIAA Paper 05-4817*, June 2005, presented at the 38th AIAA ThermophysicsConference, Toronto, ON, Canada.

[6] Moss, J. N., Glass, C. E., and Greene, F. A., "DSMC Simulations of Apollo Capsule Aerodynamics for Hypersonic Rarefied Conditions," *AIAA Paper 06-3577*, June 2006, presented at the 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, CA.

[7] Moss, J. N., Boyles, K. A., and Greene, F. A., "Orion Aerodynamics for Hypersonic Free Molecular to Continuum Conditions," *AIAA Paper 06-8081*, Nov. 2006, presented at the 14th AIAA/AHI Space Planes and Hypersonic Systems and Technologies Conference, Canberra, Australia.

[8] Wilmoth, R. G., Rault, D. F., Cheatwood, F. M., Engelund, W. C., and Shane, R. W., "Aerothremodynamic Predictions of Mars Global Surveyor," *Journal of Spacecraft and Rockets*, Vol. 36, No. 3, 1999, pp. 314–322.

[9] Takashima, N. and Wilmoth, R. G., "Aerodynamics of Mars Odyssey," *AIAA Paper 02-4809*, Aug. 2002, presented at the AIAA Atmospheric Flight Mechanics Conference and Exhibit, Monterey, CA.

[10] Moss, J. N., "Direct Simulation Monte Carlo Simulations of Ballute Aerothermodynamics Under Hypersonic Rarefied Conditions," *Journal of Spacecraft and Rockets*, Vol. 44, No. 2, 2007, pp. 289–298.

[11] Gnoffo, P. A., Buck, G., Moss, J. N., Nielsen, E., Berger, K., Jones, W. T., and Rudavsky, R., "Aerothermodynamic Analyses of Towed Ballutes," *AIAA Paper 06-3771*, June 2006, presented at the 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, CA.

[12] Reese, J. M., Gallis, M. A., and Lockerby, D. A., "New Directions in Fluid Dynamics: Non-Equilibrium Aerodynamic and Microsystem Flows," *Phil. Trans. R. Soc. Lond. A.*, Vol. 361, 2003, pp. 2967–2988.

[13] Seokkwan, Y., Gnoffo, P. A., White, J. A., and Thomas, J. L., "Computational Challenges in Hypersonic Flow Simulations," *AIAA Paper 07-4265*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[14] Gombosi, T. I., *Gaskinetic Theory*, Cambridge University Press, New York, 1994.

[15] Agarwal, R., Chen, R., and Cheremisin, F., "Computation of Hypersonic Shock Wave Flows of Diatomic Gases Using the Generalized Boltzmann Equation," *AIAA Paper 07-4541*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[16] Chen, R., Agarwal, R., and Cheremisin, F., "Computation of Hypersonic Flow of a Diatomic Gas in Rotational Nonequilibrium Past a Blunt Body Using the Generalized Boltzmann Equation," *AIAA Paper 07-4550*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[17] Larsen, P. S. and Borgnakke, C., "Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture," *Journal of Computational Physics*, Vol. 18, 1975, pp. 405–420.

[18] Harvey, J. K. and Gallis, M. A., "Review of Code Validation Studies in High-Speed Low-Density Flows," *Journal of Spacecraft and Rockets*, Vol. 37, No. 1, 2000, pp. 8–20.

[19] Kolobov, V. I., Bayyuk, S. A., Arslanbekov, R. R., Aristov, V. V., Frolova, A. A., and Zabelok, S. A., "Construction of a Unified Continuum/Kinetic Solver for Aerodynamic Problems," *Journal of Spacecraft and Rockets*, Vol. 42, No. 4, 2005, pp. 598–606.

[20] Kolobov, V. I., Arslanbekov, R. R., Aristov, V. V., Frolova, A. A., and Zabelok, S. A., "Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement," *Journal of Computational Physics*, Vol. 223, No. 2, 2007, pp. 589–608.

[21] Le Tallec, P. and Mallinger, F., "Coupling Boltzmann and Navier-Stokes Equations by Half Fluxes," *Journal of Computational Physics*, Vol. 136, 1997, pp. 51–67.

[22] Tiwari, S. and Klar, A., "An Adaptive Domain Decomposition Procedure for Boltzmann and Euler Equations," *Journal of Computational and Applied Mathematics*, Vol. 90, 1998, pp. 223–237.

[23] Burt, J. and Boyd, I. D., "A Low Diffusion Particle Method for Simulating Near-Equilibrium Flows," *AIAA Paper 07-5035*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[24] Suzuki, Y. and van Leer, B., "Application of the 10-Moment Model to MEMS Flows," *AIAA Paper 05-1398*, Jan. 2005, presented at the 43rd AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[25] McDonald, J. and Groth, C., "Numerical Modeling of Micron-Scale Flows Using the Gaussian Moment Closure," *AIAA Paper 05-5035*, June 2005, presented at the 35th AIAA Fluid Dynamics Conference and Exhibit, Toronto, ON.

[26] Pareschi, L. and Caflisch, R. E., "An Implicit Monte Carlo Method for Rarefied Gas Dynamics," *Journal of Computational Physics*, Vol. 154, No. 1, 1999, pp. 90–116.

[27] Wijesinghe, H. S. and Hadjiconstantinou, N. G., "A Discussion of Hybrid Atomistic-Continuum Methods for Multiscale Hydrodynamics," *International Journal for Multiscale Computational Engineering*, Vol. 2, 2004.

[28] Wadsworth, D. C. and Erwin, D. A., "One-Dimensional Hybrid Continuum/Particle Simulation Approach for Rarefied Hypersonic Flows," *AIAA Paper 90-1690*, June 1990, presented at the 5th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, Seattle, WA.

[29] Wadsworth, D. C. and Erwin, D. A., "One-Dimensional Hybrid Continuum/Particle Approach for Rarefied Flows," *AIAA Paper 92-2975*, July 1992, presented at the 23rd AIAA Plasmadynamics and Lasers Conference, Nashville, TN.

[30] Hash, D. B. and Hassan, H. A., "Assessment of Schemes for Coupling Monte Carlo and Navier-Stokes Solution Methods," *Journal of Thermophysics and Heat Transfer*, Vol. 10, No. 2, 1996, pp. 242–249.

[31] Hash, D. B. and Hassan, H. A., "A Decoupled DSMC/Navier-Stokes Analysis of a Transitional Flow Experiment," *AIAA Paper 96-0353*, Jan. 1996, presented at the 34th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[32] Wijesinghe, H. S., Hornung, R. D., Garcia, A. L., and Hadjiconstantinou, N. G., "Three-dimensional Hybrid Continuum-Atomistic Simulations For Multiscale Hydrodynamics," *Journal of Fluids Engineering*, Vol. 126, 2004, pp. 768–777.

[33] Hadjiconstantinou, N. G., Garcia, A. L., Bazant, M. Z., and He, G., "Statistical Error in Particle Simulations of Hydrodynamic Phenomena," *Journal of Computational Physics*, Vol. 187, 2003, pp. 274–297.

[34] Hash, D. B. and Hassan, H. A., "Two-Dimensional Coupling Issues of Hybrid DSMC / Navier-Stokes Solvers," *AIAA Paper 97-2507*, June 1997, presented at the 32nd AIAA Thermophysics Conference, Atlanta, GA.

[35] Roveda, R., Goldstein, D. B., and Varghese, P. L., "Hybrid Euler/Particle Approach for Continuum/Rarefied Flows," *Journal of Spacecraft and Rockets*, Vol. 35, No. 3, 1998, pp. 258–265.

[36] Roveda, R., Goldstein, D. B., and Varghese, P. L., "Hybrid Euler/Direct Simulation Monte Carlo Calculation of Unsteady Slit Flow," *Journal of Spacecraft and Rockets*, Vol. 37, No. 6, 2000, pp. 753–760.

[37] Sun, Q. and Boyd, I. D., "A Direct Simulation Method for Subsonic, Micro-Scale Gas Flows," *Journal of Computational Physics*, Vol. 179, 2002, pp. 400–425.

[38] Wang, W. L. and Boyd, I. D., "Hybrid DSMC-CFD Simulations of Hypersonic Flow over Sharp and Blunted Bodies," *AIAA Paper 03-3644*, 2003, presented at the 36th AIAA Thermophysics Conference, Orlanda, Fl.

[39] Wang, W. L. and Boyd, I. D., "A New Energy Flux Model in the DSMC-IP Method for Nonequilibrium Flows," *AIAA Paper 03-3774*, 2003, presented at the 36th AIAA Thermophysics Conference, Orlanda, Fl.

[40] Wu, J. S., Lian, Y. Y., Cheng, G., Koomullil, R. P., and Tseng, K. C., "Development and verification of a coupled DSMC-NS scheme using unstructured mesh," *Journal of Computational Physics*, Vol. 219, 2006, pp. 579–607.

[41] Dietrich, S. and Boyd, I. D., "Scalar and Parallel Optimized Implementation of the Direct Simulation Monte Carlo Method," *Journal of Computational Physics*, Vol. 126, 1996, pp. 328–342.

[42] Boyd, I. D., "Analysis of Rotational Nonequilibrium in Standing Shock Waves of Nitrogen," *AIAA Journal*, Vol. 28, No. 11, 1990, pp. 1997–1999.

[43] Boyd, I. D., "Rotational-Translational Energy Transfer in Rarefied Nonequilibrium Flows," *Physics of Fluids A*, Vol. 2, No. 3, 1990, pp. 447–452.

[44] Scalabrin, L. C. and Boyd, I. D., "Development of an Unstructured Navier-Stokes Solver for Hypersonic Nonequilibrium Aerothermodynamics," *AIAA Paper 05-5203*, 2005, presented at the 38th AIAA Thermophysics Conference, Toronto, Ontario, Canada.

[45] Scalabrin, L. C. and Boyd, I. D., "Numerical Simulation of Weakly Ionized Hypersonic Flow for Reentry Configurations," *AIAA Paper 06-3773*, 2006, presented at the 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, CA.

[46] Scalabrin, L. C. and Boyd, I. D., "Numerical Simulations of the FIRE-II Convective and Radiative Heating Rates," *AIAA Paper 07-4044*, 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[47] MacCormack, R. W. and Candler, G. V., "The Solution of the Navier-Stokes Equations Using Gauss-Seidel Line Relaxation," *Computers and Fluids*, Vol. 17, 1989, pp. 135–150.

[48] Druguet, M. C., Candler, G. V., and Nompelis, I., "Effect of Numerics on Navier-Stokes Computations of Hypersonic Double-Cone Flows," *AIAA Journal*, Vol. 43, No. 3, 2005, pp. 616–623.

[49] Scalabrin, L. C., *Numerical Simulation of Weakly Ionized Hypersonic Flow Over Reentry Capsules*, 2007, PhD thesis, Department of Aerospace Engineering, University of Michigan.

[50] Boyd, I. D., Chen, G., and Candler, G. V., "Predicting Failure of the Continuum Fluid Equations in Transitional Hypersonic Flows," *Physics of Fluids*, Vol. 7, No. 1, 1995, pp. 210–219.

[51] Wang, W. L. and Boyd, I. D., "Predicting Continuum Breakdown in Hypersonic Viscous Flows," *Physics of Fluids*, Vol. 15, 2003, pp. 91–100.

[52] Gnoffo, P. and White, J., "Computational Aerothermodynamic Simulation Issues on Unstructured Grids," *AIAA Paper 04-2371*, June 2004, presented at the 37th AIAA Thermophysics Conference, Portland, OR.

[53] Candler, G. V., "Unstructured Grid Approaches for Accurate Aeroheating Simulations," *AIAA Paper 07-3959*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[54] Gnoffo, P. A., "Simulation of Stagnation Region Heating In Hypersonic Flow on Tetrahedral Grids," *AIAA Paper 07-3960*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[55] Sun, Q., Boyd, I. D., and Candler, G. V., "A Hybrid Continuum/Particle Approach for Modeling Subsonic, Rarefied Gas Flows," *Journal of Computational Physics*, Vol. 194, 2004, pp. 256–277.

[56] Garcia, A. L. and Alder, B. J., "Generation of the Chapman-Enskog Distribution," *Journal of Computational Physics*, Vol. 140, 1998, pp. 66–70.

[57] Sun, Q. and Boyd, I. D., "Evaluation of Macroscopic Properties in the Direct Simulation Monte Carlo Method," *Journal of Thermophysics and Heat Transfer*, Vol. 19, No. 3, 2005, pp. 329–335.

[58] Anderson, J. D., *Hypersonic and High-Temperature Gas Dynamics*, McGraw-Hill, 1989.

[59] Alsmeyer, H., "Density Profiles in Argon and Nitrogen Shock Waves Measured by the Absorption of an Electron Beam," *Journal of Fluid Mechanics*, Vol. 74, 1976, pp. 497–523.

[60] Lofthouse, A. J., Boyd, I. D., and Wright, M. J., "Effects of Continuum Breakdown on Hypersonic Aerothermodynamics," *AIAA Paper 06-993*, Jan. 2006, presented at the 44th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[61] Wilmoth, R. G., Mitcheltree, R. A., Moss, J. N., and Dogra, V. K., "Zonally Decoupled Direct Simulation Monte Carlo Solutions of Hypersonic Blunt-Body Wake Flows," *Journal of Spacecraft and Rockets*, Vol. 31, No. 6, 1994, pp. 971–979.

[62] Glass, C. E. and Gnoffo, P. A., "A 3-D Coupled CFD-DSMC Solution Method With Application to the Mars Sample Return Orbiter," *NASA TM-2000-210322*, July 2000.

[63] Glass, C. E. and Horvath, T. J., "Comparison of a 3-D CFD-DSMC Solution Methodology With a Wind Tunnel Experiment," *NASA TM-2002-211777*, Aug. 2002.

[64] Holden, M. S. and Wadhams, T. P., "Code Validation Study of Laminar Shock/Boundary Layer and Shock/Shock Interactions in Hypersonic Flow. Part A: Experimental Measurements." *AIAA Paper 01-1031*, Jan. 2001, presented at the AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[65] Harvey, J. K., Holden, M. S., and Wadhams, T. P., "Code Validation Study of Laminar Shock/Boundary Layer and Shock/Shock Interactions in Hypersonic Flow. Part B: Comparison with Navier-Stokes and DSMC Solutions." *AIAA Paper 01-1031*, Jan. 2001, presented at the AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[66] Candler, G. V., Nompelis, I., and Druguet, M. C., "Navier-Stokes Predictions of Hypersonic Double-Cone and Cylinder-Flare Flow Fields," *AIAA Paper 01-1024*, Jan. 2001, presented at the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[67] Gnoffo, P. A., "CFD Validation Studies for Hypersonic Flow Prediction," *AIAA Paper 01-1025*, Jan. 2001, presented at the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[68] Moss, J. N., "DSMC Computations for Regions of Shock/Shock and Shock/Boundary Layer Interaction," *AIAA Paper 01-1027*, Jan. 2001, presented at the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[69] Kato, H. and Tannehill, J. C., "Computation of Hypersonic Laminar Separated Flows using an Iterated PNS Algorithm," *AIAA Paper 01-1028*, Jan. 2001,

presented at the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[70] Boyd, I. D. and Wang, W.-L., "Monte Carlo Computations of Hypersonic Interacting Flows," *AIAA Paper 01-1029*, Jan. 2001, presented at the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[71] Roy, C. J., Gallis, M. A., and Payne, J. L., "DSMC and Navier-Stokes Predictions for Hypersonic Laminar Interacting Flows," *AIAA Paper 01-1030*, Jan. 2001, presented at the 39th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[72] Candler, G. V., Nompelis, I., Druguet, M. C., Holden, M. S., Wadhams, T. P., Boyd, I. D., and Wang, W. L., "CFD Validation for Hypersonic Flight: Hypersonic Double-Cone Flow Simulations," *AIAA Paper 02-0581*, Jan. 2002, presented at the 40th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[73] Moss, J. N. and Bird, G. A., "Direct Simulation Monte Carlo Simulations of Hypersonic Flows with Shock Interactions," *AIAA Journal*, Vol. 43, No. 12, 2005, pp. 2565–2573.

[74] Wang, W.-L., *A Hybrid Particle/Continuum Approach for Nonequilibrium Hypersonic Flows*, 2004, PhD thesis, Department of Aerospace Engineering, University of Michigan.

[75] Kleb, B. and Wood, B., "CFD: A Castle in the Sand?" *AIAA Paper 04-2627*, June 2004, presented at the 34th AIAA Fluid Dynamics Conference and Exhibit, Portland, OR.

[76] Kleb, B. and Wood, B., "Computational Simulations and the Scientific Method," *AIAA Paper 05-4873*, June 2005, presented at the 17th AIAA Computational Fluid Dynamics Conference, Toronto, Ontario, Canada.

[77] Schwartzentruber, T. E. and Boyd, I. D., "Detailed Analysis of a Hybrid CFD-DSMC Method for Hypersonic Non-Equilibrium Flows," *AIAA Paper 05-4829*, June 2005, presented at the 38th AIAA Thermophysics Conference, Toronto, Ontario, Canada.

[78] Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "Hybrid Particle-Continuum Simulations of Non-Equilibrium Hypersonic Blunt Body Flow Fields," *AIAA Paper 06-3602*, June 2006, presented at the 9th AIAA/ASME Joint Thermophysics and Heat Transfer Conference, San Francisco, CA.

[79] Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "Modular Implementation of a Hybrid DSMC-NS Solver for Hypersonic Non-Equilibrium Flows," *AIAA Paper 07-613*, Jan. 2007, presented at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV.

[80] Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "Hybrid Particle-Continuum Simulations of Low Knudsen Number Hypersonic Flows," *AIAA Paper 07-3829*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

[81] Schwartzentruber, T. E. and Boyd, I. D., "A hybrid particle-continuum method applied to shock waves," *Journal of Computational Physics*, Vol. 215, No. 2, 2006, pp. 402–416.

[82] Schwartzentruber, T. E., Scalabrin, L., and Boyd, I. D., "A modular particle-continuum numerical method for hypersonic non-equilibrium gas flows," *Journal of Computational Physics*, Vol. 225, No. 1, 2007, pp. 1159–1174.

[83] Schwartzentruber, T. E., Scalabrin, L. C., and Boyd, I. D., "Hybrid Particle-Continuum Simulations of Non-Equilibrium Hypersonic Blunt Body Flow Fields," *Journal of Thermophysics and Heat Transfer*, In Press, May 2007.

[84] Boyd, I. D., "Nonequilibrium Particle and Continuum Analyses of Stardust Entry for Near-Continuum Conditions," *AIAA Paper 07-4543*, June 2007, presented at the 39th AIAA Thermophysics Conference, Miami, FL.

ABSTRACT

A MODULAR PARTICLE-CONTINUUM NUMERICAL ALGORITHM FOR HYPERSONIC NON-EQUILIBRIUM FLOWS

by

Thomas Edward Schwartzentruber

Chairperson: Iain D. Boyd

Hypersonic vehicles traveling at high altitudes experience conditions ranging from rarefied to continuum flow. Even within a mostly continuum flow, there may be local regions of rarefied (or non-equilibrium) flow generated by sharp leading edges, by rapid expansion in the wake of a vehicle, as well as by strong gradients in shock waves and boundary layers. Although a particle method is necessary for accurate simulation of non-equilibrium regions, particle simulation of the entire flowfield is computationally expensive.

A modular particle-continuum (MPC) numerical algorithm for steady-state hypersonic flows is presented that solves the Navier-Stokes equations in regions of near-equilibrium and uses the direct simulation Monte Carlo (DSMC) method to simulate regions of non-equilibrium gas flow. Existing state-of-the-art DSMC and Navier-Stokes codes are loosely-coupled using state-based information transfer and

a novel modular implementation. The MPC algorithm allows for complete spatial and temporal scale decoupling whereby particle and continuum regions are simulated using different mesh densities and different sized timesteps. Particle and continuum regions are identified using a continuum breakdown parameter and automatically adapt during the hybrid simulation.

The MPC method is tested for hypersonic flow of nitrogen over a 2D cylinder at various Mach numbers where the global Knudsen number is 0.01. The MPC method is shown to reproduce full DSMC simulation results with a high degree of accuracy for flowfield quantities, surface properties, and local velocity distribution functions. The computational speedup achieved by the MPC method over full DSMC simulation ranges from 2.7 to 3.3 for the cylinder flows and is found to scale directly with the number of particles replaced by a continuum description. In addition, MPC simulations of axi-symmetric planetary probe and hollow cylinder-flare configurations with global Knudsen numbers of 0.001, are compared with full DSMC solutions and experimental data. MPC simulations of both configurations are shown to accurately reproduce DSMC results for velocity slip, temperature jump, thermal non-equilibrium, surface properties, and the extent of separated flow. For these mainly continuum flows, the MPC method is shown capable of reproducing full DSMC solutions with an order-of-magnitude decrease in both computational time and memory.