An Excursion with the Boltzmann Equation at Low Speeds: Variance-Reduced DSMC

by

Husain Ali Al-Mohssen

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2010

Author .....
Department of Mechanical Engineering
March 19, 2010

Certified by .....
N. G. Hadjiconstantinou
Associate Professor
Thesis Supervisor

Accepted by .....
David E. Hardt
Chairman, Department Committee on Graduate Theses

An Excursion with the Boltzmann Equation at Low Speeds: Variance-Reduced DSMC

by

Husain Ali Al-Mohssen

Submitted to the Department of Mechanical Engineering
on March 19, 2010, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The goal of the present thesis is to develop a practical method for simulating low-signal kinetic (small-scale) gaseous flows. These flows have recently received renewed attention in connection with the design and optimization of MEMS/NEMS devices operating in gaseous environments; they are typically described using the Boltzmann equation which is most efficiently solved using a stochastic particle simulation method known as direct simulation Monte Carlo (DSMC). The latter is a simple and versatile simulation method which is very efficient in producing samples of the single particle distribution function which can be used for estimating hydrodynamic properties. Unfortunately, in cases where the signal of interest is small (e.g. low-speed flows), the computational cost associated with reducing the statistical uncertainty of simulation outputs becomes overwhelming.

This thesis presents a variance reduction approach for reducing the statistical uncertainty associated with low-signal flows thus making their simulation not only possible but also efficient. Variance reduction is achieved using a control variate approach based on the observation that low-signal flows are typically close to an equilibrium state. As with previous variance reduction methods, significant variance reduction is achieved making the simulation of arbitrarily small deviations from equilibrium possible. However, in contrast to previous variance-reduction methods, the method proposed, which we will refer to as VRDSMC, is able to reduce the variance with virtually no modification to the standard DSMC algorithm. This is achieved by introducing an auxiliary equilibrium simulation which, via an importance weight formulation, uses the same particle data as the non-equilibrium (DSMC) calculation; subtracting the equilibrium from the non-equilibrium hydrodynamic fields drastically reduces the statistical uncertainty of the latter because the two fields are correlated. By retaining the basic DSMC formulation, in contrast to previous approaches, the VRDSMC approach combines ease of implementation with computational efficiency and the ability to simulate all molecular interaction models available within the DSMC formulation.

Our validation tests show that the proposed VRDSMC method provides consid-

erable variance reduction for only a small increase in computational cost and approximation error compared to equivalent DSMC simulations. In other words, by addressing the major weakness associated with DSMC, VRDSMC is well suited to the solution of low-signal kinetic problems of practical interest.

Thesis Supervisor: N. G. Hadjiconstantinou
Title: Associate Professor

Acknowledgments

Wow! What a journey!

It is very hard to write these words as there are genuinely so many people to be grateful to. I probably would not be at MIT had it not been for the help and advice of Maher, and I certainly would not have done so well had it not been for friendship and mentoring of Ammar and Samir. I am grateful to Reem for the fantastic support I have received over the years from Houston, and I am also indebted to Deborah for the years of help and support that she has provided to me and the rest of our research lab.

I would like to thank all of the great people with whom I regularly interact either through the social circles I have been affiliated with at MIT or in the greater Boston area. I have learned from (and leaned on) many people, especially Nathan, Babak, Belal, and Sarah(s) among many other kind souls. I would particularly like to thank Katie for being there for my family when we needed support the most.

It has been exciting over the last couple of years to have worked with those that helped start and fund Syphir, especially Courtland and Abdulrahman. I really look forward to seeing how far we can take our start-up!

On the academic front, I have had a fantastic experience collaborating with Faisal and Zubair over the last 5 years. I am most grateful for the many hours they have spent patiently explaining to me tools and approaches that have been critical to the success of this work. I am grateful for knowing Maria-Louisa and the other kind and fun students who helped me prepare for and do well in my quals; our numerous technical discussions have blossomed into friendships that I will always cherish. I would also like to thank Michael and Martin for being such great committee members and sticking with my project for such a long time.

I am also grateful to the great friends that have continued to support me over the years, both those from my childhood (Mishal and Kamal) and new ones that I hope to stay close with for decades to come (Gail, Mostafa, and Hussam among others).

It is very hard for me to imagine an office that is more "productive" than 3-355a.

I enjoyed the company and advice of Sanith, Thomas, and other lab members and genuinely cherish the amazingly close friendships that I have developed with Gregg, Ghassan, Colin, and Mr. Bagheri. I never thought I would learn so much about life, science, and myself from my peers and I know I am a better person today because of them.

My family and I are most deeply indebted to the many exceptional medical professionals at MIT Medical and Boston's Children's Hospital. I do not exaggerate when I say that we owe them a dept of gratitude for everything we hold dear, including the life and well-being of Ali.

I would finally like to thank my family in Saudi Arabia (my father, mother, and the rest of the family) and Boston. Zainab and Ali have gone through so much, and I hope I will be able to make it up to them one day. Speaking of family, I would also like to thank Nicolas, Mahmood, and Desiree for providing what seems to be an infinite reservoir of support, understanding, wisdom, and jokes. In particular, Dr. Hadjiconstantinou has been an amazing advisor and confidant who has allowed me to explore many fascinating and challenging problems. I will forever be thankful for and indebted to his caring, wisdom, kindness, and patience. I hope to get the chance to share what I have learned from him with others I meet in the future. To him and the other great people who helped me, I can only say a most humble "Thank you".

In the final analysis, I am most grateful to God for selecting me for this most exciting and challenging journey. I wonder what He has in store next for me.....

كم من عهود عذبة في عدوة الوادي النضير.....

Contents

List of Figures

List of Tables

Table 1: Symbols & Notation

Symbol	Description	Symbol	Description
n	number density	q_y	heat flux in y direction
ρ	density	$q_{y,0}$	reference heat flux
n_0	reference number density	π_{xy}	shear
ρ_0	reference density	k	Boltzmann Constant
$\boldsymbol{c}$	particle velocity vector	t	time
$\boldsymbol{x}$	position vector	m	molecular mass
W	weight (probability ratio)	λ	mean free path
$R(\boldsymbol{c})$	function/moment of $\boldsymbol{c}$	Kn	Knudsen number
$\langle R \rangle$	expected value of moment	σ	collision cross section
f	number density function	$f_{eq,0}$	global equilibrium density function
f_{eq}	equilibrium number density function	$\hat{f}$	KDE reconstruction of f
$\boldsymbol{c}_i$	velocity of particle i	$\sigma^2\{W_i\}$	variance of samples W_i
N_{cell}	number of particles in a cell	Δt	time step
N_{Eff}	effective number of particles	Δy	cell width
N_{in}	number of particles arriving at a boundary condition in a timestep	U_W	wall speed
$\overline{R}$	estimate of moment R	σ_u	measure of variance of u
$\overline{R}_{VR}$	VR estimator of moment R	ε	normalized KDE kernel radius
W_i	weight of particle i	$\boldsymbol{c}_i \sim f(\boldsymbol{c})$	samples $\boldsymbol{c}_i$ drawn from distribution $f(\boldsymbol{c})$
MB	Maxwell-Boltzmann distribution	T	temperature
S_i	set of particles that are close to particle i	T_0	reference temperature
$\|S_i\|$	number of elements in set S_i	$\overline{\|S_i\|}$	average number of elements in set S_i
Ω	scattering angle	u_x,u_y,u_z	cell velocity in x,y,z directions
c_r	relative speed	V	cell volume
$\hat{c}_r$	collision probability	$\overline{c}$	average particle velocity
c_{mpv}	most probable velocity	c_0	reference particle velocity
$N_{\boldsymbol{c}_i}$	number of particles of class $\boldsymbol{c}_i$	$\tau_{collision}$	mean time between collisions
MX	maximum $W_i c_r$ in a cell	$f_{MB,loc}$	local MB distribution

Chapter 1

Introduction and Context

The study of rarefied gases has been an active field of research for well over a century, with initial work by Boltzmann [1] paving the way to the widespread acceptance of the molecular theory of gases. The Boltzmann Equation (BE) written more than a century ago is still in widespread use in everyday engineering and science with a surprisingly small amount of change.

Recent interest in small-scale, low-speed, gaseous flows [39] has renewed the need for efficient methods for solving Boltzmann-type kinetic equations [40]. In this work, we develop a method which addresses the major weaknesses associated with one of the most successful methods of simulating the Boltzmann equation, known as the Direct Simulation Monte Carlo method (DSMC), by reducing the statistical uncertainty associated with sampling hydrodynamic quantities in low-signal simulations.

1.1 Thesis Overview

In the present Chapter, we present a detailed introduction to the problem of interest, namely low-signal gaseous flows, as well as some of the basic ideas in the field of kinetic gas theory and kinetic gas simulation. Furthermore, we provide an overview of the best currently available methods for numerically describing slow rarefied gas flows. In Chapter 2 we introduce the basic ideas used in this Thesis for achieving variance reduction. Specifically, we discuss the concept of importance weights and

show how they can be used to produce variance-reduced estimators that address the weakness associated with DSMC. In the same Chapter we also give an overview of Kernel Density Estimation, which is another important tool extensively used in this work.

In Chapter 3 we present a new variance-reduced DSMC method which we will refer to as VRDSMC. Chapters 4 and 6 present further refinements and validation of the VRDSMC method.

In Chapter 6 we also introduce an alternative approach for deriving some of the theoretical results of Chapter 3: this new approach not only generalizes the derivations of the Chapter, but also provides a theoretical framework for extending our work to other types of stochastic particle simulation methods. In Chapter 7 we apply our refined VRDSMC method to a more complex and detailed problem. Finally, we conclude in Chapter 8 with a summary and prospectus.

1.2 Dilute Gases at Small Scales

Although the majority of dilute-gas hydrodynamics can be described using the Navier-Stokes equations, when the molecular mean free path becomes of the order of the characteristic flow lengthscale, the latter description is no longer valid. This happens at low pressures when the mean free path is large [12] or in small scale devices [57] where characteristic lengthscales are small. This Thesis focuses on the latter applications that have become commonplace with the advent of MEMS/NEMS [18] (We note here that due to the applications associated with reentry vehicles in the upper atmosphere, this field is frequently referred to as Rarefied Gas Dynamics). The hydrodynamics of dilute gases at all scales can be described using kinetic theory [40, 46, 4, 1], which describes gases at the molecular level using probability distributions. For dilute gases, the assumption of molecular chaos [40] and binary collisions allows kinetic theory to describe the gas in terms of the single particle distribution function $f(\boldsymbol{x}, \boldsymbol{c}, t)$ which gives the number of particles in a differential volume at location $\{\boldsymbol{x}, \boldsymbol{c}\}$ in phase space at time t. For engineering applications, many gases at

standard temperature and pressure satisfy the dilute gas criteria [12, 4]. The evolution of $f(\boldsymbol{x}, \boldsymbol{c}, t)$ is governed by the Boltzmann equation that is described in Section 1.3.

The degree of breakdown of the Navier-Stokes description is quantified by the Knudsen number $Kn = \lambda/H$, where λ is the molecular mean free path and H is a characteristic lengthscale [2]. For a hard-sphere gas which can be analyzed exactly in some ways, the (equilibrium) value of λ is given by $(\sqrt{2}\pi\, n\, d^2)^{-1}$ where d is the gas diameter and n is the gas number density. Because not all gases can be well-approximated by hard-spheres, usually a viscosity-based definition is used for the mean free path, namely $\lambda = \frac{\mu}{P_0}\sqrt{\frac{\pi R T_0}{2}}$. The mean time between collisions is defined as $\tau_{collision} = \frac{\lambda}{\bar{c}}$, where $\bar{c} = \sqrt{\frac{8RT_0}{\pi}}$ is the mean molecular speed. Here R is the gas constant, T_0 is the gas equilibrium temperature and P_0 is the gas equilibrium pressure.

As expected, when $Kn \ll 1$, molecular motion is diffusive and the Navier-Stokes description holds. For $Kn \gg 1$, particle motion is primarily collisionless (ballistic) and can be described by a collisionless Boltzmann equation. In the regime $0.1 \lesssim Kn \lesssim 10$, particle motion is neither purely diffusive nor purely ballistic. As a result, a Navier-Stokes description, in general, cannot describe this regime.

The present Thesis is focused on developing efficient methods for solving the governing kinetic equation describing gas dynamics in this regime. At atmospheric pressure, $0.1 \lesssim Kn \lesssim 10$ corresponds to a characteristic length scale of $0.5\mu m \lesssim H \lesssim 5nm$.

1.3 Boltzmann Equation

In this section we present the equation governing the evolution of the particle distribution function in dilute gases known as the Boltzmann Equation (BE). The BE

$$\frac{\partial f}{\partial t} + \boldsymbol{c}\cdot\frac{\partial f}{\partial \boldsymbol{x}} = \left[\frac{\partial f}{\partial t}\right]_{Collision} \tag{1.1}$$
$$\left[\frac{\partial f}{\partial t}\right]_{Collision} = \int\int \left(f' f_1' - f f_1\right) c_r \sigma d\Omega d\boldsymbol{c}_1$$

can be rigorously derived from the Liouville Equation [6] and is given here for a hard-sphere gas in the absence of body forces. Here c_r is the magnitude of the relative velocity of particles with velocities $\boldsymbol{c}$ and $\boldsymbol{c}_1$

$$c_r = \|\boldsymbol{c}_r\|, \boldsymbol{c}_r = \boldsymbol{c} - \boldsymbol{c}_1$$

and

$$f = f(\boldsymbol{x}, \boldsymbol{c}, t) \qquad f_1 = f(\boldsymbol{x}, \boldsymbol{c}_1, t) \quad f' = f(\boldsymbol{x}, \boldsymbol{c}', t) \quad f_1' = f(\boldsymbol{x}, \boldsymbol{c}_1', t),$$

where $\boldsymbol{c}'$, $\boldsymbol{c}_1'$ are the post-collision velocities, related to the pre-collision velocities through the scattering angle Ω and σ the molecular cross section. In this section and in the remainder of the paper, unless otherwise stated, integration in velocity space extends over $\mathbb{R}^3$, while the solid angle integration is over the surface of the unit sphere.

It is sometimes useful is to re-write the collision term in Equation 1.1 in the following equivalent "weak" form [46]:

$$\left[\frac{\partial f}{\partial t}\right]_{Collision} = \frac{1}{2}\int\int\int \left(\delta_1' + \delta_2' - \delta_1 - \delta_2\right) f_1 f_2 c_r \sigma d\Omega d\boldsymbol{c}_1 d\boldsymbol{c}_2 \tag{1.2}$$

where $\delta_1 = \delta(\boldsymbol{c}_1 - \boldsymbol{c}), \delta_2 = \delta(\boldsymbol{c}_2 - \boldsymbol{c}), \delta_1' = \delta(\boldsymbol{c}_1' - \boldsymbol{c}), \delta_2' = \delta(\boldsymbol{c}_2' - \boldsymbol{c})$.

The hard-sphere collision model assumes particles only interact when their centers are a molecular diameter away from each other giving a molecular collision cross section $\sigma = \frac{\pi d^2}{4}$ that is constant. The hard-sphere model is not the only interaction model; many others exist [12], including the Maxwell collision model ($\sigma \propto \frac{1}{c_r}$), and the Variable Hard Sphere (VHS) model ($\sigma \propto c_r^{-A}$ for a constant $A > 0$) which reproduces a more realistic transport coefficient (e.g. viscosity) dependence on temperature.

The NS system of equations can be derived from the BE through what is known as the Chapman-Enskog Expansion [19] in the limit $Kn \ll 1$. Connection to hydrodynamics is made through the moments of $f(\boldsymbol{x}, \boldsymbol{c}, t)$. Specifically, the gas number density is defined as

$$\langle n \rangle = \int f \, d\boldsymbol{c}$$

the flow velocity as

$$\langle \boldsymbol{u} \rangle = \frac{\int \boldsymbol{c} f \, d\boldsymbol{c}}{\int f \, d\boldsymbol{c}}$$

and so on. The angled brackets here are used to remind us that these quantities are moments of f. In the remainder of this Thesis, we will be using the standard hydrodynamic notation that does not involve brackets, thus using n to denote the number density, $\boldsymbol{u} = \{u_x, u_y, u_z\}$ the flow velocity, T the temperature and $\boldsymbol{q} = \{q_x, q_y, q_z\}$ the heat flux vector. In the interest of simplicity, we will use these symbols to also denote the cell-averaged simulation estimates of these quantities. Cell-averaged estimators are described in more detail in Appendix B.

1.4 Low-Signal Problems

Although the field of kinetic theory has been dominated in a large part of the 20th century by applications related to high-speed/low-pressure problems that are common in the aerospace industry [24, 41], this has slowly changed over last two decades. As the development of MEMS and small applications has matured, it has become more common to find gas problems that can not be accurately described by the NS system of equations, even with slip corrections, as the characteristic length-scale decreases. Today, the development of accurate engineering models of such flows is not only important for the fundamental understanding, but also for the design [25, 57], and even fabrication of such devices [31].

An important differentiating feature of micro applications is that they are slow flows with low Reynolds and Mach (Ma) numbers, in contrast to older applications that were typically characterized by $Ma \gtrsim O(1)$. Because the deviation from equilibrium scales with the Mach number [25], this regime change ($Ma \ll 1$ instead of $Ma \geq 1$) has resulted in new challenges in the search for efficient methods for solving the BE. In particular, when performing molecular-like simulations of such flows, low Ma translates to low signals that require overwhelming amounts of sampling to discern.

As an example of a low-signal (i.e. slow flow) MEMS device, let us look at a

simple hard-drive head. A hard drive head is suspended over a rotating platter using hydrodynamic forces that enable it to read and write data at high density. For a typical 4200rpm drive, the gap between the drive surface and the head is about 30nm [64] resulting in $Kn = O(1)$. However, the characteristic velocity associated with the disk is $Ma \lesssim 0.1$ which is hard to model using today's simulation methods as we will elaborate on in the next section.

1.5 Solving the Boltzmann Equation

There are a number of general approaches to solving the BE for low-signal flows. One route is to numerically solve the equation as in the work of Sone et. al. [7, 46]; this approach is challenging because, in general, the solution has to be discretized in 7 dimensions (3 spatial, 3 velocity and time) which becomes computationally prohibitive. Though potentially very accurate, explicit numerical solutions of the BE are very rare, as they are typically achieved by exploiting certain problem symmetries and are usually obtained at great computational expense [7, 25]. Analytical approximations can also be very powerful, especially if they lead to descriptions that are simple or numerically tractable. Sections 1.5.1-1.5.3 discuss some of these approaches.

1.5.1 Linearization of the Boltzmann Equation

Due to the complexity associated with the Boltzmann collision operator, most theoretical studies proceed by linearizing this operator. This approximation is only valid for small departures from equilibrium which fortunately, are very frequent in practical applications of interest. The resulting linearized BE can be solved numerically [8] or analytically.

1.5.2 Slip Flow Models

Slip flow models attempt to push the regime where the NS description can be used beyond $Kn \ll 0.1$ up to $Kn \approx 0.1$ by applying a number of corrections to the wall

boundary condition. This approach works because the breakdown of the continuum assumption happens gradually and for small Knudsen numbers is localized around boundaries [39]. Depending on the geometry in question, and the boundary conditions of interest, the NS description with slip can be maintained for problems with $Kn \simeq 0.4$ or more [34].

When satisfied, slip-flow models are by far the best method of solving slow kinetic problems. This is because they are computationally cheap and are derived rigorously from the Hilbert expansion of the linearized BE [46].

1.5.3 The BGK Model

The Bhatnagar-Gross-Krook collision model [4]

$$\left[\frac{\partial f}{\partial t}\right]_{Collision} = \nu(\boldsymbol{c})[f(\boldsymbol{c}) - f_l(\boldsymbol{c})] \tag{1.3}$$

also referred to as the BGK model is sometimes used to replace the Boltzmann collision operator. Here f_l is a local equilibrium Maxwell-Boltzmann distribution. This model essentially makes the rather crude assumption that the role of the collision operator is to drive the system towards local equilibrium at a rate $\nu(\boldsymbol{c})$. Despite the crudeness, the BGK method is probably the most widely used collision model after the BE itself and it has been used very successfully in many (especially isothermal [6]) applications [23]. Although, in this work, we focus on solving kinetic flows using the exact hard-sphere BE, our methodology is applicable to direct Monte Carlo Simulations of the BGK Model [67].

1.6 Direct Simulation Monte Carlo (DSMC):

Although quite powerful, analytical or deterministic numerical solutions of the Boltzmann equation can only be obtained for very simple specialized problems [46]. In other words, the majority of engineering problems of practical interest are not amenable to such approaches.

In principle, the BE can be solved using a molecular-dynamics approach. However in 1963 Bird invented DSMC [12], a stochastic particle method which has gone on to be by-far the most successful numerical method for solving the BE. At the heart of its success is the ability to solve engineering (complex) problems using modest computational resources without requiring complex setup or discretization. Moreover, DSMC is unconditionally stable, and very easily parallelizable (see [36, 54, 24]).

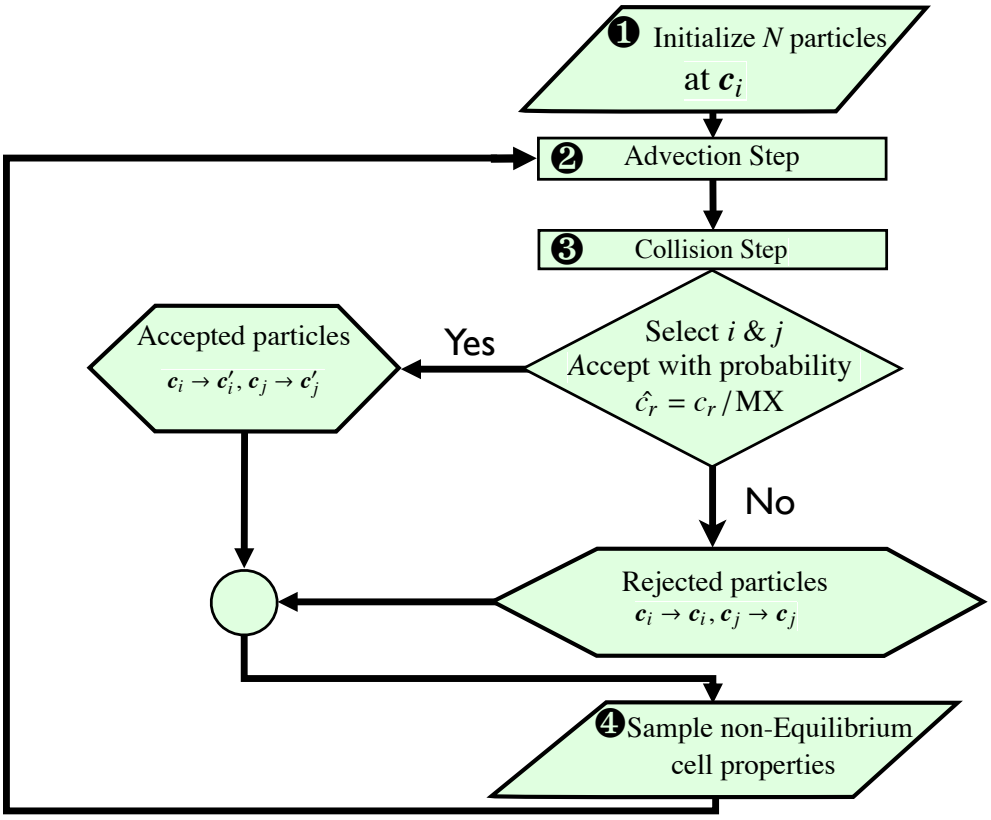

Figure 1-1: A flowchart of the standard DSMC method. We will return to variations of this in the next chapters as we build the method that is the subject of this Thesis.

As usual in particle methods, the distribution function in DSMC is represented by N particle-simulators by writing

$$f \approx N_{Eff} \sum_{i=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{x}_i)\delta(\boldsymbol{c} - \boldsymbol{c}_i) \tag{1.4}$$

where each particle represents N_{Eff} physical molecules. The computational domain is

divided into cells that contain N_{cell} particles. The Boltzmann equation is integrated in time by splitting molecular motion into a series of a collisionless advection and collision sub-steps of length Δt.

The advection sub-step updates the distribution function due to the action of the advection operator

$$\frac{\partial f}{\partial t} + \boldsymbol{c} \cdot \frac{\partial f}{\partial \boldsymbol{x}} = 0 \tag{1.5}$$

simulated by updating all the particle positions which move in a collisionless manner. The collision sub-step updates the distribution function due to the action of the collision operator

$$\left[\frac{\partial f}{\partial t}\right]_{Collision} = \frac{1}{2} \int \int \int \left(\delta_1' + \delta_2' - \delta_1 - \delta_2\right) f_1 f_2 c_r \sigma d\Omega d\boldsymbol{c}_1 d\boldsymbol{c}_2 \tag{1.6}$$

Collisions take place between particle partners selected randomly within the cell. A convergence proof for this algorithm in the limit of taking $\Delta t \to 0$ and $\Delta x \to 0$, can be found in [10, 12], while an analysis of the error associated with the timestep discretization can be found in [20].

Cell quantities of interest are calculated as averages over the sample particle simulators in that cell. The main steps, as summarized in Figure 1-1, are:

1. **Initialize particles:** Based on initial conditions, create particles in simulation domain with positions $\boldsymbol{x}_i$ and velocities $\boldsymbol{c}_i$.

2. **Advection Step:** Particle positions are updated based on their current velocities. For each particle

   $$\boldsymbol{x}_i \to \boldsymbol{x}_i + \Delta t \boldsymbol{c}_i$$

   This step also includes boundary condition imposition, e.g. detecting wall interactions and updating particle velocities based on the boundary condition.

3. **Collision Step:** For each cell $N_{candidate} = N_{Eff} N_{cell}^2 M X \sigma \Delta t / (2V)$ candidate

particles from that cell are randomly selected and collided with probability

$$P = c_r/MX$$

where MX is the maximum relative velocity in a cell. If a pair i and j are accepted for collision, their velocities are updated according to

$$\boldsymbol{c}_i \rightarrow \boldsymbol{c}_i'$$

and

$$\boldsymbol{c}_j \rightarrow \boldsymbol{c}_j'$$

where the post-collision velocities $\boldsymbol{c}_i'$, $\boldsymbol{c}_j'$ are calculated using a random scattering plane and conservation of momentum and energy. No velocity update is performed for rejected particle pairs.

4. **Sample step:** Properties of interest are calculated by sampling particle simulators. For example, if $\langle R(\boldsymbol{c})\rangle = \int R(\boldsymbol{c})f(\boldsymbol{c})d\boldsymbol{c}$ then the DSMC estimate is given by

$$\overline{R(\boldsymbol{c})} = N_{Eff}\sum_{i=1}^{N} R(c_i)$$

DSMC in Low-Signal Flows

Since the kinetic properties of interest are calculated as averages over samples from each cell, the DSMC method produces results that are inherently stochastic in nature and can be thought of as being "noisy". For N samples, it is easy to show that, in general, the variance of $\overline{R}$, $\sigma^2\{\overline{R}\}$, will scale as

$$\sigma^2\left\{\overline{R}\right\} \propto \frac{\sigma_R^2}{N} \tag{1.7}$$

where σ_R is a constant that is determined by the physical conditions of the problem of interest (e.g. local gas temperature, velocity etc.); this is discussed in more detail by

Hadjiconstantinou et. al. in [29]. Clearly, if we are interested in a constant relative error $\sqrt{\sigma^2\{\overline{R}\}}/\langle R\rangle$ then $\langle R\rangle \to 0$ requires $N \to \infty$. In other words, as the problem of interest approaches equilibrium (for example, the wall speed $U_W \to 0$ in a Couette flow problem), more samples are needed to maintain a constant relative uncertainty in our estimates and keep our solutions useful. In fact, due to the slow convergence with N ($\sigma\{\overline{R}\} \propto N^{-\frac{1}{2}}$) low signal problems quickly become intractable.

1.7 Variance Reduction

Due to the overwhelming cost associated with resolving low-signal flows using Monte Carlo formulations, variance reduction approaches have recently received considerable attention: Baker and Hadjiconstantinou [36] showed that solving for the deviation from equilibrium drastically reduces the statistical uncertainty and enables the simulation of arbitrarily small deviations from equilibrium. In this general approach, the BE is re-written in terms of the deviation function $f_d = f - f_{MB}$ (with f_{MB} being an appropriately chosen reference Maxwell-Boltzmann distribution) and results in a set of equations that can efficiently simulate f_d.

The same authors also showed that variance-reduced formulations can be developed for both particle methods [61], referred to as deviational, and PDE-type approaches [36, 60]; a particle method that is equivalent to the one detailed in [61] has also been proposed by Chun and Koch [35]. Unfortunately, in these *particle* methods, if collision operator 1.1 is used, particle cancellation in the collision dominated regime is required for stability [35, 61]; this adversely affects both accuracy and efficiency.

Later the variance-reduction approach was extended to a Discontinuous Galerkin formulation in an attempt to combine the advantages of PDE approaches (high-order convergence) with low-variance Monte Carlo evaluations.

1.7.1 LVDSMC

More recently, Homolle and Hadjiconstantinou [50, 54] have extended the deviational simulation approach in a manner that removed the stability limitation reported by

previous researchers [62, 35]. The main innovation was to introduce a special form of the collision operator that can be written in a form that "pre-cancels" particle creation and deletion leading to a significantly smaller number of simulation particles and thus avoiding the problem of uncontrolled particle number growth. Using this approach, they developed an efficient particle method for simulating the hard-sphere gas known as LVDSMC (Low-Variance Deviational Simulation Monte Carlo) [50, 54]. LVDSMC differs from DSMC only in ways necessary to simulate the deviation from equilibrium, and so combines the strengths of particle methods with substantially reduced variance. This method has recently been extended to treat the relaxation-time BGK approximation [66].

The starting point of the LVDSMC method is the hard-sphere collision integral [51]

$$\left[\frac{\partial f}{\partial t}\right]_{Collision} = C(f,f)\,(\boldsymbol{c})$$

where

$$C(f,h) = \int\int \left(f'h'_1 - fh_1\right) \|\boldsymbol{c}-\boldsymbol{c}_1\| \sigma d\Omega d\boldsymbol{c}_1$$

By making the substitution $f \rightarrow f_{MB} + f_d$, this can be re-written as

$$\left[\frac{\partial f}{\partial t}\right]_{Collision} = C(f_{MB}+f_d, f_{MB}+f_d) = C(f_{MB},f_{MB})+C(f_{MB},f_d)+C(f_d,f_{MB})+C(f_d,f_d)$$

Since a Maxwell-Boltzmann distribution does not change under the influence of the collision integral, the first term is zero while the last one is 2nd order and will be ignored in the interest of simplicity. (In their work they are able to include it without significant challenges [50, 54]). The linear terms are given by

$$C(f_{MB},f_d) + C(f_d,f_{MB}) = \int\int \left(f'_{MB}f'_{d1} + f'_d f'_{MB1}\right) \|\boldsymbol{c}-\boldsymbol{c}_1\| \sigma d\Omega d\boldsymbol{c}_1 - \int\int f_{MB}f_{d1} \|\boldsymbol{c}-\boldsymbol{c}_1\| \sigma d\Omega d\boldsymbol{c}_1 - \int\int f_d f_{MB1} \|\boldsymbol{c}-\boldsymbol{c}_1\| \sigma d\Omega d\boldsymbol{c}_1 \quad (1.8)$$

In Equation 1.8 the first term of the right hand side can be written as

$$\int\int \left(f'_{MB}f'_{d1} + f'_{d}f'_{MB1}\right) ||\boldsymbol{c}-\boldsymbol{c}_1||\sigma d\Omega d\boldsymbol{c}_1 = \int K_1(\boldsymbol{c},\boldsymbol{c}_1)f_d(\boldsymbol{c}_1)d\boldsymbol{c}_1 \tag{1.9}$$

while the second term can be written as

$$\int\int f_{MB}f_{d1}\|\boldsymbol{c}-\boldsymbol{c}_1\|\sigma d\Omega d\boldsymbol{c}_1 = \int K_2(\boldsymbol{c},\boldsymbol{c}_1)f_d(\boldsymbol{c}_1)d\boldsymbol{c}_1 \tag{1.10}$$

Finally the last term can be written as

$$\int\int f_{MB1}f_d\|\boldsymbol{c}-\boldsymbol{c}_1\||\sigma d\Omega d\boldsymbol{c}_1 = \nu(\boldsymbol{c})f_d(\boldsymbol{c}) \tag{1.11}$$

for some functions $K_1(\boldsymbol{c},\boldsymbol{c}_1)$, $K_2(\boldsymbol{c},\boldsymbol{c}_1)$, and $\nu(\boldsymbol{c})$ that are explicitly listed in the given reference. The details of the method are quite complex but the key point about the approach is that Equation 1.11 can be interpreted as a loss term that deletes particles while the convolutions in 1.9 and 1.10 create deviational particles. We will further discuss some of the features of LVDSMC in Chapter 5; We close by noting that LVDSMC is a rigorous method (convergence proof can be found in [58]) that has very little bias and it has been used to produce accurate results reliably over the last few years. Most recently it has been extended to other collision models like the VHS [58], although implementations are still in progress, primarily due to the additional complexity involved.

1.8 Thesis Objective and Solution Approach

The objective of the present Thesis is to develop a numerical method that enables the efficient simulation of low-signal rarefied gas flows. More specifically, our objective is to produce a method that retains many of the features that made DSMC so successful over the last four decades, while providing a significant advantage over DSMC in low-signal flows that are of current interest. Specifically, we seek a method that is:

1. Accurate; in the sense that is rigorously based on the governing equation (BE).

2. Simple compared to other methods that are used today to solve the BE. Ideally, it will be only marginally more complex than DSMC.

3. Practical; in the sense that it should not have excessive resource requirements (unlike for example [35, 62]) or only apply to special cases (e.g. [7, 65]).

4. Flexible; so that it can easily incorporate as many collision models or special boundary conditions as possible.

5. Efficient; in the sense that allows the simulation of "large" problems ([42, 41]) with a cost that scales with linearly (or close to) N.

Although many of the methods examined in Section 1.7 have some of these features, we believe that the method proposed in this Thesis holds a significant advantage in terms of requirements 2 and 4.

Chapter 2

Variance Reduction Using Importance Weights and Kernel Density Estimation

This Chapter will give a general introduction to a set of tools and concepts that are critical to our proposed approach and that will be developed further in Chapters 3 and 6. The first three Sections of this Chapter give a general introduction of the concept of importance sampling and how it can be used to find variance-reduced estimators of some integrals that are common, especially when evaluating hydrodynamic properties. A key concept that will be introduced and used will be that of importance weights. Section 2.4 proceeds to show how stability problems can appear in applications when importance weights are generated in a conditional manner and how the stability problem can be dealt with. The final section introduces Kernel Density Estimators (KDE) which are mathematical tools that will be instrumental in producing stable variance-reduced calculation methods that are compatible with DSMC and low-signal flows.

2.1 Historical Introduction to Variance Reduction

Variance reduction is a well-established approach used in many numerical analysis applications ranging from medicine to finance [32] in addition to many disciplines of science and engineering [38]. Variance reduction using weights lends itself particularly well to applications describing linear processes, for example, simulations of luminosity [15], or radiation problems [9].

Variance reduction techniques have been used for particle simulation applications in many different contexts especially when there is little interaction between the simulated particles. In [33] a variance reduction technique is used to increase the fidelity of the simulation of Brownian dynamics of suspended particles resulting in major computational savings. Indeed, a major inspiration for this work is the use of variance reduction to simulate long molecules or chains of molecules [47]. This work has been extended to a more solid mathematical basis in [13].

In the area of kinetic gas simulation, variance reduction approaches using weights have been used to simulate gas mixtures where one of the components is of trace amounts and needs to have its properties accurately sampled. In such situations, it is common for particles of different effective weights to be used, despite some challenges related to random walks in momentum and energy as the simulation progresses in time [12]. In particular, [14] introduced a method of conserving mass, momentum and energy that has been successfully used in many applications [31].

2.2 Importance Sampling, Weights, and Variance Reduction

Before we can describe our variance reduction approach, we introduce importance sampling, a useful tool that is used in many applications, especially in performing

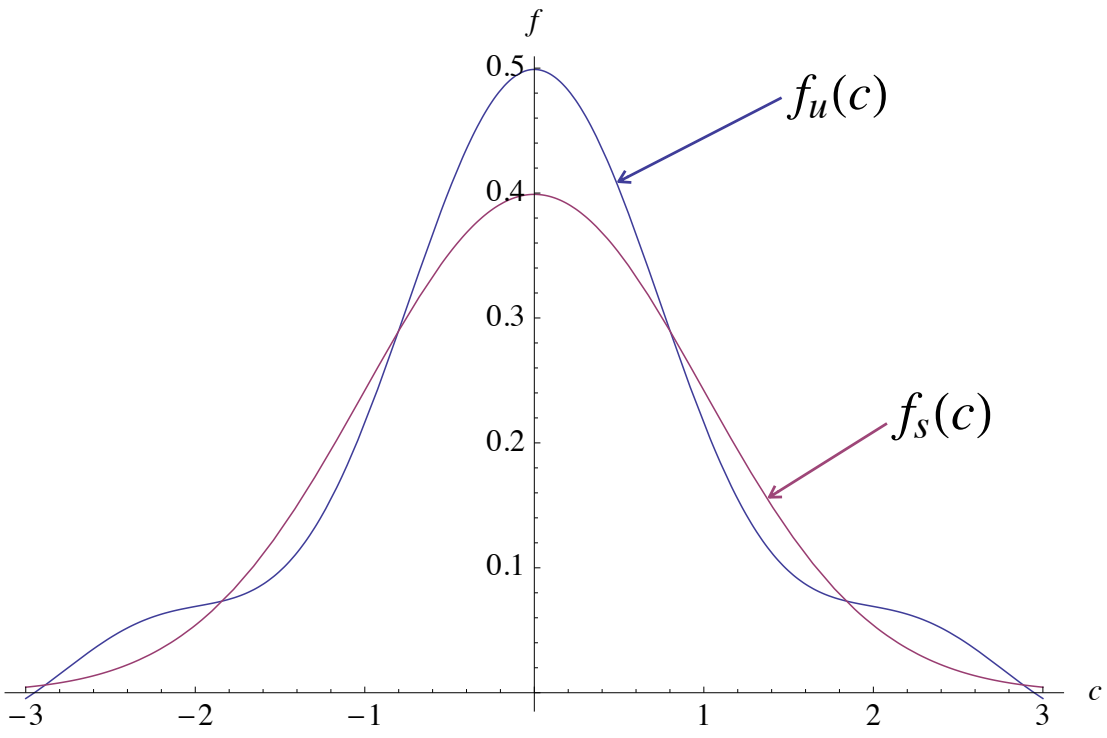

Figure 2-1: Plot of the functions used in the example described in Section 2.2. The purple line is that of $f_s(c)$ which is a Gaussian function while the blue line is the more complex function $f_u(c)$.

Monte Carlo integration. The simplest example is that of evaluating the integral

$$I = \int_a^b f_u(c)dc$$

for some complex function f_u. This can be done by finding the average value of f_u over the interval $[a, b]$, $\bar{f}_u$, and writing $I = (b-a)\,\overline{f_u}$. $\overline{f_u}$ can be estimated by uniformly generating N samples in the integration interval and using

$$\overline{f_u} = \frac{1}{N}\sum_{i=1}^{N} f_u(c_i)$$

where $c_i \sim$ uniformly distributed from $[a, b]$, where $\sim$ denotes "drawn from" the stated distribution.

It is well established [32] that the error in the above estimate scales as $\frac{1}{\sqrt{N}}$ which implies that a large number of samples is required if an accurate estimate of the integral is desired.

Is there some alternative formulation which results in a reduced uncertainty if we are able to generate samples from a distribution f_s such that $f_s \approx f_u$? It turns out that this can be readily done using a standard approach [48] called Importance Sampling integration. The first step in this process is to define an *importance weight*

$$W(c) = \frac{f_u(c)}{f_s(c)} \tag{2.1}$$

which can be evaluated at every sampling point c_i. Here and in the rest of this Thesis we will use the notation

$$W_i = W(c_i)$$

Intuitively we interpret W_i as the ratio of the probability of finding sample c_i in the distribution f_u for every sample we find in f_s. Using this, we can write

$$I = \int_a^b f_u(c)dc = \int_a^b \frac{f_u(c)}{f_s(c)} f_u(c)dc = \int_a^b W(c) f_s(c)dc$$

For N samples $\{c_i\} \sim f_s$ we make the approximation $f_s(c) \simeq \frac{1}{N} \sum_{i=1}^{N} \delta(c - c_i)$. This can be used to approximate the integral I by direct substitution

$$\overline{I} = \frac{1}{N} \sum_{i=1}^{N} W_i$$

To illustrate this concept, let us evaluate the integral

$$I = \int_{-2}^{2} f_u(c)dc$$

Table 2.1: Importance sampling integration using weights. The table compares the uncertainty (standard deviation) of the importance sampling approach to that of uniform sampling.

	Error Standard Deviation		
Sample Points	Importance Sampling	Uniform Sampling	IS Advantage
10	0.0725	0.2074	2.86
100	0.0231	0.0611	2.64
500	0.0113	0.0272	2.42
5000	0.0035	0.0091	2.63

where

$$f_u(c) = H(c) + \frac{1}{10} J_0(3c)$$

such that J_0 is a Bessel function of the 1st kind of order 0 and

$$f_s(c) = H(c) = \frac{e^{-\frac{c^2}{2}}}{\sqrt{2\pi}}$$

is a function that can be sampled and evaluated analytically as shown in Figure 2-1. By creating samples c_i and calculating weights from Equation 2.1, I can be estimated using the importance weights and compared to the exact result. Table 2.1 shows the results of this approach for a number of different sample sizes and the benefit importance sampling brings for this simple problem.

Importance Sampling can be thought of as a variance-reduction technique since it reduces the variance associated with estimators. In the next Section we discuss a related variance reduction technique based on the method of control variates.

2.2.1 Variance Reduction Using Importance Weights

Let us assume we are interested in evaluating

$$\langle R \rangle = \int R(c) f(c) dc \tag{2.2}$$

for some function $f(c)$ that can be sampled using a Monte Carlo procedure as

$$\overline{R} = \frac{1}{N}\sum_{i=1}^{N} R(c_i) \tag{2.3}$$

An alternative approach to 2.3 is to use a function $f_{ref}(c)$ (that is close to $f(c)$) such that the integral

$$\langle R \rangle_{ref} = \int R(c) f_{ref}(c) dc$$

can be evaluated in a deterministic way. By rewriting Equation 2.2 as

$$\langle R \rangle = \int R(c) f(c) dc = \int R(c) f(c) dc - \int R(c) f_{ref}(c) dc + \int R(c) f_{ref}(c) dc$$

and judiciously choosing f_{ref}, we can obtain the following variance-reduced estimate of $\langle R \rangle$ (that will be labeled $\overline{R}_{VR}$ for the rest of this document)

$$\overline{R}_{VR} = \overline{R} - \overline{R}_{ref} + \langle R \rangle_{ref} \tag{2.4}$$

Provided f_{ref} is appropriately chosen, this approach can have a substantial advantage in estimating $\langle R \rangle$ compared to Equation 2.3 [48]. Using the device of weights, defined here as

$$W_i = \frac{f_{ref}(c_i)}{f(c_i)}$$

expression 2.4 becomes

$$\overline{R}_{VR} = \frac{1}{N}\sum_{i=1}^{N} (1 - W_i)\, R(c_i) + \langle R \rangle_{ref} \tag{2.5}$$

In other words, provided f_{ref} is selected such that $\|W_i - 1\| \ll 1$, this approach results in an estimator with significantly smaller variance[63]. This is illustrated via an example in Section 2.3.

Low Signal Flows and Variance Reduction The variance reduction approach just described lends itself naturally for our purposes: in DSMC, the hydrodynamic

properties of interest (density, velocity, etc.) are simply integrals of the form (2.2) that are estimated via sampling leading to expressions such as 2.3. As will be detailed in the next chapters, the main thrust of this work is to use variance-reduced estimators of the form 2.5 to accurately evaluate hydrodynamic properties of interest.

2.3 Variance Reduction Magnitude as a Function of Deviation From Equilibrium: a Simple 1D Problem

Before we proceed, we would like to discuss some properties of the variance-reduced estimator $\bar{R}_{VR}$ (see Equation 2.5) introduced in the last Section. Specifically, the goal of this section is to examine how the variance of this estimator, $\sigma^2\{\overline{R}_{VR}\}$, depends on f and f_{ref}. Ideally we would like $\sigma^2\{\bar{R}_{VR}\} \to 0$ as $f \to f_{ref}$ sufficiently fast so that arbitrarily small deviations from equilibrium can be sampled (with finite resources). We will see in Chapter 3 that this property is the key feature that will allow us to simulate kinetic problems that are arbitrarily close to equilibrium.

To proceed, let us consider a simple case where $f(c) = \frac{1}{\sqrt{2\pi}\varsigma}\exp(-\frac{(c-u)^2}{2\varsigma^2})$ and $f_{ref}(c) = \frac{1}{\sqrt{2\pi}\varsigma}\exp(-\frac{c^2}{2\varsigma^2})$ for some constant ς. Let us assume that we are interested in $\langle c\rangle = \int cf(c)dc$ (known here analytically, $\langle c\rangle = u$). A Monte Carlo evaluation of this integral results in statistical uncertainty characterized by a variance of $\frac{\varsigma^2}{N}$, where N is the number of samples. Clearly as $u \to 0$ the relative statistical error $\frac{\sqrt{\varsigma^2/N}}{u} \to \infty$. It is precisely this phenomenon that prevents DSMC from resolving low-signal flows.

We now proceed to find how the statistical uncertainty associated with $\bar{c}_{VR}$ scales with u. If we take N samples $\{c_i\}$ generated from the distribution $f(c)$ the corresponding weights are:

$$W(c) = \frac{f_{ref}(c)}{f(c)} = \exp\left[\frac{u(u-2c)}{2\varsigma^2}\right]$$

And the set $\{c_i, W_i\}$ (where $W_i = W(c_i) = \frac{f_{ref}(c_i)}{f(c_i)}$) allows us to sample the properties

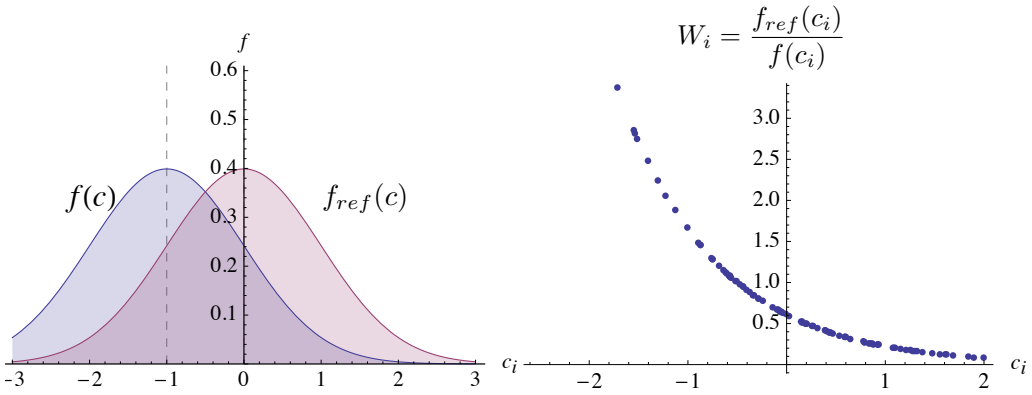

Figure 2-2: Left: Plot of distribution functions f and f_{ref} both of which are Gaussian with the same variance(ς^2) but different means. Right: The corresponding set $\{c_i, W_i\}$ which samples the distribution f_{ref}. As $f \to f_{ref}$, $\sigma^2\{W_i\} \to 0$.

of f_{ref}. A sample set is illustrated on the right plot of Figure 2-2.

We start by using Equation 2.4 to write a variance-reduced estimator of $\langle c \rangle$

$$\overline{c}_{VR} = \overline{c} - \overline{c}_{ref} + \langle c \rangle_{ref}$$

Since the last term is zero, the estimator becomes

$$\overline{c}_{VR} = \frac{1}{N} \sum_{i=1}^{N} (1 - W_i)\, c_i$$

Of particular interest is the behavior of the variance of this estimator for small

signals ($u \to 0$). This can be analytically evaluated by using the definition of variance:

$$\sigma^2\{\mathfrak{x}\} = \int \mathfrak{x}^2 f(\mathfrak{x}) d\mathfrak{x} - \left(\int \mathfrak{x} f(\mathfrak{x}) d\mathfrak{x} \right)^2$$

yielding

$$\sigma^2\{(1 - W_i)\, c_i\} = \int c^2 (1 - W(c))^2 f(c) dc - u^2$$

which can be evaluated to

$$\sigma^2\{(1 - W_i)\, c_i\} = e^{u^2/\varsigma^2}(u^2 + \varsigma^2) - \varsigma^2$$

When $u^2 \ll \varsigma^2$ this can be expanded as

$$\sigma^2\{(1 - W_i)\, c_i\} \simeq \left(\frac{u^2}{\varsigma^2} + 1 \right) \left(u^2 + \varsigma^2 \right) - \varsigma^2 = 2u^2 + O(\frac{u^4}{\varsigma^4})$$

Ignoring the higher order term we see that the variance scales like u^2 and the variance of $\bar{c}_{VR}$ will go to 0 as $u \to 0$. In other words, the variance-reduced estimator will maintain its utility regardless of how small u becomes. Alternatively, the relative statistical error $\frac{\sqrt{2u^2/N}}{u} = \sqrt{\frac{2}{N}}$ remains constant and independent of u as $u \to 0$. This result is ideal because it implies that for a given relative statistical uncertainty, arbitrarily small signals (u) can be captured at a cost that does not scale with u.

2.4 Variance Reduction and Stability: a Biased Coin Example

To further explore the ideas introduced up to this point and to set the stage for the introduction of principles that we will need later, let us examine an other simple "toy" problem that is based on coin flipping. Let $\{0, 1\}$ (or equivalently $\{heads, tails\}$) to be the space of allowed samples c_i which are drawn from some specified coin distribution. Furthermore, we assume there are two classes of coins that give us the samples c_i with different probabilities. The first distribution is a fair coin (denoted by subscript F)

that starting from a state c_i will arrive at state c_i' with a probability

$$T_F(c_i \to c_i') = \frac{1}{2} \tag{2.6}$$

In other words, regardless of the current state the coin will give us 0 or 1 with equal probability. In contrast, we have a biased coin (subscript B) that produces states/samples c_i' according to the probabilities

$$T_B(c_i \to c_i' = 1) = \frac{1}{2} + \alpha \tag{2.7}$$

$$T_B(c_i \to c_i' = 0) = \frac{1}{2} - \alpha \tag{2.8}$$

for some $0 < \alpha < \frac{1}{2}$.

We can use direct sampling to evaluate the expected mean of the coin flip. Assuming there are N samples $\{c_i\}$ from the biased coin, the expected mean is:

$$\langle c \rangle_B = \frac{1}{N} \sum_{i=1}^{N} c_i \tag{2.9}$$

An identical formula can be written for the fair coin if we can directly generate samples c_i from it. Let us imagine that we are not able to directly sample the fair coin but instead are given the set $\{c_i, W_i\}$ where the c_i samples come from the biased coin and W_i are chosen to properly reflect the relative likelihood that a particular event happens in the fair experiment given that it happened in the biased experiment. Clearly, using importance sampling, the mean of the fair coin can be written as

$$\langle c \rangle_F = \frac{1}{N} \sum_{i=1}^{N} W_i c_i$$

But since we already know that the long term value of this is $\frac{1}{2}$ we can use Equation 2.4 to write the following variance-reduced estimator of the mean of the biased coin:

$$\langle c \rangle_{VR,B} = \frac{1}{N} \sum_{i=1}^{N} (1 - W_i)\, c_i + \frac{1}{2} \tag{2.10}$$

We expect that this estimator will be much more efficient than estimator 2.9 as long as $W_i \sim 1$ [63, 68].

This process becomes somewhat more complicated when attempting to generate the set $\{W_i\}$ sequentially based on previous states. Let us consider a set of samples and weights $\{c_i, W_i\}$ as before; furthermore let us take a "step" updating the samples $c_i \to c_i'$ according to the biased distribution Equation 2.7-2.8. Intuitively, we see that for every biased coin that starts at c_i and ends at 1 there are

$$\frac{T_F(c_i \to 1)}{T_B(c_i \to 1)} = \frac{\frac{1}{2}}{\frac{1}{2} + \alpha} = \frac{1}{2\alpha + 1}$$

fair coins that arrive at 1. In a similar fashion, for every biased coin that arrives at 0 there are

$$\frac{T_F(c_i \to 0)}{T_B(c_i \to 0)} = \frac{\frac{1}{2}}{\frac{1}{2} - \alpha} = \frac{1}{1 - 2\alpha}$$

fair coins arriving at 0. For each new biased coin state c_i', we can represent the new state of the fair coin $\{c_i', W_i'\}$ where

$$W_i' = W_i \frac{T_F(c_i \to c_i')}{T_B(c_i \to c_i')} \tag{2.11}$$

In other words if we continue to flip coins in a biased way, the set $\{c_i', W_i'\}$ will continue to represent a fair coin distribution as long as weights are updated according to Equation 2.11.

If we take a set of N coins and apply these update rules we will notice that the variance of the set of weights $\sigma^2\{W_i\}$ will increase as we apply more and more steps using Equation 2.11. Symbolically, if we take t steps we will have

$$\lim_{t \to \infty} \sigma^2 \{W_i\} |_t = \infty \tag{2.12}$$

Practically, this means that we can not continue to apply the above update rule and still have a useful variance-reduced estimator of the properties of the biased coin.

To illustrate this we show in Figure 2-3 a numerical simulation of $N = 1,000$ coins

and plot the variance $\sigma^2\{W_i\}|_t$ as a function of the number of times we update the weights. Clearly, after a certain time, this approach is of little practical value.

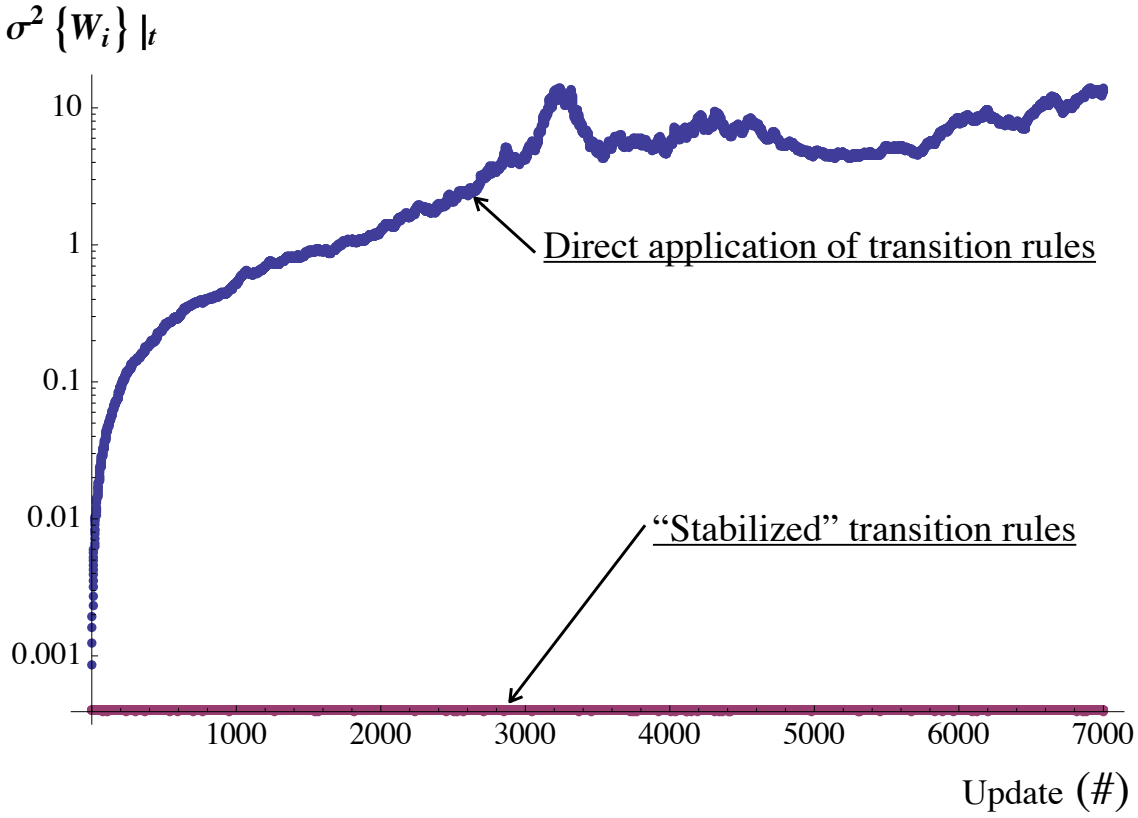

Figure 2-3: Illustration of different update rules for a simple iterative coin weight update scheme. On the one hand the variance of the weights grows without bound if we use the simple conditional weight update rule but we have a stable weight growth if we use the stabilized update rules explained in the main text. This plot was produced for $\alpha = \frac{1}{100}$ using $N = 1000$ simulated coins.

So what can be done to update the weights such that the simulation of the coin state is stable over time? The answer lies in using more than the conditional probability of the change of state based on *only* a sample's current state. Instead, we explicitly estimate the distribution functions of biased coins $f_B(c)$ and the fair coins $f_F(c)$. Since the N coin samples c_i are drawn from the biased distribution, we can

write

$$f_B(0) = \frac{||S_0||}{N}$$
$$f_B(1) = \frac{||S_1||}{N}$$

where S_{c_i} is the set of coins that are at c_i and $\|S_i\|$ is the number of elements in that set implying $||S_0|| + ||S_1|| = N$ and $f_B(0) + f_B(1) = 1$. Using importance sampling we can apply a similar approach to calculate the distribution of fair coins:

$$\begin{aligned} f_F(0) =& \frac{1}{N} \sum_{i \in S_0} W_i \\ f_F(1) =& \frac{1}{N} \sum_{i \in S_1} W_i \end{aligned} \tag{2.13}$$

These relations allow us to write an estimate of weights of particles at c_i which utilizes the definition

$$\hat{W}_i(c_i) = \frac{f_F(c_i)}{f_B(c_i)} = \frac{\sum\limits_{i \in S_{c_i}} W_i}{||S_{c_i}||} \tag{2.14}$$

rather than conditional probability argument 2.11.

We will call Equation 2.13 and Equation 2.14 our *stabilization step* because when they are applied they have the effect of keeping the variance $\sigma^2\{W_i\}$ from growing without bound, regardless of the length of the simulation. A plot of the variance of such a stabilized simulation is given in Figure 2-3.

Conceptually, one reason we are able to stabilize the coin simulation is that we are able to explicitly estimate the values of the distribution functions f_B and f_F at the sample points c_i' by counting the samples that are at each state instead of estimating samples of these probabilities via the conditional probabilities T. We will see later in Section 3.3.2 that our proposed variance-reduced simulation of the BE has a stabilization step that is based on this idea. The stabilization step in that case will be more complex for reasons we will describe in the next section.

2.5 Kernel Density Estimation and Stability

In the last section we saw how the key to producing a stable propagation scheme for weights is finding a way of estimating the unknown distributions f_B and f_F from their samples. Unlike a coin that has only two states, we will be later interested in gas distributions that are over $\mathbb{R}^3 \times \mathbb{R}^3$ making it impossible to directly sum over each point in the space to estimate the value of the distribution function. The generalization of the "measurement" step used in Section 2.4 for functions that are defined over continuous variables is called Kernel Density Estimation (KDE) and will be introduced here. As we will see later (Section 3.3.2), KDE will be used to calculate estimates of distribution functions from their samples in situations where the distributions are not analytically available.

2.5.1 Introduction to KDE

Let us assume we have N samples of an unknown probability distribution function f. We can reconstruct an approximation of f, which we will label $\hat{f}$, by writing it as a sum of kernel functions:

$$\hat{f}(c) = \frac{1}{N} \sum_{i=1}^{N} K(c - c_i) \simeq f(c) \tag{2.15}$$

such that K is a normalized ($\int K(c)\, d\, c = 1$) and positive function with a local support. In this setup, $\hat{f}$ is called a *Kernel Density Estimate of* f and can be a faithful representation of f given enough samples and a kernel K that has a small enough support.

To illustrate this concept, let us look at a simple visualization of an example function and its KDE reconstruction. In Figure 2-4 we show a KDE reconstruction of a Gaussian function f from 100 sample points $c_i \sim f$. In this particular example

we have selected the kernels to be simple box functions

$$K(\Delta c) = \begin{cases} \frac{1}{\varepsilon} & |\frac{\Delta c}{\varepsilon}| < 1 \\ 0 & otherwise \end{cases} \tag{2.16}$$

Although the correspondence between f and its reconstruction $\hat{f}$ is far from perfect, the two functions get closer to each other as we increase the number of samples and decrease ε.

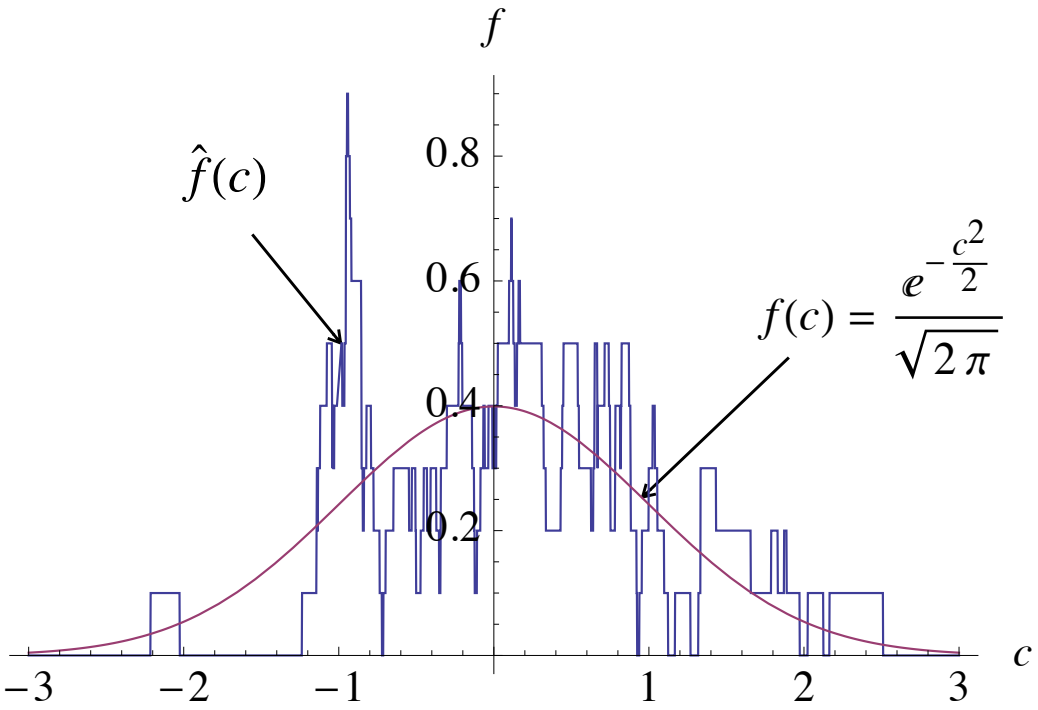

Figure 2-4: KDE reconstruction of a Gaussian distribution function using kernels of width ε=0.1 and 100 sample points. The original function $f(c)$ is smooth, while the reconstructed function $\hat{f}(c)$ is not since it is a sum of the box kernels in 2.16.

It can be shown that in the limit of a large number of samples, the above procedure

will result in reconstructed functions $\hat{f}$ that are calculated using

$$\hat{f}(c) = \int K(c - c')f(c')dc' \tag{2.17}$$

In fact, Equation 2.15 can be though of as a numerical approximation of this relation by making the approximation $f(c) \approx \frac{1}{N}\Sigma\delta(c - c_i)$. This last relation is useful because it allows us to deduce some general properties of a KDE approximation.

2.5.2 KDE Limitations and The Average Number of Contributing Samples

One of the important features of KDE is that there is a fundamental trade-off between the ability to faithfully reproduce f without bias and the uncertainty in $\hat{f}$. Kernels with large supports produce smoother approximations of f since more samples contribute to the value of $\hat{f}$ at each point but this happens at the cost of introducing bias in the reconstruction. Conversely, the bias introduced in $\hat{f}$ is significantly decreased as we choose kernels that have narrow supports and the bias completely disappears in the limit $\varepsilon \to 0$. This bias-smoothness trade-off has long been known [3, 11],in fact has been shown that this is a fundamental limitation that can not be overcome despite some creative attempts [16]. This means that more and more samples are needed if we want to reconstruct a distribution with the minimum amount of bias possible. Although this is a burden in all cases, it is even more problematic when the functions we are interested in are in multiple dimensions.

We will see in the next chapters how KDE is the key to making our proposed variance reduction method practical for simulations of arbitrary time. Unfortunately, we will also see that as the sole approximation in our approach, it is also responsible for the majority of the numerical error in our simulations.

Chapter 3

Variance-Reduced DSMC

In this Chapter we present a variance-reduced DSMC formulation which can simulate arbitrarily small deviations from equilibrium. The variance reduction is achieved using the methodology presented in Chapter 2. In other words, the present Chapter describes how importance weights and Kernel Density Estimation can be seamlessly integrated within the DSMC algorithm, to yield VRDSMC, an efficient low-variance algorithm that differs very little from the original DSMC algorithm. The material presented here has appeared in a more condensed form in [59, 68].

3.1 Method Overview

As we described in Section 1.8, one of our objectives was the development of a variance reduction method which retains the majority of DSMC features. The method described in this Chapter fulfills this requirement. As a result, as we show in Chapter 6, the present formulation can be readily extended to other molecular models; in contrast, extension of LVDSMC to other molecular models – other than the relaxation-time approximation [66]– can be very challenging [58]. Moreover, the formulation presented here does not rely on explicit knowledge of the collision operator, which is advantageous not only because it can be used to simulate more complex collision processes, but also due to its considerable potential to be extended to other particle simulation methods [67, 69].

Our approach is based on the variance reduction technique of control variates discussed in Section 2.2.1. Specifically, our approach utilizes an auxiliary equilibrium simulation that uses the same data as the main DSMC simulation but uses importance weights to account for the difference between the two distributions. This chapter discusses how importance weights defined by initial and boundary conditions evolve in time under the action of Boltzmann dynamics. Unfortunately, as we also saw in Section 2.4, stability problems appear for long simulation times when conditional weight update rules are used. An extensive discussion of how these are overcome in this work is given in Section 3.3.2. Our formulation is validated by comparison to DSMC results for a number of benchmark flows in Section 3.4.

3.2 VRDSMC: Basic Concepts

As we saw in Section 2.2.1, the basic idea behind the present approach is to produce the variance-reduced estimator, $\overline{R}_{VR}$, by writing

$$\overline{R}_{VR} = \overline{R} - \overline{R}_{eq} + \langle R \rangle_{eq} \tag{3.1}$$

and ensuring that $\overline{R}$ and $\overline{R}_{eq}$ are estimated using correlated molecular data, while the distribution f_{eq} (the reference distribution is taken to be some equilibrium) is chosen such that $\langle R \rangle_{eq}$ is known. This concept is illustrated in Figure 3-1 for a simple relaxation problem; the figure shows how actual simulation data [59] of $\overline{R}$, $\overline{R}_{eq}$, and $\langle R \rangle_{eq}$ can be combined to yield the low-uncertainty estimator $\overline{R}_{VR}$. In this particular case, the figure shows a hydrodynamic variable (here $R = c_x^4$) of interest for the relaxation problem outlined in Section 3.2.1.

The major challenge associated with implementing this approach is in the development of a framework which provides molecular data that sample the non-equilibrium single-particle distribution function $f(\boldsymbol{c})$, while at the same time are correlated to the (equilibrium) DSMC data that sample $f_{eq}(\boldsymbol{c})$. To achieve this we have chosen to use

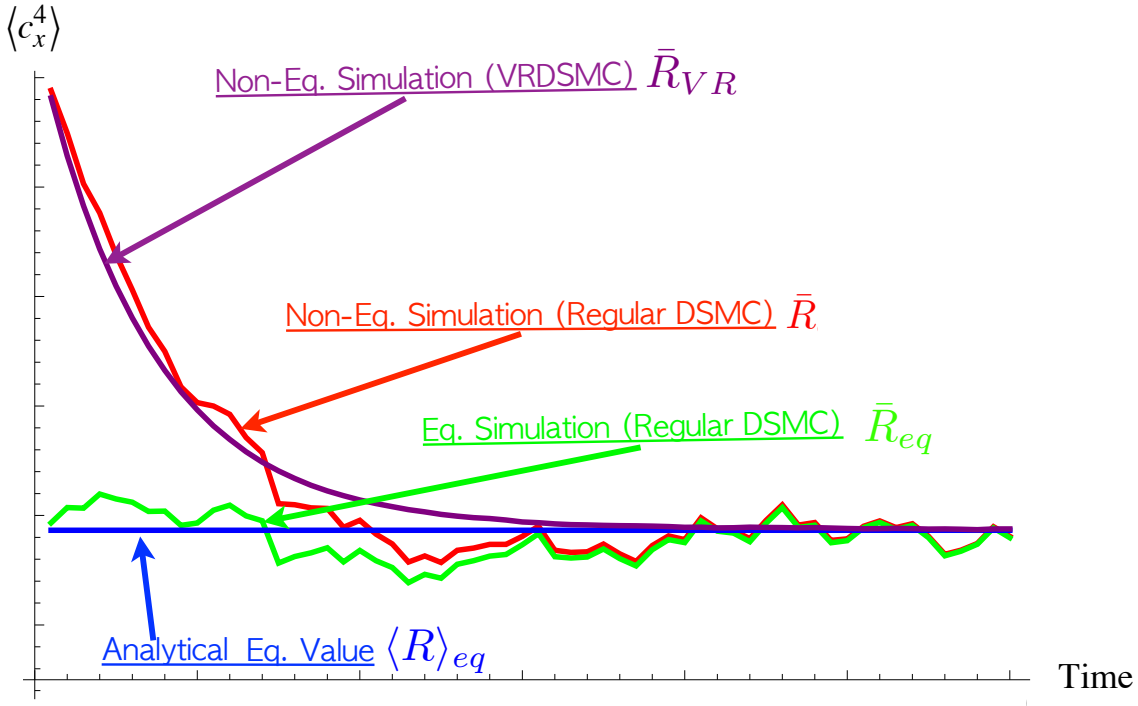

Figure 3-1: Illustration of the variance reduction principle for a molecular relaxation problem [59]. The variance of $\bar{R}_{VR}$ is significantly reduced by replacing the "noisy" estimate $\bar{R}_{eq}$ with its exact expected value $\langle R_{eq} \rangle$.

the importance weight formulation, first introduced in Chapter 2 by defining $W(\boldsymbol{c})$

$$W(\boldsymbol{c}) = \frac{f_{eq}(\boldsymbol{c})}{f(\boldsymbol{c})} \tag{3.2}$$

Using this definition, an estimate of the equilibrium property is

$$\overline{R}_{eq} = \sum_{i=1}^{N_{cell}} W_i R(\boldsymbol{c}_i)$$

where $\boldsymbol{c}_i$ is drawn from $f(\boldsymbol{c})$ and W_i provides a correction for the relative frequency of

each sample $\boldsymbol{c}_i$ in the two distributions. Using this formulation, Equation 3.1 becomes

$$\overline{R}_{VR} = \sum_{i=1}^{N_{cell}} (1 - W_i)\, R(\boldsymbol{c}_i) + \langle R \rangle_{eq}$$

and can be evaluated by sampling the non-equilibrium distribution only, provided the weights W_i are known. In the next section we present a method for initializing the weights W_i and a description of the rules that govern for the dynamical evolution of these weights based on the governing Boltzmann equation.

It is straightforward to verify (recall Section 2.3) that the variance of the estimator $\overline{R}_{VR}$ is significantly smaller than the variance of $\overline{R}$ when the distribution functions f and f_{eq} are close (i.e. $\|W_i - 1\| \ll 1$). In this sense, a Maxwell-Boltzmann equilibrium distribution

$$f_{MB}(\boldsymbol{c}; n_{MB}, \boldsymbol{u}_{MB}, T_{MB}) = n_{MB}(\pi c_{MB}^2)^{-3/2} \exp(-\frac{\|\boldsymbol{c} - \boldsymbol{u}_{MB}\|^2}{c_{MB}^2})$$

where $c_{MB} = \sqrt{2RT_{MB}}$, is a reasonable and convenient choice, since in cases where variance reduction is needed, i.e. when the deviation from equilibrium is small, the parameters n_{MB}, T_{MB} and $\boldsymbol{u}_{MB}$ can usually be easily chosen such that $\| W_i - 1\| \ll 1$.

In this Chapter and the next we will be using a single global reference equilibrium state

$$f_{eq,0}(\boldsymbol{c}) = f_{MB}(\boldsymbol{c}; n_0, \boldsymbol{u}_0, T_0)$$

where n_0,T_0 and $\boldsymbol{u}_0 = \boldsymbol{0}$ are chosen global equilibrium properties throughout the simulation domain. In the interest of brevity, we will use f_{eq} to represent this state when there is no possibility of ambiguity. In Chapter 5 we will introduce local reference states and will use $f_{MB,loc}$ to represent a local equilibrium reference state and $f_{eq,0}$ to represent the global reference state defined above.

3.2.1 Quantifying Variance Reduction For a Simple Case

Sections 2.2 and 2.3 illustrated how we can use importance weights to produce estimators that have variance that decreases as the sampling distribution gets closer to a reference state. In this section we use a relaxation problem to demonstrate that this property still holds for 3-dimensional distributions of molecular velocities. In particular, we will take as an example problem a homogeneous relaxation problem that was presented in [59, 50]. In that work, we studied the relaxation to equilibrium for a gas that is initially in a state

$$f(\boldsymbol{c}, t=0) = \frac{1}{2}\left(f_{MB}(\boldsymbol{c};n_0, \boldsymbol{u}_\alpha, T_0) + f_{MB}(\boldsymbol{c};n_0, -\boldsymbol{u}_\alpha, T_0)\right)$$

with some reference number density n_0 and T_0 and $\boldsymbol{u}_\alpha = \{\alpha c_0, 0, 0\}$. Here α parameterizes the deviation from equilibrium with $f(\boldsymbol{c}, t=0) \rightarrow f_{MB}(\boldsymbol{c}; n_0, \boldsymbol{0}, T_0) = f_{eq,0}(\boldsymbol{c})$ as $\alpha \rightarrow 0$.

We would like to illustrate how the variance of the variance-reduced estimator of $\langle c_x^4 \rangle$ scales with the magnitude of α and will restrict our attention for now at the initial point of the simulation (i.e. $t = 0$), since it is the only point that we have an analytical formula for both f_{MB} and f.

The estimator $\overline{c_x^4}_{VR}$ is given by

$$\overline{c_x^4}_{VR} = \overline{c_x^4} - \overline{c_x^4}_{eq,0} + \left\langle c_x^4 \right\rangle_{eq,0}$$

and can be evaluated using

$$\overline{c_x^4}_{VR} = N_{Eff} \sum_{i=1}^{N_{cell}} \left(1 - W_i\right) c_{x,i}^4 + \left\langle c_x^4 \right\rangle_{eq,0} \text{ where } \boldsymbol{c}_i \sim f(\boldsymbol{c})$$

To understand the variance of this estimator let us calculation the variance of the individual terms $(1-W)c_x^4$ as a function of α as follows

$$\sigma^2\left\{\left(1 - W\right) c_x^4\right\} = \int \left(1 - W(\boldsymbol{c})\right)^2 \left(c_x^4\right)^2 f(\boldsymbol{c}) d\boldsymbol{c} - \left(\int \left(1 - W(\boldsymbol{c})\right) c_x^4 f(\boldsymbol{c}) d\boldsymbol{c}\right)^2 \quad (3.3)$$

We can explicitly evaluate this by numerically evaluating the above integrals using the explicit formula of f and f_{MB}.We plot the variance versus the non-equilibrium parameter in Figure 3-2 which clearly shows that $\sigma^2\{(1-W)c_x^4\} \to 0$ as $\alpha \to 0$.

In fact Figure 3-2 shows that $\sigma^2\{(1-W)c_x^4\} \propto \alpha^4$. This is because it can be shown that $\langle c_x^4 \rangle |_{t=0} - \langle c_x^4 \rangle_{eq} = 3\alpha^2 \frac{2kT}{m}$ and so, in this particular case, we have a signal that is proportional to α^2.

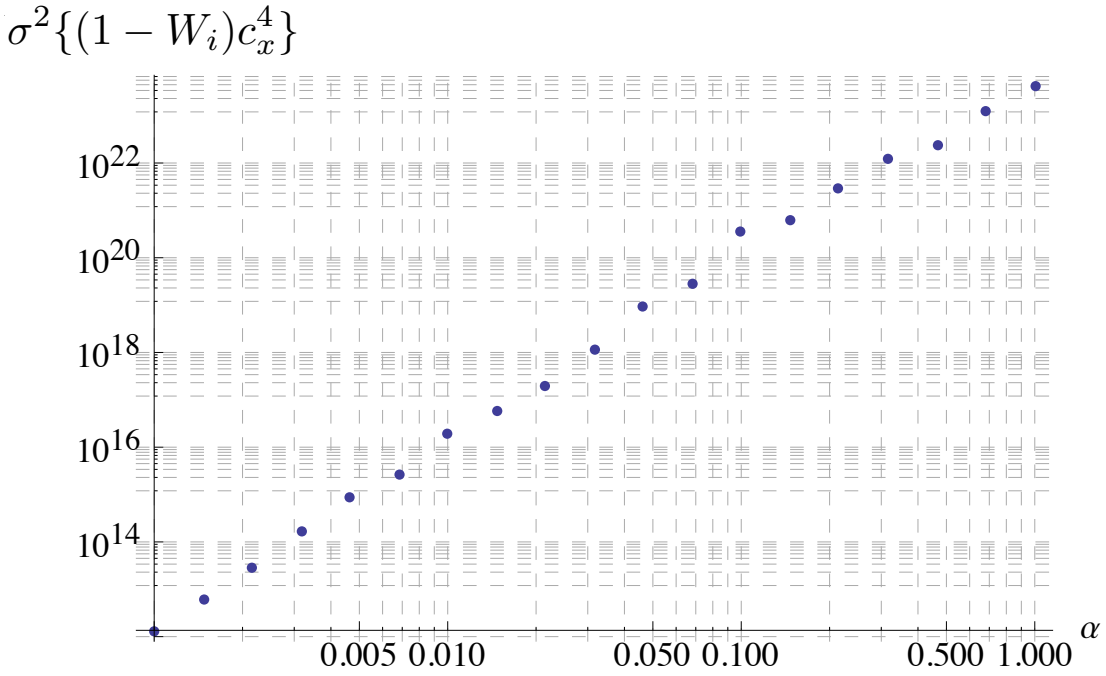

Figure 3-2: Variance of variance-reduced estimator $\overline{c_x^4}_{VR}$ vs. α for the homogeneous relaxation problem of Section 3.2.1. These plots were produced by numerically evaluating Equation 3.3; for the sake of simplicity, we take $N_{Eff} = 1$.

3.3 VRDSMC Implementation

Having introduced the basics of using variance reduction to provide low-uncertainty estimators, we will proceed in this section to incorporate them in DSMC.

In the present Chapter as well as the next, we consider a hard-sphere gas in the absence of external fields. As will be clear from the analysis below, both assumptions can be relaxed in a fairly straightforward manner. We refer the reader to Chapter 6 as well as [67] for a more detailed exploration of these issues.

3.3.1 Weight Update Rules

In VRDSMC, the difference between the equilibrium and non-equilibrium distributions is described by the set of weights $\{W_i\}$ giving the relative frequency of finding a particle of velocity $\boldsymbol{c}_i$ in the equilibrium simulation relative to the non-equilibrium simulation. As the simulation evolves in time, the non-equilibrium distribution and the particle velocities $\boldsymbol{c}_i$ (sampling this distribution) change. Below, we develop the rules that will be used to describe the evolution of particle weights in response to these changes.

3.3.1.1 Initialization

The initial values of the weights can be readily determined since initially both f and f_{eq} are explicitly known (i.e. the initial distributions of the simulated gas is known for both the equilibrium and non-equilibrium simulation). Specifically, when DSMC particles are initialized at the start of the simulation, the weight corresponding to each particle is calculated using the definition 3.2.

3.3.1.2 Boundary Conditions

Boundary conditions are in some ways similar to initial conditions in the sense that, in typical applications of interest, particle velocities (and thus weights) are redrawn from a pre-specified distribution when interacting with a wall. The boundary conditions associated with the Boltzmann equation in the applications of interest here are typically of the no-flux type [6]. In Chapter 4 we give a longer discussion of boundary conditions; for now let us consider a fully accommodating no-flux boundary. In this case, the boundary condition for particles that come in contact with the boundary is

the distribution

$$f_W(\boldsymbol{c}) = f_{MB}(\boldsymbol{c}; n_W, \boldsymbol{u}_W, T_W) = \frac{n_W}{(\pi c_W^2)^{3/2}} \exp\left(\frac{-\|\mathbf{c} - \boldsymbol{u}_W\|^2}{c_W^2}\right) = n_W g_W(\boldsymbol{c})$$

parameterized by the wall properties, namely the "wall number density" n_W and the other wall parameters $\boldsymbol{u}_W$, T_W and $c_W = \sqrt{2kT_W/m}$. The wall number density may be thought of as the number density of an equilibrium gas that would be in contact with the boundary; it is determined from mass conservation at the wall, namely

$$-n_W \int_{\mathbf{c}\cdot\mathbf{n}>0} \mathbf{c} \cdot \boldsymbol{n} g_W(\boldsymbol{c}) d\boldsymbol{c} = \int_{\mathbf{c}\cdot\mathbf{n}<0} \boldsymbol{c}\cdot\boldsymbol{n} f(\boldsymbol{c}) d\boldsymbol{c} \tag{3.4}$$

where $\boldsymbol{n}$ denotes the wall normal pointing into the gas. Here for simplicity we have assumed that $\boldsymbol{n} \cdot \boldsymbol{U}_W = 0$ the more general expression can be found in [46]. If n_W is known, the weights of re-emitted particles can be readily calculated from Equation 3.2; one such case is isothermal low-speed flows with no flow component normal to the wall which can be shown to be a constant-density flow. With such an arrangement weights are completely determined at the walls and as we will see in Section 3.3.2, will result in superior stability properties especially at high Kn.

In the more general case, n_W may need to be explicitly calculated using equation Equation 3.4 which we will explore more fully in Chapter 4. In this Chapter we apply this boundary condition by using the following process: At the beginning of every timestep we assume a value for n_W, e.g. $n_W = n_0$, and particle weights for particles colliding with the wall are assigned based on this value. At the end of the timestep the total weight of the particles that collided with the wall during that timestep is tallied and the weight of particles re-emitted from the wall rescaled such that there is no net change in the total equilibrium particle weights due to the wall interaction. In Chapter 4 we show that this approach is numerically equivalent to the more general approach derived there.

3.3.1.3 The Advection Sub-step

During the advection sub-step the auxiliary simulation needs to integrate

$$\frac{\partial f_{eq}}{\partial t} + \boldsymbol{c}\cdot\frac{\partial f_{eq}}{\partial \boldsymbol{x}} = 0 \tag{3.5}$$

Making the substitution $f_{eq} = Wf$ we obtain

$$f(\frac{\partial W}{\partial t} + \boldsymbol{c}\cdot\frac{\partial W}{\partial \boldsymbol{x}}) + W(\frac{\partial f}{\partial t} + \boldsymbol{c}\cdot\frac{\partial f}{\partial \boldsymbol{x}}) = 0$$

Since $f > 0$ and $\frac{\partial f}{\partial t} + \boldsymbol{c}\cdot\frac{\partial f}{\partial \boldsymbol{x}} = 0$ we conclude that, during the advection step, the weights need to satisfy

$$\frac{\partial W}{\partial t} + \boldsymbol{c}\cdot\frac{\partial W}{\partial \boldsymbol{x}} = 0$$

or in other words, weights are advected with their corresponding particles.

3.3.1.4 Collision Sub-step

In this section we present a derivation based on the standard hard-sphere form of the collision integral; extension to other collision models (e.g. variable hard sphere [12]) directly follows. An alternative derivation based on conditional probability arguments is presented in Chapter 6.

During the collision sub-step, the auxiliary simulation integrates the equilibrium version of Equation 1.6, namely

$$\left[\frac{\partial f_{eq}}{\partial t}\right]_{Collision} = \frac{1}{2}\int\int\int \left(\delta_1' + \delta_2' - \delta_1 - \delta_2\right) W_1 W_2 f_1 f_2 c_r \sigma d\Omega d\boldsymbol{c}_1 d\boldsymbol{c}_2$$

To facilitate the interpretation of this equation within the context of the DSMC

collision algorithm, we rearrange it in the following form:

$$\left[\frac{\partial f_{eq}}{\partial t}\right]_{Collision} = \frac{MX}{2}\int\int\int\left(\delta_1' + \delta_2' - \frac{\delta_1}{W_2} - \frac{\delta_2}{W_1}\right) W_1 W_2 f_1 f_2 \frac{c_r}{MX}\sigma d\Omega d\boldsymbol{c}_1 d\boldsymbol{c}_2 + \frac{MX}{2}\int\int\int\left(-\delta_1 - \delta_2 + \frac{\delta_1}{W_2} + \frac{\delta_2}{W_1}\right)\frac{\frac{c_r}{MX}}{\left(1-\frac{c_r}{MX}\right)} W_1 W_2 f_1 f_2 \sigma (1 - \frac{c_r}{MX}) d\Omega d\boldsymbol{c}_1 d\boldsymbol{c}_2 \tag{3.6}$$

where MX is an upper bound for Wc_r. Using an importance sampling interpretation [36, 52, 53, 61], the first term of this equation can be thought of as an event occurring with probability $\hat{c}_r = c_r/MX$, while the second one as an event occurring with probability $(1-\hat{c}_r)$. In other words, provided collisions are accepted with probability $\hat{c}_r$ (and thus rejected with probability $1-\hat{c}_r$) in the DSMC calculation, [61] provides a means of connecting the complementary events of collision acceptance and rejection in the "main" DSMC collision routine, with weight evolution in the auxiliary equilibrium calculation.

To make this more concrete, consider a collision-candidate particle pair with velocities $\boldsymbol{c}_1$and $\boldsymbol{c}_2$ and weights W_1 and W_2, respectively. If the collision is accepted in DSMC, according to the first term in Equation 3.6, a particle pair with velocities $\boldsymbol{c}_1'$ and $\boldsymbol{c}_2'$ and weights W_1W_2 should be created, in addition to a pair of negative particles with velocities $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ and weights W_1 and W_2, respectively. Note that, by design, the negative particles cancel the colliding particles and thus the collision proceeds by the update $\boldsymbol{c}_1 \rightarrow \boldsymbol{c}_1'$, $W_1 \rightarrow W_1W_2$ and $\boldsymbol{c}_2 \rightarrow \boldsymbol{c}_2'$, $W_2 \rightarrow W_1W_2$. Since the update $\boldsymbol{c}_1 \rightarrow \boldsymbol{c}_1'$, $\boldsymbol{c}_2 \rightarrow \boldsymbol{c}_2'$ is part of the original DSMC algorithm, we conclude that if the collision is accepted in DSMC the weight update is $W_1, W_2 \rightarrow W_1W_2$. In the case of a rejected collision (in DSMC), the second term in Equation 3.6 implies that $W_1W_2\frac{\hat{c}_r}{1-\hat{c}_r}$ negative particles with velocities $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$, as well as $W_1\frac{\hat{c}_r}{1-\hat{c}_r}$ particles with velocity $\boldsymbol{c}_1$ and $W_2\hat{c}_r/(1-\hat{c}_r)$ particles with velocity $\boldsymbol{c}_2$ need to be created. Combining these with the colliding particles we obtain a net of $W_1(1-W_2\hat{c}_r)/(1-\hat{c}_r)$ at $\boldsymbol{c}_1$and $W_2(1-W_1\hat{c}_r)/(1-\hat{c}_r)$ at $\boldsymbol{c}_2$.

In summary, if the DSMC collision is accepted, the colliding particle weights are

Table 3.1: Summary of weight update rules.

	In	**Intermediate Steps**	**Final Result**
1.			
Accepted probability= $\hat{c}_r$	W_1 at $\boldsymbol{c}_1$ and W_2 at $\boldsymbol{c}_2$	Create: W_1W_2 at $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ Annihilate: W_1 at $\boldsymbol{c}_1$ and W_2 at $\boldsymbol{c}_2$	W_1W_2 at $\boldsymbol{c}_1$ and W_1W_2 at $\boldsymbol{c}_2$
Rejected probability=$1-\hat{c}_r$	W_1at $\boldsymbol{c}_1$ and W_2 at $\boldsymbol{c}_2$	Create: $\frac{W_1\hat{c}_r}{1-\hat{c}_r}$ at $\boldsymbol{c}_1$ and $\frac{W_2\hat{c}_r}{1-\hat{c}_r}$ at $\boldsymbol{c}_2$ Annihilate: $\frac{W_2W_1\hat{c}_r}{1-\hat{c}_r}$ at $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$	$\frac{1-W_2\hat{c}_r}{1-\hat{c}_r}$ at $\boldsymbol{c}_1$ and $\frac{1-W_1\hat{c}_r}{1-\hat{c}_r}$ at $\boldsymbol{c}_2$

updated as W_1,$W_2 \rightarrow W_1W_2$; if the DSMC collision is rejected, the candidate particle weights are updated as $W_1 \rightarrow W_1(1-W_2\hat{c}_r)/(1-\hat{c}_r)$ and $W_2 \rightarrow W_2(1-W_1\hat{c}_r)/(1-\hat{c}_r)$. These steps are summarized in Table 3.1.

3.3.2 Stability Considerations

In a manner analogous to the coin simulation in Section 2.4, the implementation of this algorithm reveals a numerical stability issue. In particular, after a few collision times the variance of particle weights is observed to diverge (i.e. individual particle weights tend to either 0 or ∞), which results in loss of variance reduction. This behavior not only is similar to what we saw in Section 2.4, it also has many similarities to the stability problems observed in other variance-reduced particle methods [49, 65, 35], despite significant differences in formulation. Furthermore, we find that the instability appears in collision dominated flows ($Kn < 1$), while collisionless and near collisionless ($Kn \gtrsim 3$) calculations are much more stable. In our experience, this issue is one of the biggest challenges associated with this approach as well as previous variance reduction approaches.

As in Section 2.4, in the present formulation, this behavior is a result of the particular collision weight update rules used. In collisionless flows this issue simply does not arise, while in near-collisionless flows ($Kn \gg 1$) with fixed density boundaries

particles reach the walls (where their weights are reassigned) before their weights diverge.

As in Section 2.4, we have developed an approach for stabilizing the calculation based on KDE. As we introduced before, KDE is used to reconstruct the distribution functions at the end of each timestep and is explained in the next subsection.

3.3.2.1 Stabilization Using Kernel Density Estimation

To proceed, let us reconstruct f using $\hat{f}(\boldsymbol{c}) = \int K(\boldsymbol{c}-\boldsymbol{c}')f(\boldsymbol{c}')d\boldsymbol{c}' \simeq f(\boldsymbol{c})$, for some appropriate kernel $K(\boldsymbol{c}-\boldsymbol{c}')$. If such an approach is used, from the weight definition 3.2 we can make the approximation

$$\hat{W}(\boldsymbol{c}) = \frac{\hat{f}_{eq}(\boldsymbol{c})}{\hat{f}(\boldsymbol{c})} = \frac{\int K(\boldsymbol{c}-\boldsymbol{c}')W(\boldsymbol{c}')f(\boldsymbol{c}')d\boldsymbol{c}'}{\int K(\boldsymbol{c}-\boldsymbol{c}')f(\boldsymbol{c}')d\boldsymbol{c}'} \tag{3.7}$$

In the work presented here we have used the normalized kernel

$$K(\boldsymbol{c}-\boldsymbol{c}') = \begin{cases} \frac{1}{4/3\pi\epsilon^3 c_0^3} & if||\boldsymbol{c}-\boldsymbol{c}'|| < \varepsilon c_0 \\ 0 & otherwise \end{cases}$$

Using Equations 1.4 and 3.7, we obtain an expression for the "reconstructed" weight

$$\hat{W}_i = \frac{1}{||S_i||}\sum_{k\in S_i} W_k \tag{3.8}$$

where W_k are the weights obtained using the collision update rules detailed in Table 3.1. Here, S_i denotes the set of particles that are within a sphere of radius εc_0 centered on particle i in velocity space, and we denote the number of such particles with $\overline{||S_i||}$. Particles within S_i can be found using a KD Tree [48] in $O(Log(N_{cell}))$ operations, implying that the overall algorithm will scale as $O(N_{cell}Log(N_{cell}))$ instead of $O(N_{cell})$ for regular DSMC. In our implementation we find the set S_i using a non-uniform velocity space binning that is both faster and simpler to code which is detailed in Section A.1.

This procedure introduces a new discretization parameter, ε, that in general affects

the results of our simulation as we will discuss in more detail in Chapter 8. A large ε means that we average over many particles which improves the stability of the calculation but also introduces numerical error. Because of this, the numerical accuracy of VRDSMC is directly affected by the averaging radius parameter ε which we would ideally want to make as small as possible. On the other hand, if ε is too small, there will be a small number of particles in each estimate of $\hat{W}_i(\boldsymbol{c})$ and in such a situation we will need to increase N_{cell} if we are to keep $\overline{\|S_i\|}$ constant. In fact, by numerical experimentation with problems close to equilibrium, we were able find that the relation between $\overline{\|S_i\|}$ and ε for a stable calculation is

$$\overline{||S_i||} \simeq 0.26 N_{cell} \varepsilon^3$$

In other words stability strongly depends on the average number of particles in a sphere of radius εc_0 and not on the total number of particle in a cell. For example, a 3 order of magnitude increase in N_{cell}, while holding $\overline{\|S_i\|}$ constant, only results in a 20% change in the average weight variance, $\sigma^2\{W_i\}$, in a simple shear (Couette) flow. Figure 3-3, on the other hand, shows that $\overline{\|S_i\|}$ has a strong effect on stability for various Knudsen numbers. We also see that stability, defined as $\frac{\sigma^2\{W_i\}}{\sigma^2\{W_i\}|_{Kn=10,||S_i||=0}} = O(1)$, is strongly affected not only by $\overline{\|S_i\|}$ but also by Kn. As expected, increasing $\overline{\|S_i\|}$ (by increasing N_{cell} for a fixed ε) improves the stability; moreover, flows characterized by $Kn > 1$ are stable in a wide range of values of $\overline{\|S_i\|}$, while flows with $Kn < 1$ typically require $\overline{\|S_i\|} \gg 1$ for stability. We close by noting that the data of Figure 3-3 were generated using the analytically known value of n_W (low-speed Couette flow is essentially isothermal). The effect of more complex flow conditions on stability is discussed in the next Chapter.

Although we have decided to employ KDE reconstruction of weights to stabilize our simulation, it is not clear how to choose which samples (particles) to employ for this reconstruction. After much experimentation we found that a good balance between bias and stability is achieved if we only apply of KDE to *accepted* (for col-

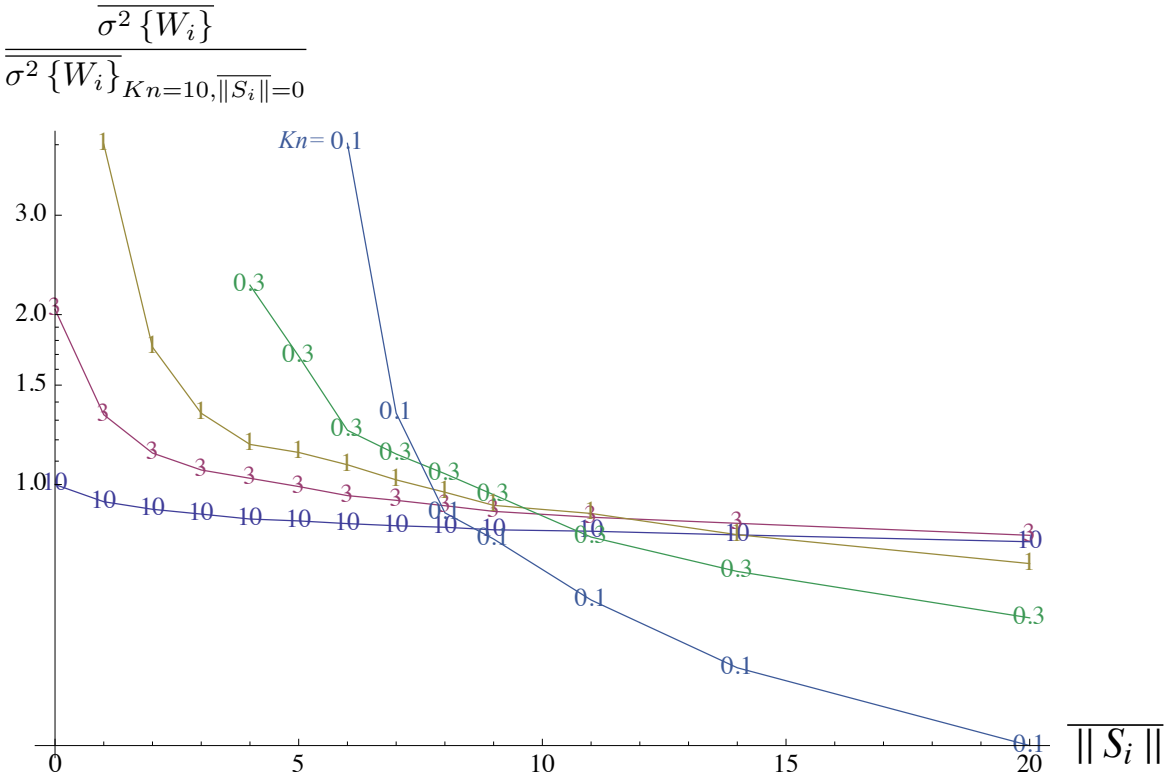

Figure 3-3: Steady-state average weight variance in a Couette flow as a function of $\overline{\|S_i\|}$ and Kn. Note that we normalize our stability measure by the average variance of a $Kn = 10$ flow problem since it requires no stabilization when we have walls with fixed density. Note stability when $\overline{\|S_i\|} = 0$ (i.e. no KDE is performed) due to the fixed n_W in contrast to Figure 5-5.

lision) particles. In contrast, original attempts [59] applied the KDE approximation to all particles in the domain after the end of every collision step; in many cases, this approach requires more than $100,000$ particles per cell for accurate results. By introducing further refinements in the following Chapters, we will show that VRDSMC can produce accurate and stable calculations with $N_{cell} < 3,000$ for many problems of interest in the range $0.1 < Kn < 10$.

A flow chart of the proposed simulation method is shown in Figure 3-4; modifications to the original DSMC algorithm are shown in blue highlight.

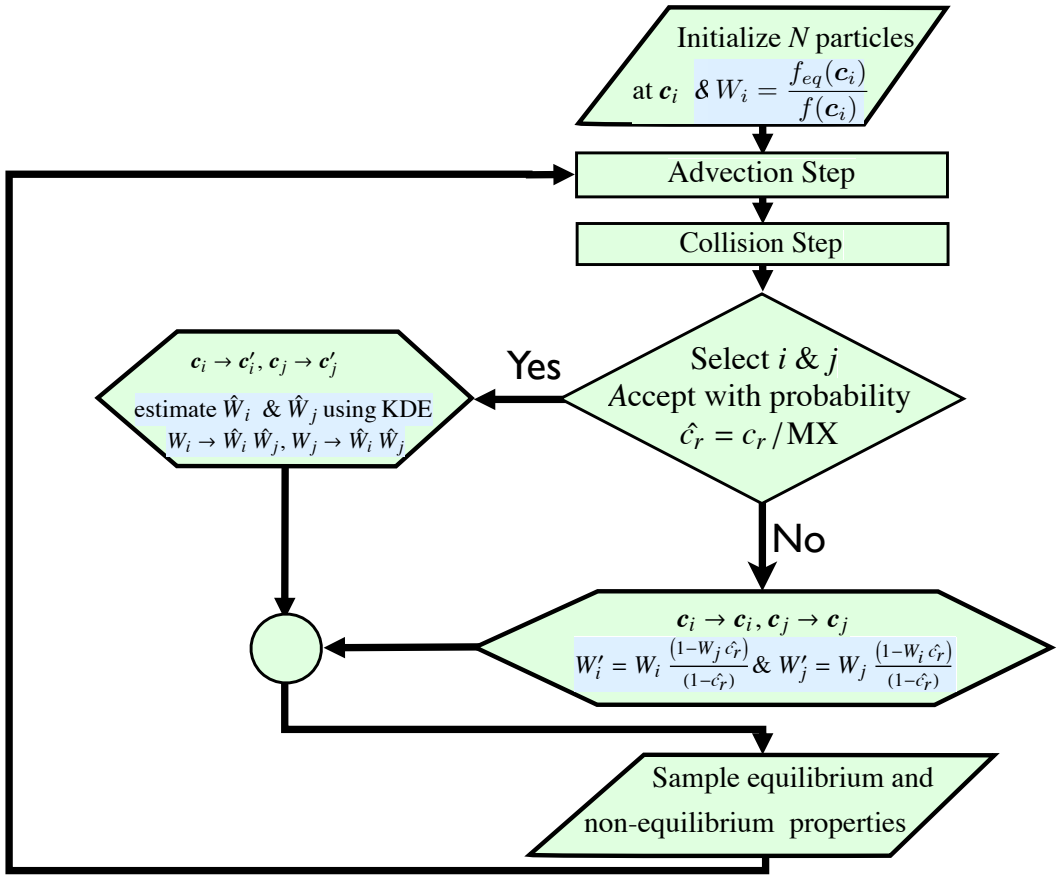

Figure 3-4: Flowchart of VRDSMC as described in this Chapter.

3.4 Validation and Computational Performance

An extensive validation of a variant of VRDSMC applied to the homogeneous relaxation problem (like the one described in Section 3.2.1) can be found in a previous publication [59] of that study. Here we note the most important result, namely that bias decreases with ε. This is of course not a surprising result since the use of the reconstructed distribution $\hat{f}$ instead of f is the only approximation we make in our scheme (in addition to the numerical approximations inherited from DSMC).

In this section, we validate VRDSMC using various one-dimensional problems for both transient and steady situations to show that similar trends hold.

3.4.1 Validation Using One-dimensional Problems

Figure 3-5 is a schematic of the one-dimensional geometry that we will consider in this section and the reminder of this Thesis. We consider a dilute gas of density n_0 between two fully accommodating, parallel plates (walls) a distance H apart in the y direction. The gas is initially stationary and at equilibrium with temperature T_0 and velocity $\boldsymbol{u} = \mathbf{0}$. The coordinate system that we adapt is such that the walls move in the x direction and the domain is divided into N_y cells in the y direction with width $\Delta y = \frac{H}{N_y}$.

Example 1: 1D steady state Couette flow problem We first consider a simple Couette flow in which the plates move in the x direction with velocities $U_W = \pm 0.085 c_0$. The wall velocity magnitude was chosen to minimize the DSMC cost since as shown below and before, VRDSMC can resolve arbitrarily small flow disturbances at fixed cost. Figure 3-6 shows the resulting flow field for $Kn = 1$ and Kn=0.1 at steady state. The discrepancy between the results of VRDSMC (as presented in this chapter) and DSMC is less than 1% for both cases. For the $Kn = 1$, the calculation uses $N_{cell} = 500$ and $\overline{\|S_i\|} = 10$ which corresponds to $\varepsilon \simeq 0.43$. In contrast, for the same $\overline{\|S_i\|}$, we find that for $Kn = 0.1$ we are only able to reproduce the DSMC solution to within $\sim 1\%$ by using $N_{cell} = 50,000$, which corresponds to

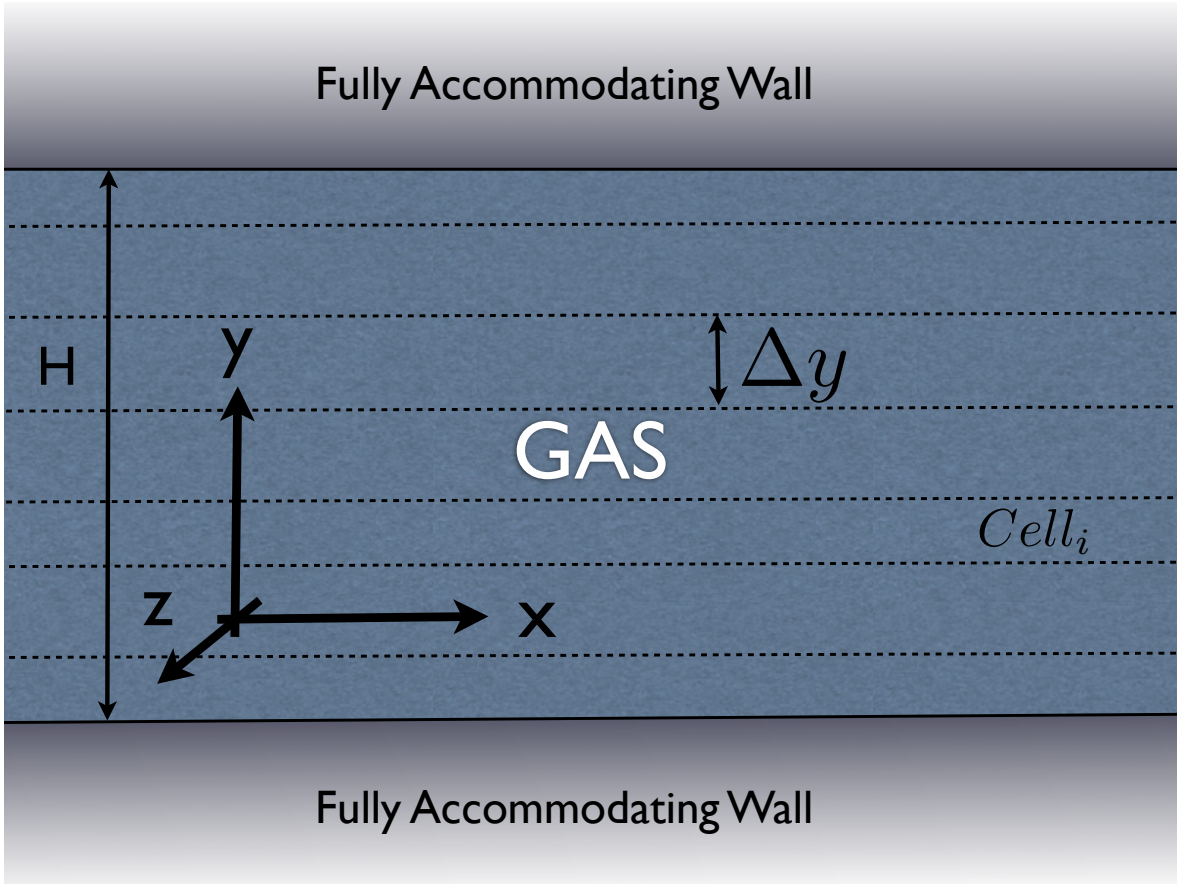

Figure 3-5: Schematic of the physical setup of our one-dimensional validation problems.

$\varepsilon \simeq 0.09$.

We note that for $U_W \ll c_0$, Couette flow problems are, to a good approximation, isothermal. We could though simplify these calculations by assuming $n_W = n_0$. This action would also have the effect of requiring no KDE for stability for $Kn \gtrapprox 2.5$.

Example 2: 1D unsteady boundary heating problem In these examples we consider the transient response of the gas to an impulsive boundary temperature change. Specifically, at time $t = 0$, the wall temperatures impulsively change from T_0 to $T_0 = \pm 0.033 T_0$. Figure 3-7 shows the normalized results for the temperature (T), the heat-flux in the wall-normal direction (q_y), the density (ρ) and the y component of flow velocity (u_y) for $Kn = 10$ and $\overline{\|S_i\|} = 10$; ρ, T and u_y and q_y are normalized

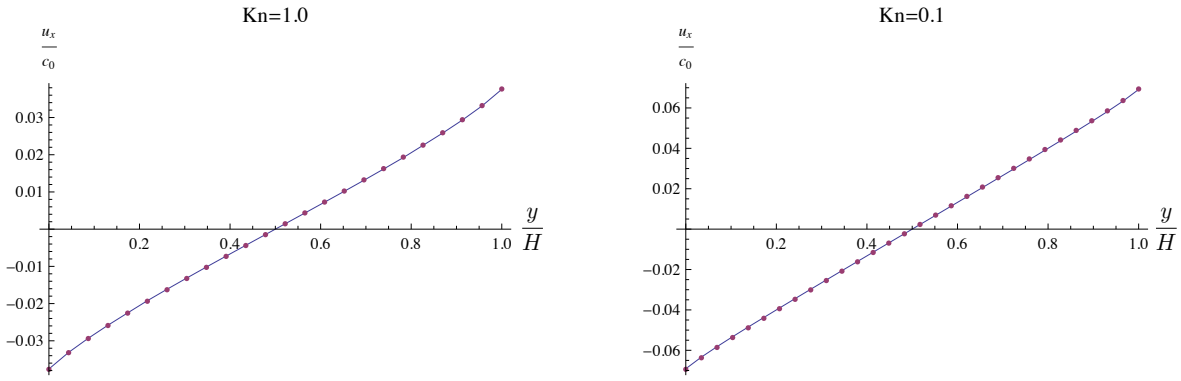

Figure 3-6: Steady state DSMC results (dots) and VRDSMC results (solid lines) for Couette flow. Left Kn=1.0 with $\overline{\|S_i\|} = 10$ and $N_{cell} = 500$; Right: $Kn = 0.1$ with $\overline{\|S_i\|} = 10$, $N_{cell} = 50,000$.

by ρ_0, T_0, c_0 and ρc_0^3 respectively.

Figure 3-8 shows similar transient results for the $Kn = 1.0$ case, where VRDSMC can reproduce the DSMC solution within 1% using $\overline{\|S_i\|} = 10$ and $N_{cell} = 500$.

3.4.2 Magnitude of Variance Reduction Compared to DSMC

Figure 3-9 shows a comparison of the relative statistical uncertainty in the flow velocity, $\sigma_u = \sigma/U_W$, achieved by the two methods when simulating a steady Couette flow at $Kn = 1$. Here, $\sigma = \sqrt{\sigma^2\{u\}}$ is the standard deviation in the flow velocity samples. The figure shows that VRDSMC exhibits a constant relative statistical uncertainty for $U_W/c_0 \ll 1$, as expected, and in sharp contrast to DSMC whose statistical uncer-

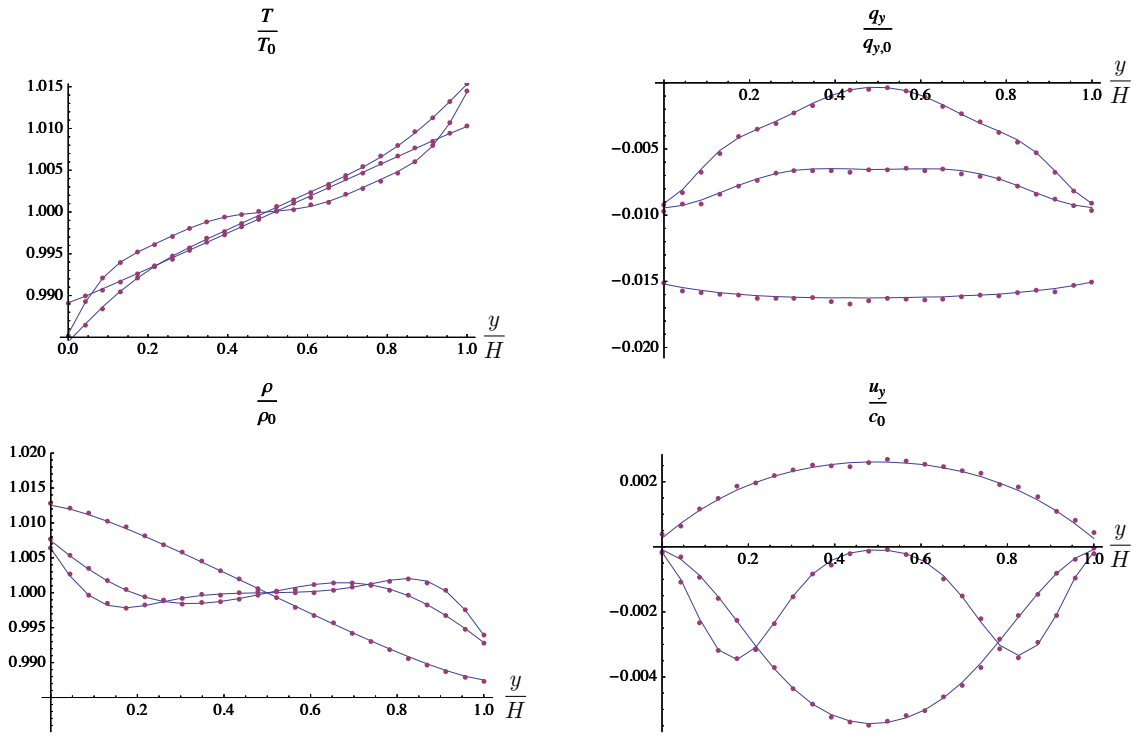

Figure 3-7: Transient results for an impulsive boundary temperature change problem for $Kn = 10$ and $\overline{\|S_i\|} = 10$. Solid lines denote VRDSMC results while DSMC results are shown in dots. The snapshots shown correspond to $t = 5, 10, 40\Delta t$ where $\Delta t = \frac{1}{240}\sqrt{\pi}\lambda/(2c_0)$.

tainty for $U_W/c_0 \ll 1$ is dominated by equilibrium fluctuations, resulting in $\sigma_u \propto \frac{1}{U_W}$ in this limit [29]. We can thus conclude that VRDSMC provides a very considerable amount of variance reduction and thus computational benefit compared to DSMC for the same number of particles per cell.

3.4.3 Approximation Error and Limitations

Although, as shown in Figure 3-3, VRDSMC is unconditionally stable at $Kn \gtrsim 3$ when the boundary has a prescribed density (n_W), for $3 \gtrsim Kn$ a KDE procedure is required which introduces numerical error (bias). This error can be decreased by

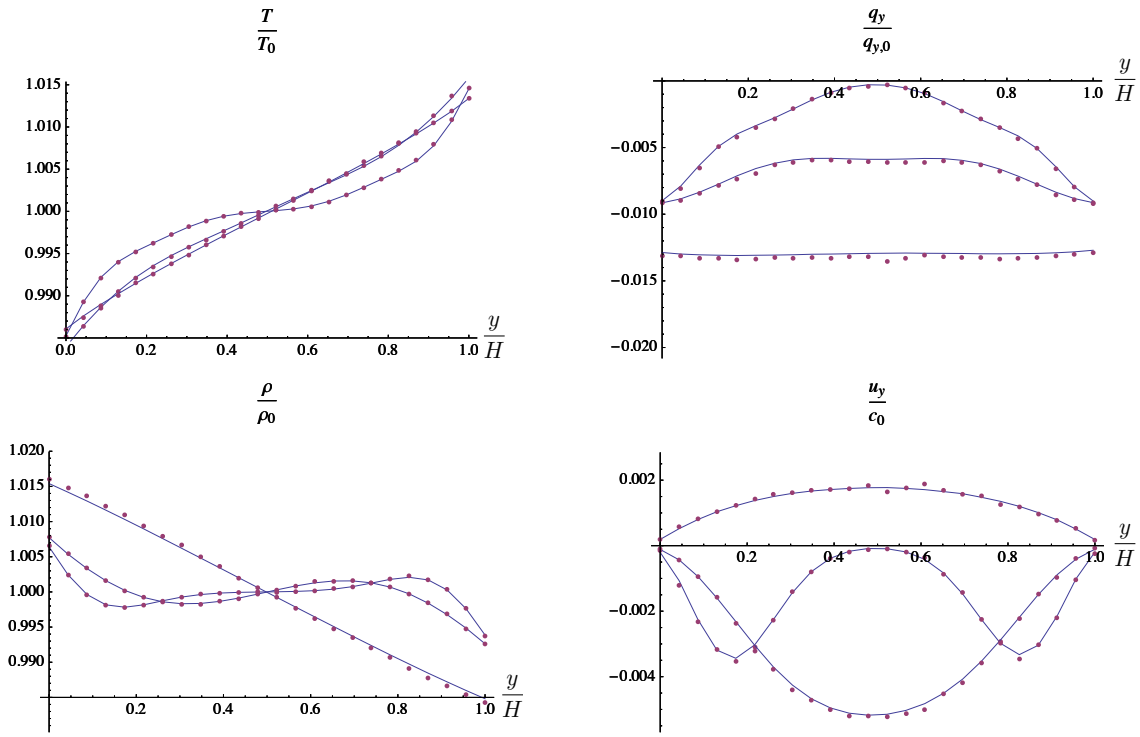

Figure 3-8: Transient results for an impulsive boundary temperature change problem for Kn= 1.0. Solid lines denote VRDSMC results with $\overline{\|S_i\|} = 10$ and $N_{cell} = 500$; DSMC results are shown in dots. The snapshots shown correspond to $t = \{5, 10, 40\}$ where $\Delta t = \frac{1}{24}\sqrt{\pi}\lambda/(2c_0)$.

reducing the discretization parameter ε, at the expense of requiring a larger N_{cell} for stability (recall that $\overline{\|S_i\|} \gg 1$ is required for small Kn). In other words, as already explained in the previous sections, although accurate low-Kn calculations are feasible, they do require large numbers of particles per cell. This requirement is almost non-existent in DSMC, which can provide very accurate solutions with as few as 100 particles per cell, independently of the Knudsen number. We will see in the next Chapters how this constraint can be substantially reduced for most problems of interest by using local equilibrium reference states when performing collisions.

The requirement of large N_{cell} for accuracy is not very limiting in one-dimensional

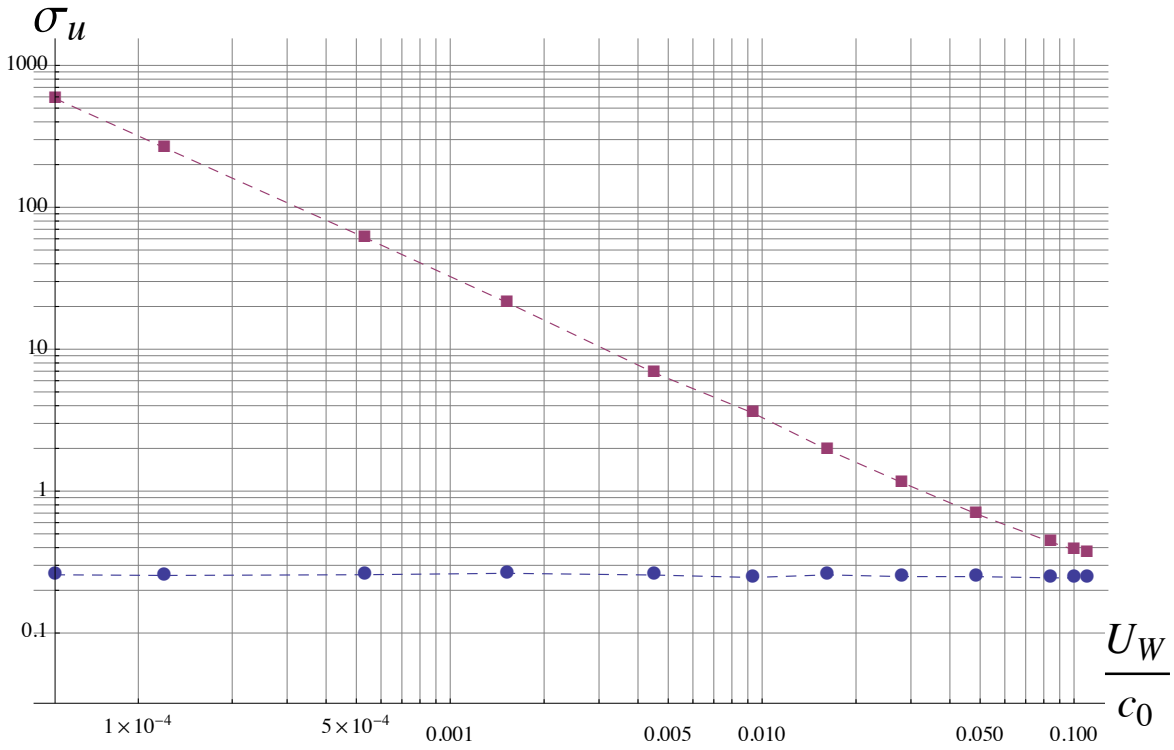

Figure 3-9: Comparison of the relative statistical uncertainty ($\sigma_u = \sqrt{\sigma^2\{u\}}/U_W$) of the DSMC (squares) and VRDSMC (circles) methods for different values of wall velocities. Results are for steady Couette flow at $Kn = 1.0$ for 500 particles per cell at steady state. We clearly see that VRDSMC has constant relative error while the relative error of DSMC increases as $U_W \to 0$.

flows, but can be limiting in higher dimensions if the Knudsen number is small in all dimensions. On the other hand, these large numbers of particles do contribute towards reducing the already small statistical uncertainty of the calculation. In other words, if low statistical uncertainty calculations are required and a large number of particles (or ensembles) were to be used in DSMC, then VRDSMC provides the full benefit shown in Figure 3-9. It should also be noted that our numerical experiments have shown that the numerical error associated with this version VRDSMC is not strongly affected by other discretization parameters (e.g. Δy) and thus provided ε remains small, much like DSMC [20, 17, 21], accurate solutions can be obtained with

fairly coarse grids. Consequently, the total number of particles that are needed for an accurate simulation is not excessively large. We will return to this issue in Chapter 6 where we give a more detailed account of the tradeoffs associated with VRDSMC and discuss how it compares to other simulation methods.

3.5 Chapter Review

In this Chapter we have presented a basic algorithm of a variance-reduced version of DSMC whose main ingredients are importance weights and Kernel Density Estimation. The VRDSMC method was validated using DSMC results for a number of different flows for $0.1 \leq Kn \leq 10$. Our numerical results show that provided a sufficient number of particles is used, significant variance reduction is achieved with little additional discretization error. However, these results also show that in the most general case a basic trade-off between accuracy (requiring a small ε) and stability (requiring a large $\overline{\|S_i\|}$) exists for a given number of simulation particles N_{cell}. Consequently, a large number of particles may be needed to accurately simulate small Knudsen number flows but, as we have shown above, once we have a sufficient number of particles, all problems of interest ($0.1 \leq Kn \leq 10$) can be simulated with small numerical error.

It is interesting to note that for most problems, stable simulations using VRDSMC can be performed without using KDE weight reconstruction if our interest is limited to transient flows when $t \lesssim 3$ mean collision times, provided we use the appropriate boundary condition (described in Section 6.2.3). In this case VRDSMC will have nearly identical execution times and no measurable discrepancy between the two simulation methods.

In Chapter 5, we introduce the concept of an adjusted equilibrium reference state to dramatically reduce the required number of particles in a cell for a given bias and variance. Therefore, the final version of VRDSMC that we will arrive at will not have many of the limitations that we discussed here and it will become practical to simulate kinetic flows in a very wide spectrum of Kn at reasonable cost. As stated

before, one major advantage of the proposed VRDSMC method is that it requires essentially no modification of the DSMC algorithm and introduces relatively little additional complexity. Moreover, it can be easily extended to other collision models and processes (e.g. chemical reactions). We will return to these themes in more detail in Chapter 6.

Chapter 4

Detailed Treatment of the No-Flux Boundary Condition

In this Chapter we give a more detailed description no-flux boundary conditions associated with solid boundaries (walls). In particular, starting from the definition of the no-flux boundary condition we will derive the weight update rules associated with interaction with the wall. We will restrict our discussion here to fully accommodating walls (the Maxwell accommodation model [6]) at arbitrary temperature T_W moving with a speed U_W in a direction parallel to the plane of the wall. Extension to non-accommodating walls or ones that are moving in the wall-normal direction is fairly straightforward, and is not presented here. Likewise, dealing with a constant pressure open wall boundary conditions is also a simple extension of the procedure used in regular DSMC calculations (see for example [31]); in a few words one can use the explicit form of the open wall boundary distributions f_W and $f_{W,eq}$ to evaluate weights of new particles entering the domain. Such boundary conditions are in fact expected to contribute to simulation stability since they reset particle weights (outgoing particles are deleted).

To proceed with the problem of interest, let the Maxwell-Boltzmann distribution associated with the boundary be written as

$$f_W(\boldsymbol{c}) = n_W g_W(\boldsymbol{c})$$

where

$$\int f_W(\boldsymbol{c})d\boldsymbol{c} = n_W$$

i.e.

$$\int g_W(\boldsymbol{c})d\boldsymbol{c} = 1$$

For the sake of simplicity, let us assume that the wall lies in the $x = 0$ plane and does not move normal to this plane (e.g. a problem setup as shown in Figure 3-5). The no-flux boundary condition [46], which describes the DSMC simulation is given by

$$-n_W \int_{c_x>0} c_x g_W(\boldsymbol{c})d\boldsymbol{c} = \int_{c_x<0} c_x f(\boldsymbol{c})d\boldsymbol{c} \tag{4.1}$$

and since N_{in} particles arrive at a wall of area Δs in a timestep of Δt, the flux of particles to the wall is given by

$$\frac{N_{Eff} N_{in}}{\Delta s \Delta t}$$

Similarly, for the equilibrium case we have

$$-n_{W,eq} \int_{c_x>0} c_x g_{W,eq}(\boldsymbol{c})d\boldsymbol{c} = \int_{c_x<0} c_x f_{eq}(\boldsymbol{c})d\boldsymbol{c} \tag{4.2}$$

Note that the right hand side of 4.2 can be written as

$$\int_{c_x<0} c_x f_{eq}(\boldsymbol{c})d\boldsymbol{c} = \int_{c_x<0} c_x W(\boldsymbol{c}) f(\boldsymbol{c})d\boldsymbol{c}$$

which can be approximated as

$$\frac{N_{Eff}}{\Delta s \Delta t} \sum_{i=1}^{N_{in}} W_i \tag{4.3}$$

Solving for n_W and $n_{W,eq}$ in Equation 4.1 and Equation 4.2 we get:

$$n_W = \frac{\int_{c_x<0} c_x f(\boldsymbol{c})d\boldsymbol{c}}{\int_{c_x>0} c_x g_W(\boldsymbol{c})d\boldsymbol{c}} = \frac{\frac{N_{Eff} N_{in}}{\Delta s \Delta t}}{\int_{c_x>0} c_x g_W(\boldsymbol{c})d\boldsymbol{c}} = \frac{\frac{N_{Eff} N_{in}}{\Delta s \Delta t}}{G_{NE}}$$

and

$$n_{W,eq} = \frac{\frac{N_{Eff}\sum_{i=1}^{N_{in}} W_i}{\Delta s \Delta t}}{\int_{c_x>0} c_x g_{W,eq}(\boldsymbol{c}) d\boldsymbol{c}} = \frac{N_{Eff}\frac{\sum_{i=1}^{N_{in}} W_i}{\Delta s \Delta t}}{G_{eq}}$$

where we define the functions G to be

$$\begin{aligned} G_{NE} &= \int_{c_x>0} c_x g_W(\boldsymbol{c}) d\boldsymbol{c} \\ G_{eq} &= \int_{c_x>0} c_x g_{W,eq}(\boldsymbol{c}) d\boldsymbol{c} \end{aligned} \tag{4.4}$$

which we will deal with shortly. The post-collision weights of the particles that have just collided with the wall can now be written as

$$W_i' = W(\boldsymbol{c}_i') = \frac{f_{eq}(\boldsymbol{c}_i')}{f(\boldsymbol{c}_i')} = \frac{n_{W,eq} g_{W,eq}(\boldsymbol{c}_i')}{n_W g_W(\boldsymbol{c}_i')} = \frac{G_{NE}}{G_{eq}} \frac{\sum_{j=1}^{N_{in}} W_j}{N_{in}} \frac{g_{W,eq,i}}{g_{W,i}}$$

The terms G_{NE} and G_{eq} can be analytically evaluated using 4.4 to yield

$$\int_{c_x>0} c_x g_{MB}(\boldsymbol{c}) d\boldsymbol{c} = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{kT_{MB}}{m}}$$

and so $G_{NE}/G_{eq} = \sqrt{T_{W,NE}/T_{W,eq}}$ which finally gives:

$$W_i' = \frac{\sum_{j=1}^{N_{in}} W_j}{N_{in}} \frac{g_{W,eq,i}}{g_{W,i}} \sqrt{\frac{T_W}{T_{W,eq}}} \tag{4.5}$$

Note that the first term in Equation 4.5 is the average weight of particles reaching a surface in a timestep while the other two terms are purely determined by the scattering properties of the wall surface.

It is interesting to note that there are alternative approaches to deriving the no flux wall boundary condition that lead to variations of Equation 4.5. We introduced one in Section 3.3.1.2 that can be shown to be equivalent to this formula for a large number of particles but will have problems in the limit of small number of particles colliding with the wall in a given timestep . Equation 4.5 can be applied even when

Δt is so small that there is only one particle colliding with the wall in each step giving:

$$W_i' = W_i \frac{g_{W,eq,i}}{g_{W,i}} \sqrt{\frac{T_W}{T_{W,eq}}} \tag{4.6}$$

Equation 4.6 can, in fact, be applied to each particle even when more particles collide with the wall in every timestep. Our numerical experiments confirm that this equation is still valid though it has a slight performance disadvantage, because it produces simulations which are marginally less stable but with a bias that is practically identical to that of Equation 4.5. A longer discussion of this as well as an alternative way of deriving it based on the conditional probability principles introduced in Chapter 6, can be found in [67].

4.1 Conservation of Equilibrium Mass In Closed Simulations

Figure 4-1 shows the total gas mass in a $Kn = 1.0$ Couette simulation domain for the equilibrium and non-equilibrium parts of the simulation; these simulations use Equation 4.5 to implement boundary conditions. Clearly, the total mass in the equilibrium simulation is randomly walking as we take more and more steps. Although this walk is very slow (and decreases with increasing N_{cell}) it adversely affects the accuracy of equilibrium gas properties at long times. The cause of this random walk is not hard to understand in light of the stochastic nature of the weight update rules (Table 3.1) and the enforcement of the no-flux boundary condition on both sides of the domain. Specifically, the particle-particle as well as the wall-particle collision steps are a source of random updates for the weights. This would not be an issue in any simulation that has a fixed number density in one of its boundary conditions (say, an open wall boundary condition for example) but the no-flux on both wall allows the total equilibrium weights to "float" and randomly walk.

One solution to this problem is to scale the total weights in the domain such that the total weight of equilibrium particles is the same as its initial value. This can be

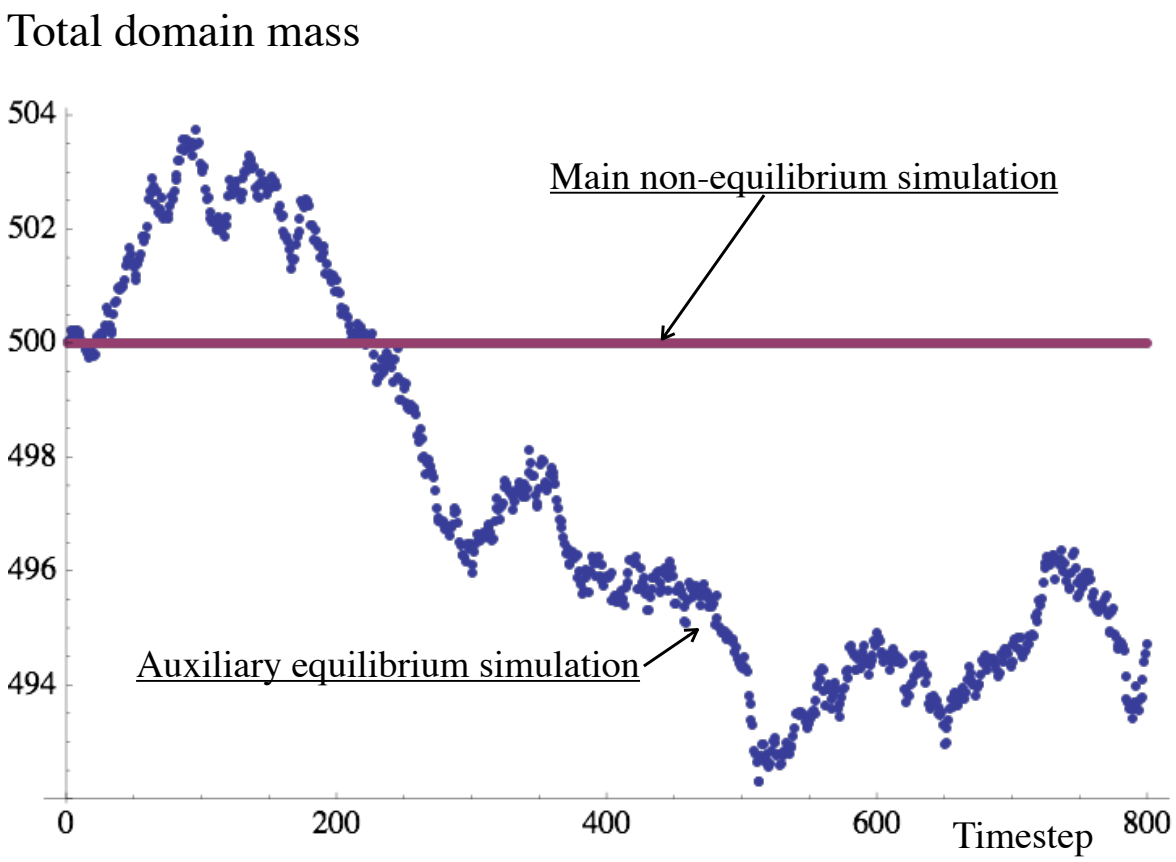

Figure 4-1: Total mass of gas in simple one-dimensional VRDSMC simulation domain as a function of timesteps for the equilibrium (blue) and non-equilibrium (purple) simulations without enforcing conservation of mass. The equilibrium simulation randomly walks due to the stochastic nature of the weight update steps associated with inter-particle collisions. Both simulations start with a total mass of 500.

justified by recalling that [38]:

$$\text{Total equlibrium mass} = \alpha = \int\int f_{eq} d\boldsymbol{x} d\boldsymbol{c} = \int\int W f d\boldsymbol{x} d\boldsymbol{c} \tag{4.7}$$

implying that

$$1 = \frac{\int\int W f d\boldsymbol{x} d\boldsymbol{c}}{\alpha} \simeq \frac{\sum_{i=1}^{N} W_i}{\alpha} \tag{4.8}$$

where N is the number of particles in all cells. The expected value of α is known

exactly using Equation 4.7 and so applying the transformation

$$W_i' = \frac{\alpha}{\left(\sum_{i=1}^{N} W_i\right)} W_i \tag{4.9}$$

to all particles in the domain will result in the exact conservation of mass in the equilibrium simulation. This step requires an extra loop over all particles in the domain and so will not change the scaling of the overall algorithm. Furtheremore, a similar relation to Equation 4.9 can derived and applied on a cell-by-cell basis.

Chapter 5

Using a Variable Reference State

In this Chapter we build on the techniques and results from the last two Chapters to produce an improved version of VRDSMC. As we will see, a small modification to the equilibrium collision routine of Chapter 3 results in a substantial reduction in the number of particles needed for accurate simulations. Specifically, we will show in Section 5.3 that the method is able to produce results that are within engineering accuracy ($\sim 1\%$ relative error) with $N_{cell} < 5,000$ for the majority of problems of practical interest. Indeed, as we will argue in Section 5.4, VRDSMC is currently one of the most appropriate methods for practically solving low-signal kinetic flows.

We start by introducing the concept of a local reference equilibrium state and describe how to utilize it to obtain improved computational performance. Section 5.2 gives a overview of the final version of the VRDSMC method that includes the results of Chapter 4. We conclude the Chapter with a few examples that illustrate the improvement in performance.

5.1 Reducing Bias Using a Local MB Reference Distribution

As outlined before and discussed further in Chapter 7, the VRDSMC procedure introduced in this work gives results that are biased due to the KDE step required to

stabilize the simulation. In particular, the bias increases with increasing ε (the radius of the density estimator kernel) which means that accurate results, especially for small Kn, require ε to be as small as possible. On the other hand, stability demands a finite average number of nearest neighbors $\overline{||S_i||}$, which implies that a relatively large N_{cell} is needed for a stable calculation. For example, we saw in Section 3.4 that the simulation of a $Kn = 0.1$ Couette flow problem required $N_{cell} \simeq 50,000$ to yield a relative error (bias) smaller than 1%.

In this chapter we show that the number of particles per cell for a stable and practically unbiased simulation can be dramatically reduced by modifying the way we perform our collision weight updates. Specifically, by temporarily modifying the reference equilibrium state from the global one, $f_{eq,0}$, (that is identical across the simulation domain) into a local Maxwell-Boltzmann distribution, $f_{MB,loc}$, we find that we need a substantially smaller N_{cell} for a given bias level.

The key challenge in implementing this approach is updating particle weights such that they represent $f_{MB,loc}$ during the collision step only, since the reminder of the formulation (initial and boundary conditions, advection) are most easily dealt with based on a global reference equilibrium.

5.1.1 Variable Reference State

We will use a simple 1D problem to illustrate how we can change the reference state of the equilibrium simulation without explicitly tracking or resampling $f(c)$. To proceed, let there be N samples $\{c_i\}$ drawn from a distribution $f_{sample}(c)$, e.g. a 1D non-equilibrium distribution. There are two reference states that we are interested in representing: the first is $f_{start}(c)$ which could be for example, the global equilibrium reference distribution $f_{eq,0}$, and f_{end} which may be a specific local MB reference distribution. By taking

$$W_i = \frac{f_{start}(c_i)}{f_{sample}(c_i)}$$

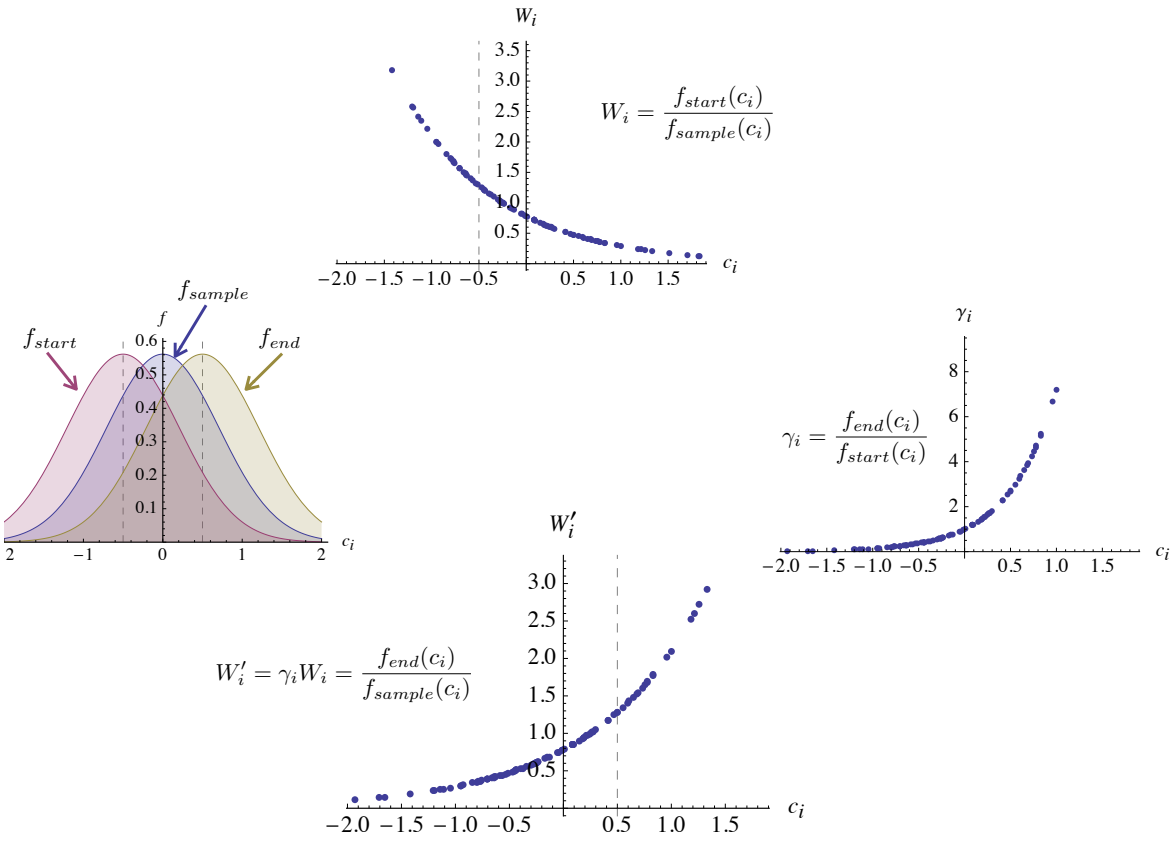

Figure 5-1: Illustration of γ_i for a simple one-dimensional problem with 100 sample points. Left: Plots of the PDF of the sampling distribution f_{sample}, the first reference state f_{start} and the final reference state f_{end}. Top: $\{c_i, W_i\}$. Right:$\{c_i, \gamma_i\}$. Bottom: $\{c_i, W_i'\}$.

we can represent and sample the distribution f_{start} using the set of pairs

$$\{c_i, W_i\} \sim f_{start}$$

What we would like is a method of finding weights W_i' such that we can sample the new distribution f_{end} by using the set of pairs

$$\{c_i, W_i'\} \sim f_{end} \tag{5.1}$$

without changing the samples c_i. Let us assume that W_i' can be found and has the

relation $W_i' = \gamma_i W_i$ for some $\gamma_i = \gamma(c_i)$. In such a case we can write:

$$\gamma_i = \gamma(c_i) = \frac{W'(c_i)}{W(c_i)} = \frac{\frac{f_{end}(c_i)}{f_{sample}(c_i)}}{\frac{f_{start}(c_i)}{f_{sample}(c_i)}} = \frac{f_{end}(c_i)}{f_{start}(c_i)}$$

Clearly this allows us to readily generate the set 5.1 without explicitly evaluating f_{sample} or f_{end} at every sample point c_i. We can use an identical argument to define a factor γ_i in three dimensions that will allow us to generate a representation of a local MB distribution $f_{MB,loc}(\boldsymbol{c})$ from a set $\{c_i, W_i\}$ that samples the global reference distribution $f_{eq,0}(\boldsymbol{c})$. In other words, if $f_{MB,loc}$ is some local Maxwell-Boltzmann distribution, we can use the relation

$$\gamma_i = \frac{f_{MB,loc}(\boldsymbol{c}_i)}{f_{eq,0}(\boldsymbol{c}_i)} \tag{5.2}$$

and

$$W_i' = \gamma_i W_i \tag{5.3}$$

to find the new weight for each particle in the cell.

To further illustrate this point using a numerical example, let us turn our attention to Figure 5-1. In the top panel we plot the analytical value of three distributions f_{sample}, f_{end}, and f_{start}. In the other three panels we plot W_i, γ_i and W_i' vs. c_i, respectively. In this example have chosen the three distributions f_{sample}, f_{end}, and f_{start} to have the means $\langle c \rangle_{sample}, \langle c \rangle_{end}, \langle c \rangle_{start}$ respectively. We see that using the transformation described above we can readily go from sampling f_{start} to f_{end} and back.

5.2 Final Algorithm Summary

The diagram in Figure 5-2 is a flowchart of the improved VRDSMC algorithm that incorporates a change of reference equilibrium (to the local equilibrium) for the collision step. This is possible because the value of the collision integral is identically zero for all Maxwell-Boltzmann distributions [6]. The key steps of this new algorithm are:

1. **Advection Step:** This is identical to a regular DSMC simulation with weights

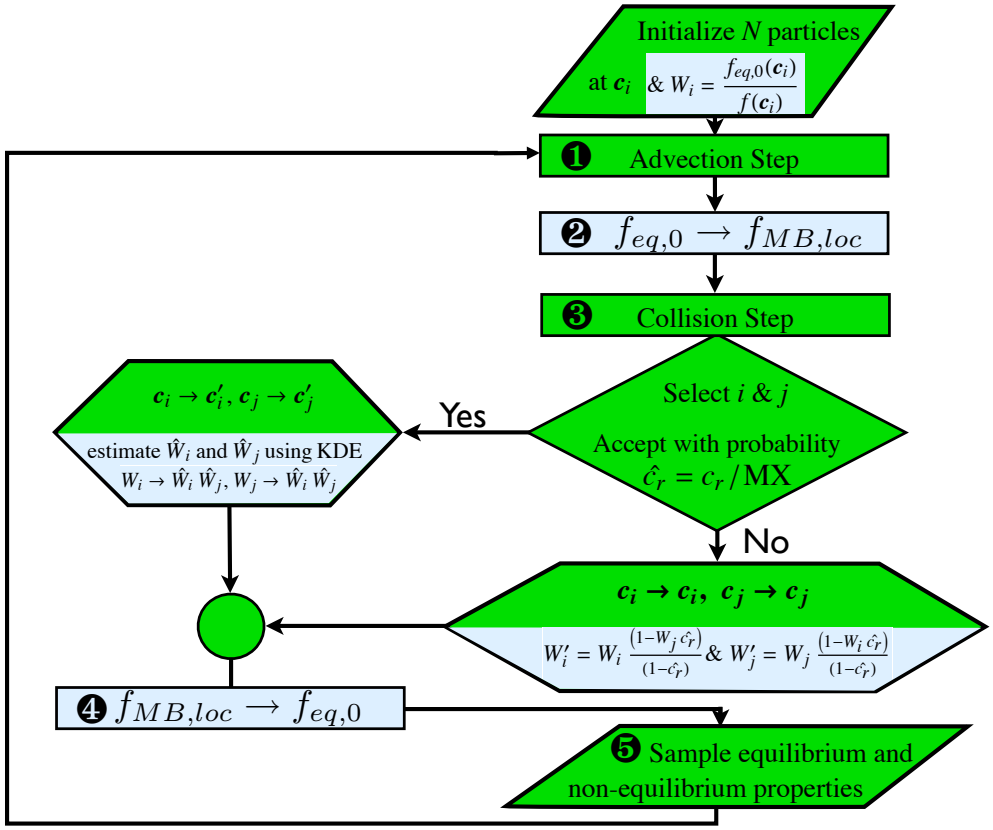

Figure 5-2: Final VRDSMC flowchart. Note that the only difference between this algorithm and the one shown in Figure 3-4 are the steps 2 and 4. These are the steps that change the reference state in the equilibrium simulation to and from a local Maxwell-Boltzmann state.

simply following the particles as detailed in Section 3.3.1.3. As a reminder, all initial conditions and wall interactions are calculated with a reference equilibrium state $f_{eq,0}$ that is constant across all cells.

2. **Change the equilibrium reference state from $f_{eq,0}$ to a local MB $f_{MB,loc}$:** For each cell, update the weights using Equation 5.2 so that the equilibrium simulation samples the local equilibrium distribution $f_{MB,loc}$. The local equilibrium distribution of a cell is specified by $\overline{n}_{VR}, \bar{\boldsymbol{u}}_{VR}$, and $\overline{T}_{VR}$, which are estimated using the variance-reduced estimator. The weight adjustment $\gamma(\boldsymbol{c}_i)$ for each cell is given by

$$\gamma_i = \gamma(\boldsymbol{c}_i) = \frac{f_{MB,loc}(\boldsymbol{c}_i; \overline{n}_{VR}, \bar{\boldsymbol{u}}_{VR}, \overline{T}_{VR})}{f_{eq,0}(\boldsymbol{c}_i)}$$

3. **Collision Step:** As detailed in Section 3.3.1, a set of candidate particles are chosen in a manner that is identical to a regular DSMC simulation and are accepted for scattering with a probability proportional to their relative speed ($\hat{c}_r = c_r/MX$ for some constant MX).

    - **Accepted Particles:** Are scattered using the standard DSMC procedures [12]. The post-collision equilibrium weights are updated ($W_i' = \hat{W}_i\hat{W}_j$) using weights estimated using a KDE with a kernel of diameter εc_0. For more details see Section 3.3.2.

    - **Rejected Particles:** Velocities are not changed but weights are updated without using KDE. For more details see Section 3.3.1.4.

4. **Change equilibrium reference state from local MB $f_{MB,loc}$ to global reference state $f_{eq,0}$:** This is the opposite of Step 2 and is needed to allow particle advection across cells. Since the collision step conserves mass, momentum and energy we can use properties $\bar{n}_{VR}$, $\bar{\boldsymbol{u}}_{VR}$, $\bar{T}_{VR}$ from Step 2 leading to

$$\gamma_i = \gamma(\boldsymbol{c}_i) = \frac{f_{eq,0}(\boldsymbol{c}_i)}{f_{MB,loc}(\boldsymbol{c}_i; \overline{n}_{VR}, \bar{\boldsymbol{u}}_{VR}, \overline{\overline{T}}_{VR})} \tag{5.4}$$

5. **Sample:** This step is very similar to the regular DSMC sampling step that is typically implemented in the sorting routine. For completeness, we include in Table 5.1 a summary of the variance-reduced estimators (as well as their DSMC counterparts) for the most common hydrodynamic properties of interest.

5.3 Results

The major advantage associated with a variable reference equilibrium is the ability to get the same level of bias with a **substantially reduced $\boldsymbol{N_{cell}}$**. This is illustrated and further discussed in this Section.

Table 5.1: Table summarizing regular DSMC estimators vs. VRDSMC estimators. Although slightly more complex, VRDSMC estimators have the same computational complexity

Prop.	VRDSMC	DSMC
$\overline{n}$	$\frac{N_{Eff}}{V}(\sum_{i=1}^{N_{cell}}(1-W_i)+N_{cell,eq})$	$\frac{N_{Eff}N_{cell}}{V}$
$\overline{u_x}$	$\frac{1}{N_{cell}N_{Eff}}(N_{Eff}\sum_{i=1}^{N_{cell}}c_{x,i}(1-W_i)+V\,n_{eq}u_{x,eq})$	$\frac{1}{N_{cell}}\sum_{i=1}^{N_{cell}}c_{x,i}$
$\overline{T}$	$\frac{m}{3k}\left\{\frac{3k\,N_{cell,eq}T_{eq}}{m\,N_{cell}} - (\bar{u}_{x,VR}^2+\bar{u}_{y,VR}^2+\bar{u}_{z,VR}^2) + \frac{1}{N_{cell}}\sum_{i=1}^{N_{cell}}(1-W_i)(c_{x,i}^2+c_{y,i}^2+c_{z,i}^2)\right\}$	$\frac{m}{3k}\left\{\frac{1}{N_{cell}}\times\left\{\sum_{i=1}^{N_{cell}}(c_{x,i}^2+c_{y,i}^2+c_{z,i}^2)\right\} -(\bar{u}_x^2+\bar{u}_y^2+\bar{u}_z^2)\right\}$
$\overline{q}$	See Appendix B	

5.3.1 Examples

The physical setup of the following simulations are identical to that presented in Chapter 3.

Steady State Simulation of Couette Flows with $Kn = 0.1$ & $Kn = 1.0$

Figure 5-3 shows the results for steady Couette Flows at $Kn = 0.1$ and $Kn = 1$; these problems were also considered in Section 3.4.1. The Figure shows that we can obtain the same accuracy ($\sim 1\%$) for substantially smaller N_{cell}. Specifically, for $\overline{||S_i||} = 10$, we used $N_{cell} = 100$ particles for the $Kn = 1$ simulation and $N_{cell} = 2500$ for the $Kn = 0.1$ simulation which is a substantially smaller than the older results (500 and $50,000$ particles respectively). In other words, for a Knudsen of 0.1 a factor of 20 improvement is achieved.

Transient Simulation of an impulsive wall-temperature change problem at $Kn = 0.1$

A more challenging problem involves the transient response of a gas to an impulsive change to its boundary temperature. In this one-dimensional simulation, we solve for

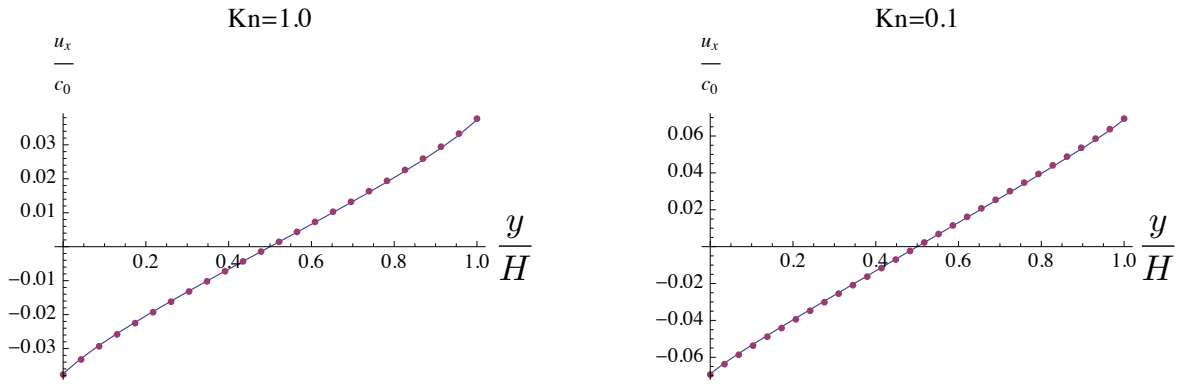

Figure 5-3: Steady state Couette flow solution. In both cases we have $\overline{||S_i||} = 10$ and less than 1% relative error can be achieved with $N_{cell} = 100$ and $2,500$ for $Kn = 1.0$ and 1.0 respectively.

the evolution of a gas that is initially at $300K$ when the walls are instantaneously changed to $310K$ and $290K$ at $t = 0^+$. Figure 5-4 shows VRDSMC results in excellent agreement with DSMC counterpart for all hydrodynamics properties of interest (namely ρ, q_y, T and u_y). In this calculation $Kn = 0.1$ and $N_{cell} = 7,500$.

5.3.2 Stability Plot Using Adjusted Reference State

Use of a variable reference state does not change the stability limits of VRDSMC as originally presented in Figure 3-3. This is verified in Figure 5-5 which shows the stability map for the algorithm of Section 5.2. The trend is very similar to the one shown in Figure 3-3 with the notable exception that the calculation is unstable even

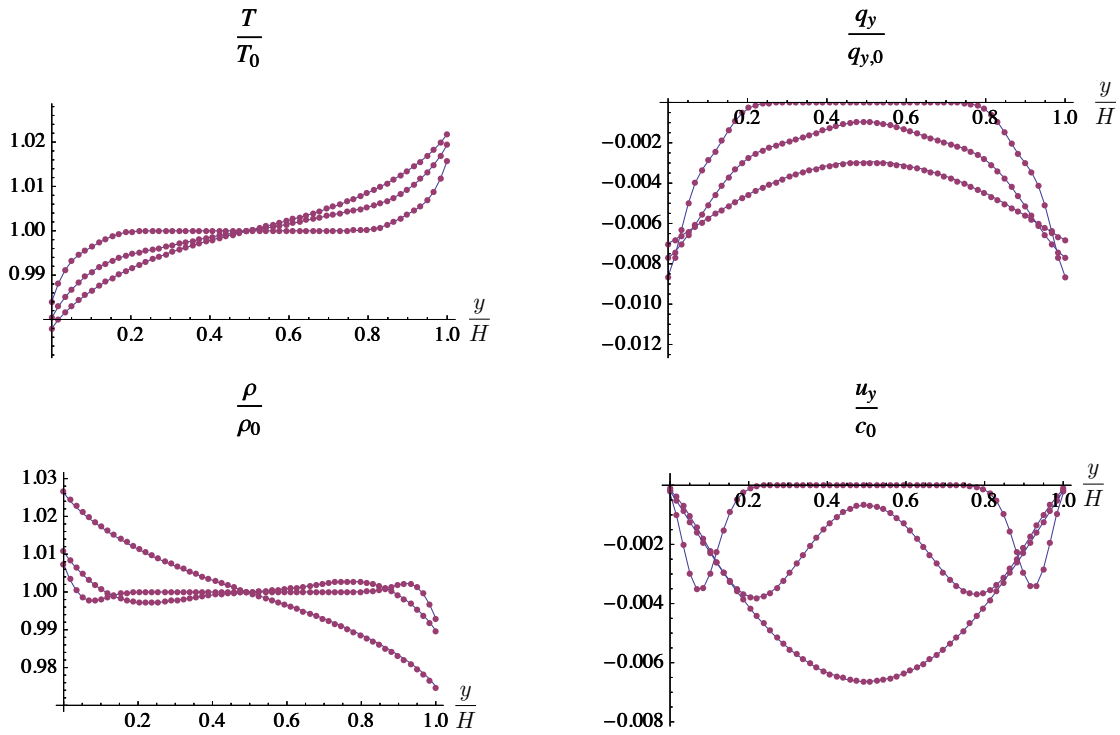

Figure 5-4: Transient temperature step problem with walls going from 300 to $300 \pm 10K$ at time $t = 0$. Snapshots of 1, 3 and 12 mean collision times as shown, for $N_{cell} = 7,500$, $\overline{\|S_i\|} = 10$, corresponding to $\varepsilon = 17\%$.

for $\overline{\|S_i\|} = 0$; this however, is a result of the boundary condition used; specifically, for the present map, boundary condition 4.5 was used, whereas for the stability map of Figure 3-3, n_W was assumed known.

Figure 5-6 shows a transient plot of $\sigma^2\{W_i\}$ for a typical 1D calculation (problem of Section 5.3.1, boundary condition 4.5). We would like to highlight two interesting features in this figure: the first is that the variance does not grow monotonically and indeed its steady state value is substantially smaller than the peak value which happens early on. The second feature is the slightly higher value of the variance in the middle of the domain which is caused by the boundary conditions reducing the variance of particles that are re-emitted into the domain. In fact, if we apply the

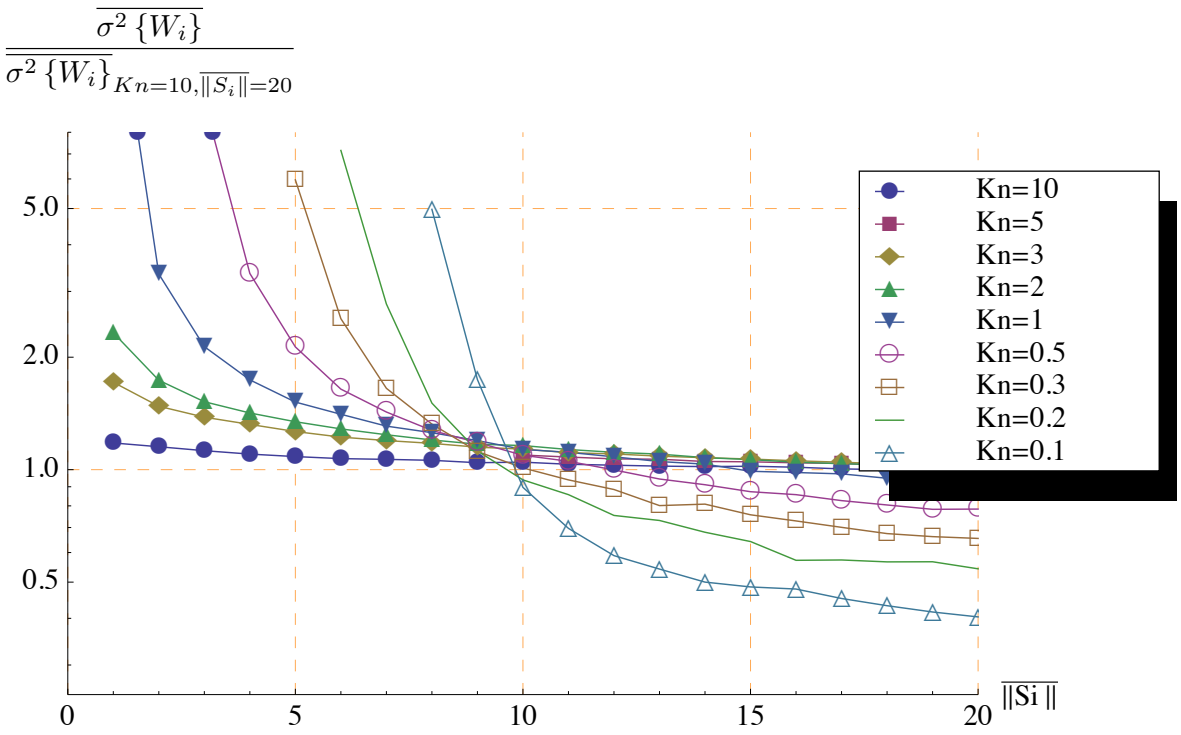

Figure 5-5: Re-creation of the stability map for a 1D Couette flow of Section 3.3.2. The trends are almost identical and show that as the average number of nearest neighbors ($\overline{||S_i||}$) increases the simulation variance decreases. Weight variance is normalized by the variance of a $Kn = 10$ flow with $\overline{\|S_i\|} = 20$.

weight update rule 4.6 instead of Equation 4.5, we will get a profile that has overall higher average variance, but with a *minimum* in the middle of the domain because implementation 4.6 contributes to $\sigma^2\{W_i\}$ rather than reducing it.

5.4 VRDSMC Performance Advantage

As one can expect from our description, VRDSMC is fairly simple to implement and has a surprising small code footprint. For example, in our implementation, our total codebase was about 1800 lines of code, only a little more than 400 of which are related to variance reduction.

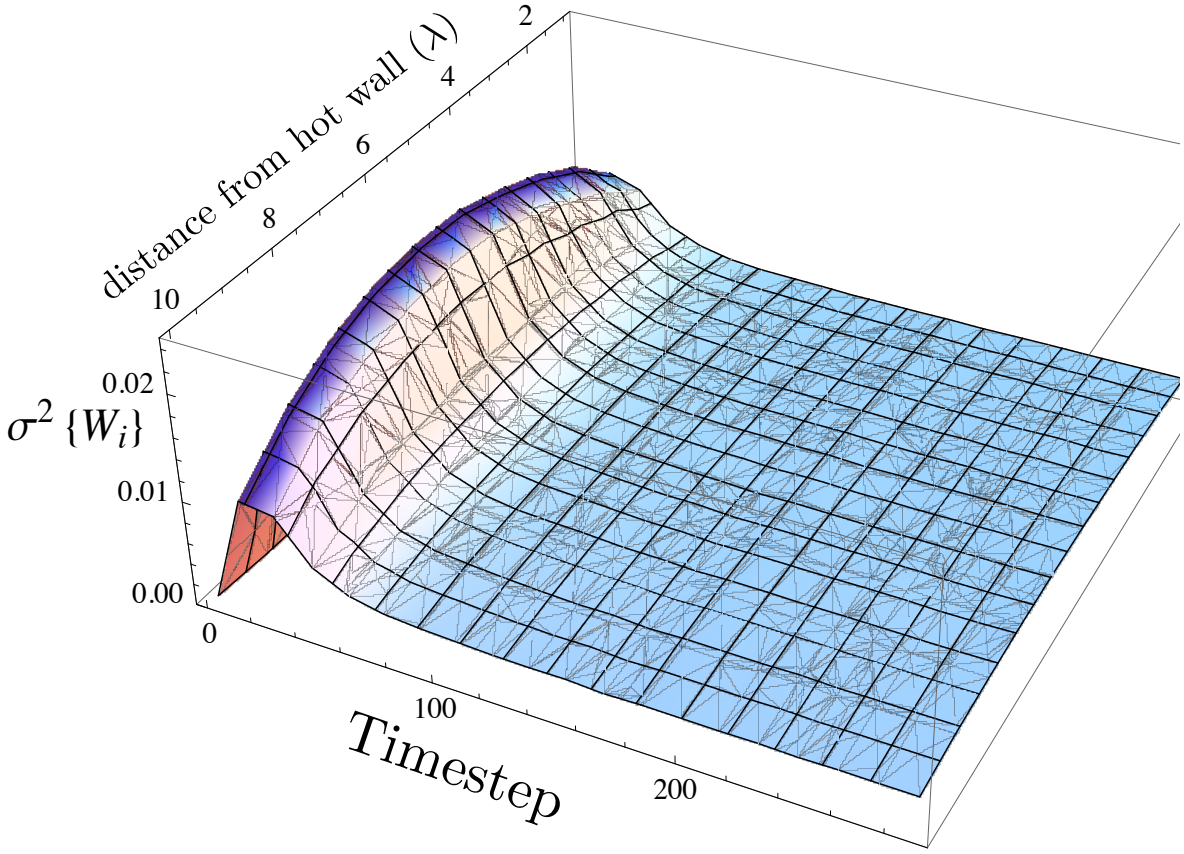

Figure 5-6: Weight variance ($\sigma^2\{W_i\}$) vs. time and space for a transient problem similar to that Section 5.3.1 (with 10 cells and $N_{cell} = 7,500$). Note the lower steady state values of variance compared to early-time values, as well as a variance maximum that occurs in the middle of the simulation domain.

5.4.1 VRDSMC Execution Speed & Advantage Over DSMC

As we saw in Section 3.4.2, for a given uncertainty, a much smaller number of samples are needed in VRDSMC compared to DSMC, an advantage that quickly increases as the signal gets smaller. Although DSMC requires many more samples than VRDSMC for a given uncertainty, this large advantage comes at a certain speed cost that as we will see, is quite tolerable in the vast majority of cases. VRDSMC is more expensive because:

- Execution time scales like $O(N_{cell}Log(N_{cell}))$ at worst, compared to O(N_{cell}) for a regular DSMC simulation. This is caused by the need to find nearest

neighbors in velocity space; this is further discussed in Appendix A.

- Once nearest neighbors are found, weights are averaged to get $\hat{W}_i$. This operation scales with $\overline{\|S_i\|}$ and is probably the largest component of the extra cost.

- Enforcement of the no-flux boundary condition requires us to keep track of particles that interacted with the wall. This is a moderate cost and does not change how the cost scales with N_{cell}.

- VRDSMC has a larger storage requirement since it requires keeping track of $\{\mathrm{W}_i\}$ and a few other variables associated with the equilibrium simulation. Overall, these are very marginal increases and would only cause a slowdown in situations where the extra RAM requirements are such that DSMC fits within the CPU's cache while VRDSMC needs to use to the system's main memory.

- The stabilization of VRDSMC without introducing appreciable bias requires a larger N_{cell} in many simulations. This is in contrast to DSMC which typically will require $N_{cell} = O(100)$. This issue is only likely to be a limiting factor at low Kn in flows that are not very close to equilibrium. This also may be a limitation in multi-dimensional flows in cases where a fine mesh is required.

As a point of reference, we ran a few calculations comparing VRDSMC and DSMC for simulations with identical parameters and listed the results in Table 5.2. Overall, the execution penalty generally lies in the $2 - 7\times$ range, which is quite moderate considering the large improvement in sampling fidelity achieved by VRDSMC. Looking more carefully at the table, we see that VRDSMC simulations become slower as $\overline{||S_i||}$ is increased when holding everything else constant. The slowdown is slightly mitigated by the fact that more stable calculations with larger $\overline{||S_i||}$ tend to have a lower MX which translates to faster collision routines since we need to process a significantly smaller number of collision candidates. Finally, it is interesting to note that the VRDSMC performance penalty is sometimes dramatically reduced in cases

Table 5.2: Time needed to run 1000 steps of a transient simulation for a Couette flow problem. Execution times are those on a 2.93GHz Core-2 Macbook Pro.

Kn	Cells	N_{cell}	$\overline{\|S_i\|}$	VRDSMC Time(s)	DSMC Time(s)	Slowdown
1.0	6	500	5	8	4	$\backsimeq 2$
1.0	6	2500	5	36	8	4.25
1.0	50	500	5	56	19	$\backsimeq 3$
1.0	50	2500	5	236	43	5.5
1.0	50	500	10	52	19	2.7
1.0	50	2500	10	238	43	5.5
0.1	50	2500	10	338	54	6.25

where the whole simulation fits in a CPU's cache when we have a small number of particles.

5.4.1.1 When does DSMC have an advantage over VRDSMC?

As one would expect, DSMC performs better than VRDSMC when we have a problem that is not close to equilibrium. The trade-off is that VRDSMC is somewhat slower as we saw above and its advantage over DSMC decreases as the problem moves away from equilibrium. For example, in our implementation, steady Couette flows with Kn$<$0.5 can probably be simulated using DSMC faster only when the velocity we are interested in sampling is bigger than about $Ma > 0.05$. The DSMC advantage however will immediately go away when we start looking at other "induced" moments (e.g. ρ, T, etc.) since they are typically of much smaller magnitude than the flow directly induced by the wall movement.

We also recall that for a sufficiently large Kn and with a specified wall number density, no KDE is required. In such cases, VRDSMC scales like $O(N_{cell})$ and performs almost as fast as DSMC making it even more attractive.

To summarize, we recommend considering VRDSMC for any low-signal kinetic flow problem with $Kn > 0.1$, provided we are close enough to equilibrium ($Ma \lesssim 0.1$).

5.4.2 Advantage Over LVDSMC

The comparison between VRDSMC and the deviational LVDSMC method of Homolle and Hadjiconstantinou [53, 51] is more complicated. This is partially because LVDSMC has only been recently developed and some aspects of its performance and limitations are not well characterized. Despite this, we have many reasons to believe that a practitioner should probably consider VRDSMC before looking at LVDSMC for the following reasons:

- As discussed elsewhere, LVDSMC is significantly more complex to implement.
- VRDSMC is simpler to derive and can be fairly easily generalized to other collision models.
- Our method is probably simpler to implement in the cases of multiple dimensions especially for complex geometries. One major reason for this is the need to create particles at cell boundaries of LVDSMC simulations as part of integrating the advection operator using a different reference state in each cell. LVDSMC can perform using only one global reference equilibrium (in which case no particle generation at cell boundaries is required) at the cost of reduced variance reduction [66].

It is a more complex issue to compare LVDSMC with VRDSMC in terms of performance for a number of reasons related to the fundamental differences between the two formulations. For example, although the evaluation of the convolution term in Equations 1.9 and 1.10 would point to the possibility of $O(N_{cell}^2)$ scaling, all current experience points to a fixed evaluation cost that does not scale with the number of particles in the cell. Another example of the complexity of comparing the two methods is that particles in LVDSMC represent deviations from equilibrium, so it is common for certain simulations to have a very large variation in the number of particles per cell across the domain which is typically not the case in DSMC or VRDSMC.

Chapter 6

Simulation Weight Update Rules Using Conditional Probability Arguments

In this Chapter we explore an alternative approach to deriving the weight update rules presented in Section 3.3.1 based on conditional probability arguments. This alternative approach is more general and allows the derivation of rules for other collision models quickly and intuitively. In the first section below we will introduce the general conditional probability approach and then proceed in Section 6.2 to apply it to the hard-sphere collision model.

6.1 Weight Evolution Rules Using Conditional Probabilities: Guiding Principles

The weight update rules given in previous Chapters can be derived in an intuitive manner using conditional probability arguments. To simplify the analysis, let us start by only looking at spatially homogeneous problems; this constraint will be relaxed later. To proceed, let us consider two simulations that use N_{cell} simulation particles, with initial velocities $\{\boldsymbol{c}_i\}$:

1. Simulation A: each simulation particle represents N_{Eff} physical particles. As the simulation progresses, the velocity of every particle i is updated from $\boldsymbol{c}_i$ at time $t = 0$ to $\boldsymbol{c}'_i$ at time $t' = \Delta t$ with probability $P_{A:\boldsymbol{c}_i \to \boldsymbol{c}'_i}$. In general, $\boldsymbol{c}'_i$ may or may not be identical to $\boldsymbol{c}_i$ and $0 \leq P_{A:\boldsymbol{c}_i \to \boldsymbol{c}'_i} \leq 1$ accounts for a transition probability that allows the simulation of both stochastic and deterministic processes.

2. Simulation B: a different simulation that is performed in parallel to simulation A using the same N_{cell} particles that are also initially at $\{\boldsymbol{c}_i\}$. However, in contrast to Simulation A where each particle represents N_{Eff} physical particles, here, each particle i represents $W_i N_{Eff}$ physical particles. $P_{B:\boldsymbol{c}_i \to \boldsymbol{c}'_i}$ is defined in a manner identical to $P_{A:\boldsymbol{c}_i \to \boldsymbol{c}'_i}$, but captures the transition probabilities for Simulation B.

One way to proceed is to independently integrate simulations A and B by updating all the particle velocities $\boldsymbol{c}_i \to \boldsymbol{c}'_i$ using the appropriate conditional probability in each simulation. In the case of simulation A we will move each particle i from its initial velocity $\boldsymbol{c}_i$ to $\boldsymbol{c}'_i$ with probability $P_{A:\boldsymbol{c}_i \to \boldsymbol{c}'_i}$ resulting in N_{cell} particles at $\{\boldsymbol{c}'_i\}_A$ each representing N_{Eff} real particles. Simulation B on the other hand, will result in particles at $\{\boldsymbol{c}'_i\}_B$ at time t' each representing $W_i N_{Eff}$ particles; since $P_{A:\boldsymbol{c}_i \to \boldsymbol{c}'_i} \neq P_{B:\boldsymbol{c}_i \to \boldsymbol{c}'_i}$ we will have, $\{c'_i\}_A \neq \{c'_i\}_B$ in general.

A workable variance reduction procedure needs both simulations to stay correlated at all times. This can be achieved by integrating both simulations in a synchronous manner, by updating the velocities of both simulations according to the transition probabilities of simulation A but **modifying** the weights W'_i at t' to ensure that they are still representing Simulation B.

Given this formulation of the problem, we proceed by explicitly calculating the number of particles in the final state in both simulations and then modifying the weights to correctly account for the proper ratio of particles in the two simulations. To wit, let us assume that for every particle of velocity class $\boldsymbol{c}_i$ there are $N_{Eff} N_{\boldsymbol{c}_i}$ particles in simulation A at initial time t. At time t' the expected number of **actual**

particles at $\boldsymbol{c}_i'$ in this simulation that were at $\boldsymbol{c}_i$ will be

$$N_{Eff} P_{A:\boldsymbol{c}_i \to \boldsymbol{c}_i'} N_{\boldsymbol{c}_i} \tag{6.1}$$

Likewise, in simulation B we will have $W_i N_{Eff} N_{\boldsymbol{c}_i}$ particles of class $\boldsymbol{c}_i$ (recall that $\{\boldsymbol{c}_i\}_A = \{\boldsymbol{c}_i\}_B$ at $t = 0$) and we should have

$$W_i N_{Eff} P_{B:\boldsymbol{c}_i \to \boldsymbol{c}_i'} N_{\boldsymbol{c}_i} \tag{6.2}$$

real particles landing at $\boldsymbol{c}_i'$ that came from $\boldsymbol{c}_i$ to be faithful to the dynamics of simulation B. Unfortunately, since we are going to update the velocities $\boldsymbol{c}_i$ based on $P_{A:c_i \to c_i'}$ we will instead have

$$W_i' N_{Eff}\, P_{A:\boldsymbol{c}_i \to \boldsymbol{c}_i'} N_{\boldsymbol{c}_i} \tag{6.3}$$

real particles in simulation B, where we have allowed W_i' yet to be undetermined. W_i' can be determined by combining Equation 6.2 and Equation 6.3 to yield

$$W_i{}' = W_i \frac{P_{B:\boldsymbol{c}_i \to \boldsymbol{c}_i{}'}}{P_{A:\boldsymbol{c}_i \to \boldsymbol{c}_i{}'}} \tag{6.4}$$

In words, if we update the velocities according to Simulation A we will need to adjust particle weights in order to properly describe simulation B. As we will show, Equation 6.4 is the basis for all the results that can be used to derive weight update rules to a variety of physical phenomena, such as advection, particle collisions, wall interactions, etc., as long as the phenomenon can be cast into this form. Intuitively, it tells us that the post-transition weight of a particle is equal to the original weight of the particle multiplied by the ratio of the required transition probability to the actual transition probability from $\boldsymbol{c}_i$ to $\boldsymbol{c}_i'$.

In the context of the present Thesis, we need to perform an equilibrium and non-equilibrium simulation. This is done by simply identifying the non-equilibrium simulation with Simulation A and the equilibrium simulation with Simulation B which

will result in the following weight update rule:

$$W_i' = W_i \frac{P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'}}{P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'}} \tag{6.5}$$

Clearly, $P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'}$ and $P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'}$ are the transition probabilities from $\boldsymbol{c}_i$ to $\boldsymbol{c}_i'$ for the equilibrium and non-equilibrium simulations, respectively. Note that the choice of which simulation is calculated directly (using $\boldsymbol{c}_i$) and which one is calculated implicitly (using $\boldsymbol{c}_i$ and weights W_i) is arbitrary. For practical reasons we made the above choice since we would like the primary calculation to remain the standard DSMC method with no modifications. To actually use Equation 6.5 to find weight update rules we will need to substitute transition probabilities appropriately, a topic we will explore in the next section.

6.2 Hard-Sphere VRDSMC Weight Update Rules Using Conditional Probabilities

To apply the principles of the last section to a DSMC simulation of equilibrium we need to examine all the different simulation sub-steps individually. We will see that it is fairly straitforward to apply Equation 6.5 to the advection sub-step and the various boundary conditions; more effort, however, will be required to correctly simulate the collision step, which we consider first.

6.2.1 Collision Transition Probabilities for Hard Sphere

The collision step in DSMC [12, 22] is based on an acceptance-rejection procedure that selects a certain number of *candidate* particles and then either *accepts* them for scattering or *rejects* them without modifying their velocities (see Section 1.6). The scattering procedure for selecting the exact post-acceptance velocities is based on simple Newtonian mechanics that conserves mass, momentum and energy and is not related (in the hard-sphere case) to the collision probability. In what follows we will use the following definitions:

$P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'}$ ≡Probability a candidate particle is accepted in the non-equilibrium simulation

$P_{\boldsymbol{c}_i \to \boldsymbol{c}_i' = \boldsymbol{c}_i} = 1 - P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'}$ ≡Probability a candidate particle is rejected in the non-equilibrium simulation

$P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'}$ ≡ Probability a candidate particle is accepted in the equilibrium simulation

$P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i' = \boldsymbol{c}_i} = 1 - P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'}$ ≡Probability a candidate particle is rejected in the equilibrium simulation

Let us start by looking at the accepted transition probabilities for both simulations and proceed in a manner that is identical to the approach for deriving DSMC directly from kinetic theory, namely by estimating the number of collisions in a cell and then creating an acceptance-rejection scheme that will faithfully simulate those collisions.

The number of particles of class $\boldsymbol{c}_j$ "collected" in time Δt by a *single* particle i of velocity $\boldsymbol{c}_i$ is

$$N_{Eff}\, c_r \sigma \frac{\Delta t}{2\,V} \tag{6.6}$$

By extension, if there are $N_{Eff} W_i$ particles of class $\boldsymbol{c}_i$ in a cell and $N_{Eff} W_j$ particles of class $\boldsymbol{c}_j$ in the same cell, the average number of collisions between class $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$ will be

$$N_{Eff}^2 W_i W_j c_r \sigma \frac{\Delta t}{2\,V}$$

and so *each simulation particle* of class $\boldsymbol{c}_i$ we will have

$$N_{Eff} W_j c_r \sigma \frac{\Delta t}{2\,V}$$

collisions. This is because the scattering of each simulation particle represents $N_{Eff} W_j$ actual collisions. Like DSMC, to account for the collision rates of all velocity classes, we simply use an acceptance-rejection procedure to sample all the velocity classes by randomly choosing collision partners in a cell. Keeping the same number of collision

candidates as DSMC (again, referring to Section 1.6) we will need:

$$N_{Eff} N_{cell} (N_{cell} - 1) MX \sigma \frac{\Delta t}{2V} \tag{6.7}$$

candidates. We can account for the weights by modifying the definitions of acceptance probabilities and the value of the probability normalization constant MX. The non-equilibrium simulation will be performed correctly if we choose collision probability

$$P_{\mathbf{c}_i \to \mathbf{c}_i'} = c_r / MX \tag{6.8}$$

while the correct simulation of equilibrium will result from choosing an equilibrium collision probability of

$$P_{eq:\mathbf{c}_i \to \mathbf{c}_i'} = \frac{W_j c_r}{MX} \tag{6.9}$$

To ensure that the collision probabilities stay physically meaningful we need to slightly modify MX from being the maximum cell-based relative velocity to

$$MX = \max_{i,j \in cell} W_i c_r$$

It is noteworthy that since every scattering of computational particle i is meant to be in place of $W_i N_{Eff}$ physical molecules, the collision probability is asymmetric and in general when i and j are collision partners $P_{eq:\mathbf{c}_i \to \mathbf{c}_i'} \neq P_{eq:\mathbf{c}_j \to \mathbf{c}_j'}$. Also, as expected, the above formulation reduces to the standard DSMC if we take $W_i = 1$ for all i.

One key advantage of this formulation compared to the one given in Chapter 3 is that it can be extended to many other collision models since the above derivation does not explicitly use any properties of the form of Boltzmann collision integral. As such, it is easy to extend the weighted approach and the variance reduction approach of this Thesis to other collision models like the BGK model, or other complex collision or possibly reactive models.

6.2.2 Final Collision Update Rules

Using the results from the last section and combining Equation 6.8, Equation 6.9 with Equation 6.5 we can explicitly find the update rules for hard-sphere VRDSMC. For the accepted particles

$$W_i{}' = W_i \frac{P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'}}{P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'}} = W_i W_j \tag{6.10}$$

while for the rejected particles

$$W_i{}' = W_i \frac{P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i' = \boldsymbol{c}_i}}{P_{\boldsymbol{c}_i \to \boldsymbol{c}_i' = \boldsymbol{c}_i}} = W_i \frac{1 - P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'}}{1 - P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'}} = W_i \frac{1 - W_j c_r / MX}{1 - c_r / MX} \tag{6.11}$$

Note that as a result of these relations, a collision candidate pair i and j with $\mathrm{W}_i \neq \mathrm{W}_j$ will have identical post-collision weights when accepted, but will have different weights when rejected.

6.2.3 Advection Substep

Conditional probability arguments can be used to derive weight evolution rules for the advection step as follows. The equilibrium and non-equilibrium transition probabilities in the advection case are simply given by

$$P_{\{\boldsymbol{x}_i, \boldsymbol{c}_i\} \to \{\boldsymbol{x}_i + \Delta t \boldsymbol{c}_i, \boldsymbol{c}_i\}} = 1$$
$$P_{eq:\{\boldsymbol{x}_i, \boldsymbol{c}_i\} \to \{\boldsymbol{x}_i + \Delta t \boldsymbol{c}_i, \boldsymbol{c}_i\}} = 1$$

where $\{\boldsymbol{x}_i, \boldsymbol{c}_i\}$ represents the state of a particle i at $\boldsymbol{x}_i$ that in time Δt moves to a new position without changing its velocity $\boldsymbol{c}_i$. Using Equation 6.5 above, results in

$$W_i' = W_i \tag{6.12}$$

as could be intuitively, perhaps, expected. Note that Equations 6.10, 6.11 and 6.12 are identical to the transition rules that were discussed in Chapter 3 as would be

required for the method to be consistent with the Boltzmann equation.

Wall Collisions We now discuss weight evolution rules for particles colliding with walls during the advection substep. In the cases where the wall post-collision velocity is not a function of the incoming velocity (i.e. a fully accommodating wall) the transition probabilities become:

$$P_{\boldsymbol{c}_i \to \boldsymbol{c}_i'} = f_W(\boldsymbol{c}_i')$$

and

$$P_{eq:\boldsymbol{c}_i \to \boldsymbol{c}_i'} = f_{eq:W}(\boldsymbol{c}_i') = f_{eq,0}(\boldsymbol{c}_i')$$

where f_W is the boundary condition associated with the wall for the non-equilibrium simulation and $f_{eq:W}$ is its equilibrium counterpart that we take to be the reference distribution function. Applying Equation 6.5 we see that our weight update rule for wall collisions is

$$W_i' = W_i \frac{f_{eq,0}(\boldsymbol{c}_i')}{f_W(\boldsymbol{c}_i')} \tag{6.13}$$

Though this relation is the most general and can be applied to almost any boundary condition, it is not practical in many cases. The stability of the calculation is substantially enhanced if the simulation domain has a point where weights are directly evaluated from an explicitly known equilibrium and non-equilibrium distribution and not from an implicit conditional estimate. Because of this we recommend using the wall boundary conditions outlined in Chapter 5 and not the relation derived here.

Body Force Boundary Condition For the case of kinetic flows with applied fields [12] the velocity (in addition to position) is updated in time. Specifically

$$\boldsymbol{c}_i \to \boldsymbol{c}_i' = \boldsymbol{c}_i + \Delta t \boldsymbol{G}$$

Where $\boldsymbol{G}$ is the body force acting on the particles. Directly applying the approach of 3.3.1 will not work since:

$$P_{\{\boldsymbol{x}_i,\boldsymbol{c}_i\}\to\{\boldsymbol{x}_i,\boldsymbol{c}_i+\Delta t\boldsymbol{G}\}} = 1$$
$$P_{eq:\{\boldsymbol{x}_i,\boldsymbol{c}_i\}\to\{\boldsymbol{x}_i,\boldsymbol{c}_i+\Delta t\boldsymbol{G}\}} = 0$$

making $W_i' = 0$ which is not helpful for what we are trying to do. An alternative approach uses the change of reference state explained in Section 5.1.1 and is needed in this case to properly find new weights. In detail the steps are:

1. Starting from a set of particles $\{\boldsymbol{c}_i, W_i\}$ representing $f_{eq,0}(\boldsymbol{c}) = f_{MB}(\boldsymbol{c}; n_0, \mathbf{0}, T_0)$.

2. Update velocities while keeping Weights constant. Because of this the set $\{\boldsymbol{c}_i, W_i\}$ will represent a Maxwell-Boltzmann distribution $f'_{eq,0}(\boldsymbol{c}) = f_{MB}(\boldsymbol{c}; n_0, \Delta t\mathbf{G}, T_0)$.

3. Update W_i to account for the shift. We do this by applying the transformation

$$W_i \to \gamma_i W_i$$

$$\text{where } \gamma_i = \frac{f_{eq,0}(\boldsymbol{c})}{f'_{eq,0}(\boldsymbol{c})}$$

where n_0 and T_0 are the global reference number density and temperature. A similar approach can be used to apply volumetric heating within a cell.

6.2.4 Limitations and Pointers to Stability Issues

The procedure outlined in this section is able to produce completely unbiased variance-reduced simulations of a HS gas that are only stable at high Knudsen numbers and fixed density. In fact, if Equation 6.13 is used to enforce the wall boundary condition, the simulation will be numerically unstable for all values of Kn. This should come as no surprise to the reader since these weight transition rules are identical to the ones derived directly from the Boltzmann equation and so will only be completely stable when KDE is used to limit the growth the variance of the weights.

The reason for the instability is that the transition probabilities that were used in the derivation of the weight update rules are those that give the probability of finding a particle at $\boldsymbol{c}_i'$ when time=t' given that it had the velocity $\boldsymbol{c}_i$ at time t for the equilibrium and non-equilibrium simulations. There is no reason to expect that the equilibrium and non-equilibrium transition probabilities from $\boldsymbol{c} \rightarrow \boldsymbol{c}_i'$ should be similar to each other after many steps and consequently, there is no guarantee that $\|1-W_i\| \ll 1$ as $t \rightarrow \infty$. Indeed, stability can be achieved when weights are estimated based on the absolute probability of finding a particle with velocity $\boldsymbol{c}_i'$ regardless of its previous velocity. This can be achieved by using KDE since it allows us to construct an estimate of the distributions f and f_{eq} at $\boldsymbol{c}_i'$.

6.3 Connection to the LVDSMC Scheme

In this section we discuss a brief investigation undertaken with the objective of finding a connection between the weight formulation described in this Thesis and the LVDSMC method of Homolle and Hadjiconstantinou [53]. In the course of that investigation we were able to develop a method that utilizes the convolution form of the BE (Equations 1.9 and 1.10) within a weight formulation. Let us define M as the inverse of W:

$$M = \frac{1}{W} = \frac{f}{f_{eq}}$$

In other words we consider a simulation that samples the known equilibrium distribution and relates it to the unknown non-equilibrium distribution via a likelihood ratio[1]. Let us define the deviation from absolute equilibrium as

$$f_d = f - f_{eq,0} = (M-1) f_{eq,0}$$

Using this relation and the kernel $K = K_1 + K_2$, we can write the collision integral in the form:

[1]There are some parallels between this approach and [65] though they only use the linearized version of the BE in the standard form (1.1).

$$\left.\frac{dM(\boldsymbol{c})}{dt}\right|_{Collision} = -\nu(\boldsymbol{c})\left(M(\boldsymbol{c})-1\right) + \frac{1}{f_{eq,0}(\boldsymbol{c})}\int K(\boldsymbol{c},\boldsymbol{c}_1)\left(M(\boldsymbol{c}_1)-1\right) f_{eq,0}(\boldsymbol{c}_1)d\boldsymbol{c}_1$$

Approximating $dM(\boldsymbol{c})/dt$ by $\frac{M^*(\boldsymbol{c})-M(\boldsymbol{c})}{\Delta t}$, we obtain the one timestep update rule

$$M^*(\boldsymbol{c}) = M(\boldsymbol{c}) - \nu(\boldsymbol{c})\left(M(\boldsymbol{c})-1\right)\Delta t + \frac{\Delta t}{f_{eq,0}(\boldsymbol{c})}\int K(\boldsymbol{c},\boldsymbol{c}_1)\left(M(\boldsymbol{c}_1)-1\right) f_{eq,0}(\boldsymbol{c}_1)d\boldsymbol{c}_1$$

This can be broken into a two-step sequence

$$M'(\boldsymbol{c}) = M(\boldsymbol{c}) - \nu(\boldsymbol{c})\left(M(\boldsymbol{c})-1\right)\Delta t \tag{6.14}$$

$$M^*(\boldsymbol{c}) = M'(\boldsymbol{c}) + \frac{\Delta t}{f_{eq,0}(\boldsymbol{c})}\int K(\boldsymbol{c},\boldsymbol{c}_1)(M(\boldsymbol{c}_1)-1) f_{eq,0}(\boldsymbol{c}_1)d\boldsymbol{c}_1 \tag{6.15}$$

with the following interpretation: In the first step we stochastically take $M(\boldsymbol{c})$ to 1 with probability $\Delta t\nu(\boldsymbol{c})$. In the second step we update $M'(\boldsymbol{c})$ to $M^*(\boldsymbol{c})$ by evaluating the convolution as a sum over the simulation particles using importance sampling.

Our numerical implementation of this scheme indicates that given enough simulation particles, the method is able to produce an accurate and stable simulation for short simulation times; however, due to the high cost of implementing the convolution step, the method is not very practical especially when compared to our final version of VRDSMC. We have not explored subsampling methods (using a small and fixed number of particles N_a to evaluate the integral in Equation 6.15) which would make the cost $O(N_a N_{cell})$. Clearly further investigation is required before a direct and precise comparison can be made.

Chapter 7

Further VRDSMC Validation and Performance Evaluation

In this Chapter we use a demanding test problem to further explore the performance of the VRDSMC method. Specifically, by comparing to a DSMC solution of the same problem, we are able to obtain useful information about the convergence properties of VRDSMC as well as the computational savings compared to DSMC.

7.1 Introduction

We study the temperature profile due to viscous dissipation in a gas sheared by two walls moving in opposite directions (Couette flow). For low wall velocities and small Knudsen numbers the temperature profile is parabolic with a temperature jump at the walls due to kinetic effects [46]. The magnitude of the heating varies quadratically with the wall speed, making this problem particularly challenging to simulate at small Mach numbers. For example, the temperature increase is of the order of $2.55K$ for the case $U_W = 0.15c_0$, $Kn = 0.1$ studied here, while at $U_W = 0.05c_0$ it is $\Delta T = 0.25K$. This makes very accurate DSMC simulations essentially impossible for $U_W < 0.1$; as a result, a number of our VRDMSC-DSMC comparisons are performed on a coarse grid. We do, however, study the results of a significantly more refined VRDSMC calculation in the last section of this chapter and compare it to a DSMC solution.

To help us interpert the simulation results, we will start in the next section with a qualitative discussion of the contribution of the KDE procedure to numerical error in VRDSMC.

7.2 Sources of Error/Bias in VRDSMC

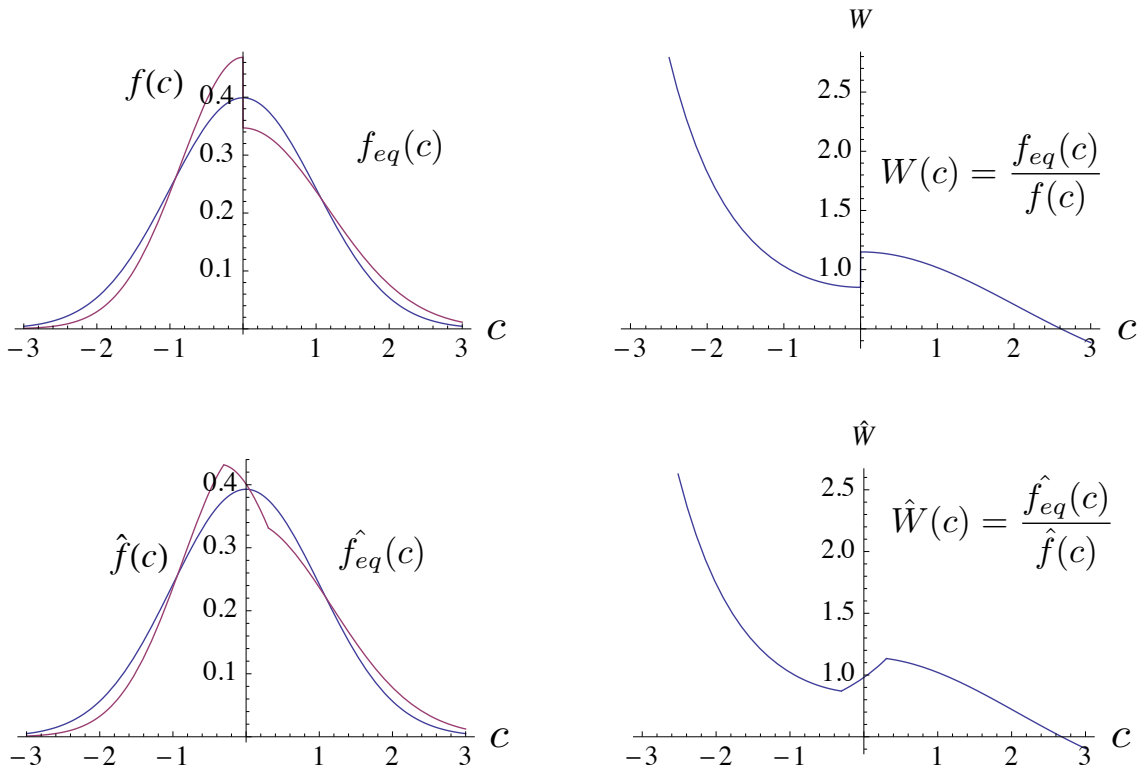

Figure 7-1: Simple illustration of KDE applied to one-dimensional functions. The figure in the top left shows the original functions; the figure below shows the expected shape of the KDE reconstruction of these functions. The plots in the right column are those of the corresponding weights for the exact function quotient and its KDE reconstruction.

Before we proceed to discuss some of the features that characterize how accurate a typical VRDSMC simulation is, let us start by giving a qualitative illustration of how a KDE estimation step introduces bias in the simulation. For the sake of simplicity

let us consider two simple one-dimensional functions $f(c)$ and $f_{eq}(c)$ that will be analogs to our non-equilibrium and equilibrium distribution functions respectively. Specifically, let us define a parameterized PDF g_α that is a Gaussian distribution of zero mean and variance α

$$g_\alpha(c) = \mathit{NormalDistribution}\,(0,\alpha) = \frac{e^{-\frac{c^2}{2\alpha^2}}}{\sqrt{2\pi}\alpha}$$

Let us assume our reference "equilibrium" state is one with a unit variance. i.e.

$$f_{eq}(c) = g_1(c)$$

Furthermore, let our "non-equilibrium" function be given by

$$f(c) = \begin{cases} g_{.85}(c) & c > 0 \\ g_{1.1}(c) & c < 0 \end{cases} \tag{7.1}$$

In other words, f is a discontinuous function, as shown in Figure 7.2. The weight function $W(c) = f_{eq}(c)/f(c)$ is shown in the top right plot of the figure. As expected, there is a discontinuity at $c = 0$.

To illustrate the KDE process, let us start by defining a normalized kernel K in one-dimension

$$K(\Delta c) = \begin{cases} \frac{1}{\varepsilon} & |\frac{\Delta c}{\varepsilon}| < 1 \\ 0 & \mathit{otherwise} \end{cases} \Rightarrow \int K(\Delta c) d\, c = 1$$

In the limit of a large number of samples, the re-constructed function $\hat{f}$ will approximate f by

$$\hat{f}(c) = \int K(c - c') f(c') dc'$$

Similarly we have for $\hat{f}_{eq}$

$$\hat{f}_{eq}(c) = \int K(c - c') f_{eq}(c') dc'$$

Qualitatively, the reconstructed functions $\hat{f}$ and $\hat{f}_{eq}$ resemble the original functions, but are generally "smeared"; this is especially clear at the discontinuity location in the original function f. In the bottom left plot in Figure 7.2 we see an example of these reconstructed functions calculated based on a kernel with $\varepsilon = 0.1$. If we use these new functions to estimate the ratio of probabilities $\hat{W}$, we will get

$$\hat{W}(c) = \frac{\hat{f}_{eq}(c)}{\hat{f}(c)}$$

that resembles the original $W(c)$, but as before without the discontinuity of the original weight function. Figure 7.2 further illustrates the distortion introduced due to the KDE procedure. The left column of the figure shows the ratio $\frac{\hat{W}(c)}{W(c)}$ for multiple values of the parameter ε. The right column compares for each ε the original equilibrium function with $\hat{W}f(c)$. The latter is the function that is effectively sampled when we use $\hat{W}$ instead of W to estimate equilibrium properties. Qualitatively, the distortion in the PDF is concentrated close to the discontinuity at the origin.

7.3 VRDSMC Bias as a Function of ε

In this section we will look at two trends that emerge as we vary VRDSMC parameters. Because the convergence properties as well as the ability of DSMC to provide accurate solutions of the BE have been well characterized [10, 20], the discussion below will compare VRDSMC to the equivalent DSMC simulations. Furthermore, due to the faintness of the temperature signal (an issue for DSMC) we have limited most of these simulations to moderately high velocities (Ma number $0.1 - 0.15$) and a very coarse grid (1 cell per mean free path) in order to practically produce these results in reasonable time.

We define the relative error as the error normalized by the maximum deviation from the equilibrium value of the quantity of interest

$$\text{Relative Error}(R(c)) = \frac{\overline{R}_{VR} - \overline{R}_{DSMC}}{\max\limits_{all\ cells}(\|\overline{R}_{VR} - \overline{R}_{eq}\|)}$$

When resolving small errors it is essential to make sure that enough time has passed so transient effects are completely negligible. For this reason, in the examples of this chapter we have chosen to start steady state sampling after 400 collision times.

Figure 7-3 shows the relative error in four hydrodynamic quantities of interest in this Couette flow problem. The figure clearly shows that the relative error monotonically decreases with decreasing ε, for all hydrodynamic qualities of interest. All simulations were run with $\overline{\|S_i\|} = 10$ which required $N_{cell} = 30,000$ for $\varepsilon \approx 0.109$ (lowest value) to $N_{cell} = 1,000$ for $\varepsilon = 0.34$ (highest value).

7.4 A Well Resolved Steady State Solution of an Induced Heating Problem at $Kn = 0.1$

In this section we show that accurate solutions with error levels well below the engineering 1% are possible with the proposed VRDSMC method. Figure 7.4 shows the steady state solution of a 1D Couette flow problem run with fine discretization and $U_W = 0.1\, c_0$. Specifically, the solution in the figure was obtained using 50 cells (giving a $\Delta y = 1/5\lambda$) a timestep that is half a cell traversal time (corresponding to $\Delta t = 1/10$ mean time between collisions) and $30,000$ particles per cell. As we saw in the examples of the last section, the induced heating and the corresponding heat flow and density changes are so low that they were particularly expensive to resolve. To achieve the low noise levels in DSMC that can be seen in the figure we needed to run multiple ensembles of the 1.5 million particle simulation for several days on a large computing cluster. And to ensure stability of the VRDSMC simulation, the average number of particles in KDE estimate was set to $\overline{\|S_i\|} = 10$ which corresponded to kernel width of $\varepsilon = 0.109$. The parameters used to solve this problem were particularly conservative and engineering results can probably be obtained with a significantly smaller number of particles by relaxing the KDE radius and setting $\overline{\|S_i\|}$ to $7 - 8$. This is particularly true for the flow velocity (u_x) in which the discrepancy between the two calculations is less than 0.04%.

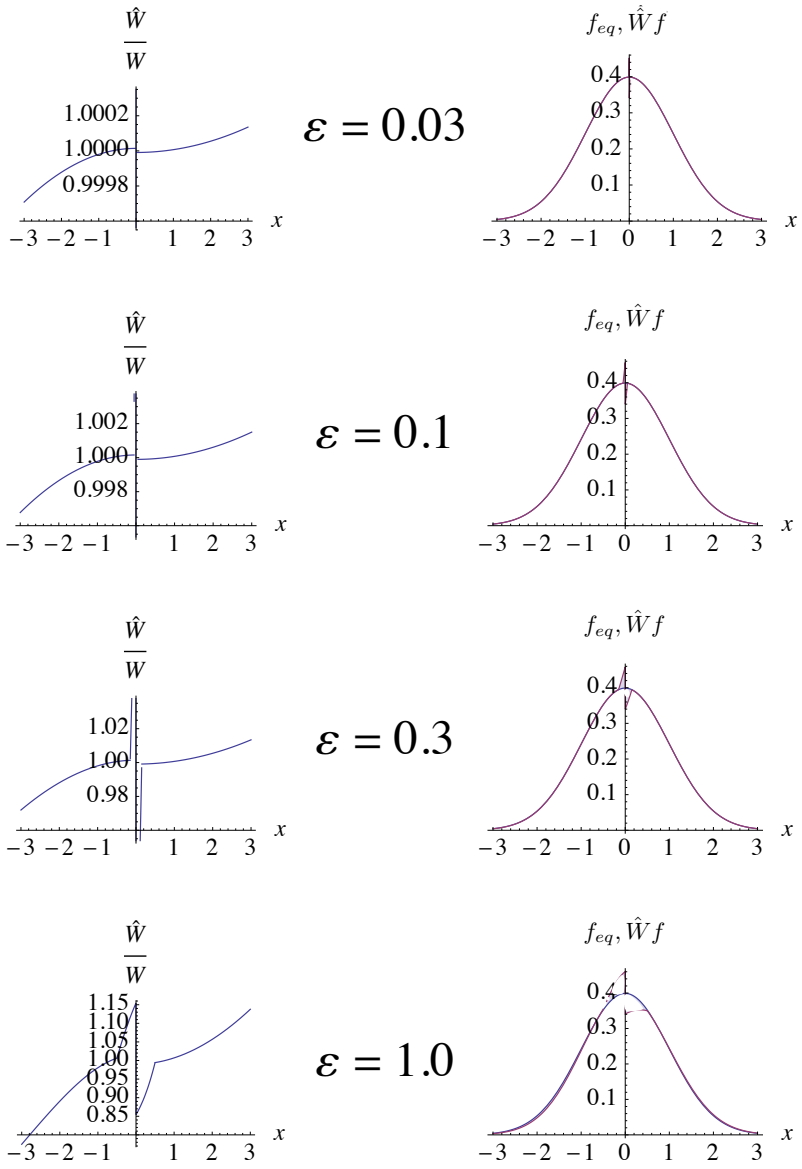

Figure 7-2: The introduction of bias in the estimates of weights is the primary cause of error in VRDSMC simulations. In this Figure, we see that as $\varepsilon \to 0$ the bias introduced goes to 0. In this figure the left column shows the ratio of $\hat{W}/W$, while the right column shows f_{eq} vs. $\hat{W}f$.

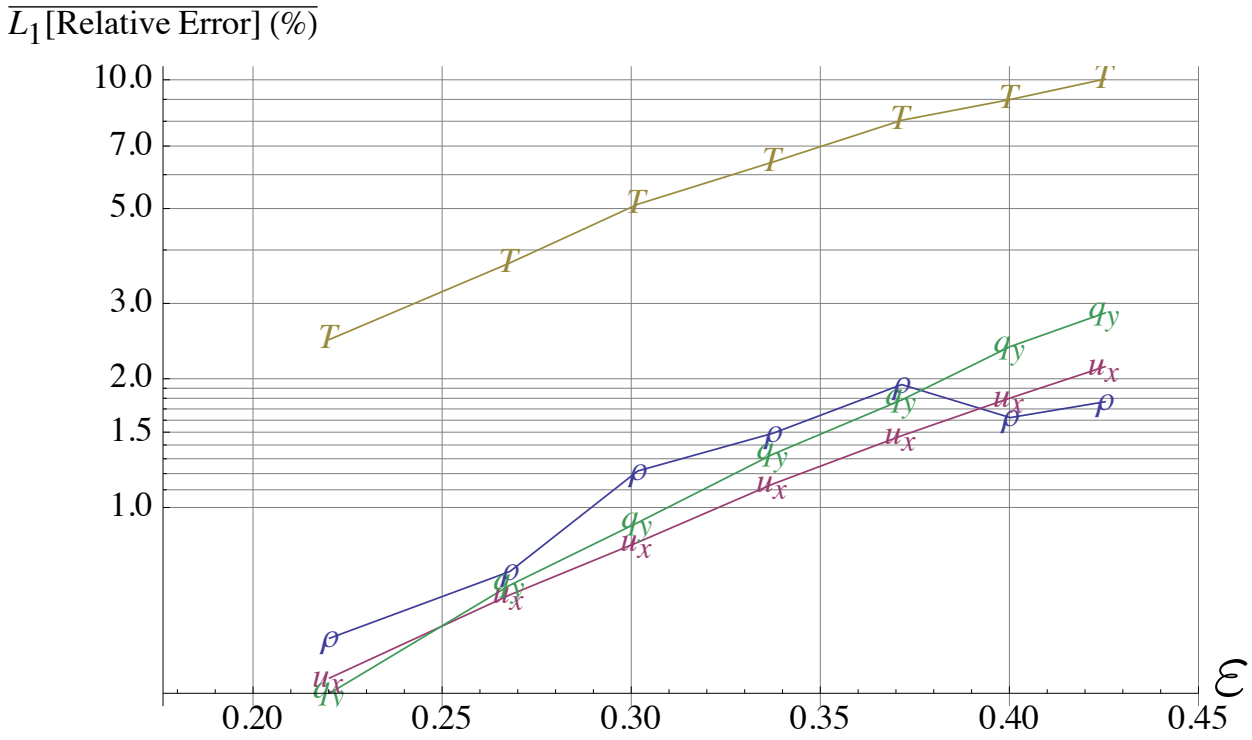

Figure 7-3: Log-linear plot of relative error for ρ, u_x, T and q_y for simulations with different KDE radii. As expected the bias decreases with decreasing ε for all the properties. These calculations were performed with 10 cells, $Kn = 0.1$ and $U_W = 0.15$.

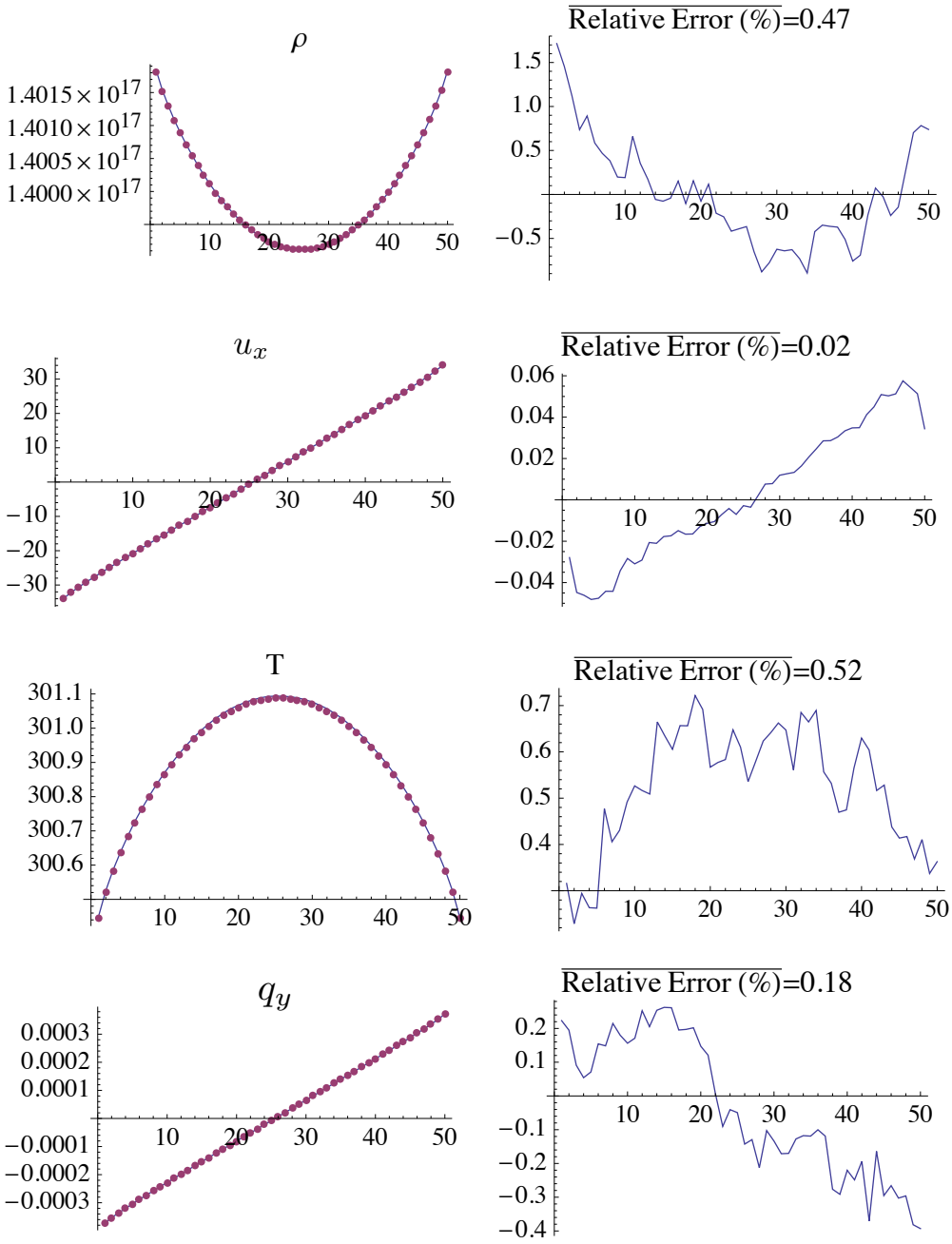

Figure 7-4: High resolution steady state solution for a 1D Couette showing the density, temperature, heat flux and velocity in the x-direction with $Kn = 0.1$. The simulation was performed using 50 simulation cells $N_{cell} = 30,000$, 50 cells and $\Delta t = \frac{1}{2}$ cell traversal time.

Chapter 8

Conclusions and Prospectus

We have developed an efficient method of simulating low-signal kinetic flows by incorporating variance reduction ideas within the prevalent particle simulation method for solving the BE. The resulting VRDSMC method retains most features that make DSMC so powerful, namely simplicity, flexibility and computational efficiency while overcoming the major disadvantage associated with DSMC, namely, poor computational performance for low-signal flows.

Our proposed method has three main ingredients: First, is the use of importance weights for variance reduction. The second critical ingredient is the use of the Kernel Density Estimation technique to limit the unbounded growth of the weight variance $\sigma^2\{W_i\}$ that was causing the method to fail. In particular, the KDE allowed us to reconstruct efficient numerical approximations of the particle distribution functions for both the main non-equilibrium simulation as well as the auxiliary equilibrium simulation, thus stabilizing the latter, albeit at the cost of some bias in the solution. The final important ingredient is the use of a local Maxwell-Boltzmann reference state *for performing the collision step* only. The adjustment of the reference state is not strictly required for the successful use of the method [59], but it makes the method much more attractive by substantially reducing the required number of particles per cell N_{cell}, especially for accurate low Kn applications.

In this work we were able to validate our method by comparing our results with DSMC solutions for a variety of 1D problems. We were able to show that VRDSMC

is able to produce accurate results with a reasonable number of particles per cell for a wide range of Knudsen numbers. The method is able to accurately reproduce transient as well as steady state results for a number of different types of boundary conditions, and to do so accurately for both the primary and secondary (induced) hydrodynamic fields.

One important recurring theme in this work is the fundamental tradeoff between variance and bias in the VRDSMC simulations. It appears that it is fairly straightforward to create a VRDSMC variant that does not introduce any bias but, unfortunately, such a method will be unstable and will be only of use for studying transient phenomena for the first few $(1-4)$ collision times. On the other hand, in the course of this investigation, we have studied a large number of stable variants that usually had an unacceptable amount of induced bias. The final variant proposed here achieves a delicate balance by creating what appears to be a stable simulation that has acceptable bias, that can additionally be made arbitrarily close to zero as long there are enough simulation particles in the domain. For steady-state, high resolution, low Kn applications that require a large number of particles and thus long integration times, Projective methods such as the one described in [55] may be useful for accelerating convergence to steady state, especially since such acceleration schemes tend to need a high accuracy transient solver to give accurate steady state solutions.

We tend to think of VRDSMC as a method that bridges two different approaches of solving kinetic gas flow problems. On one end of the spectrum we have DSMC which does not explicitly calculate the distribution function and espouses the goal of simulating the BE without explicitly using any information about the analytical form of the collision integral. In DSMC hydrodynamic properties are simply calculated from samples of the distribution and as such it is difficult to describe the fine details of the shape of the distribution function. On the other end of the spectrum we find deterministic solvers of the BE like [49, 7] which solve for the distribution function; these approaches are particularly useful for low-signal applications but still very expensive. VRDSMC produces estimates of the distribution function but only for the purpose of producing variance-reduced estimators that allow us to efficiently

measure moments of low-signal flows without resorting to a full (and very expensive) solution of the BE.

As pointed out in the introduction, the variance reduction approach introduced in this Thesis was inspired by work that started in the field of polymers [47]. Application of these ideas to kinetic gas flows was particularly challenging since stochastic particle-particle interactions are much more difficult to deal with than single particle thermalization from a *known* distribution. Noise problems due to thermal fluctuations is a very common limitation in many computational physics applications and there are have been other attempts to tackle this problem [44, 37]; it is fascinating to contemplate what other DSMC-like simulation methods this approach can be applied to.

We have spent considerable effort in ensuring that the VRDSMC method remains competitive with DSMC in terms of computational cost per timestep. Our best result is a method that scales as $O(N_{cell}Log(N_{cell}))$ in the worst case, which is close to DSMC which is an $O(N_{cell})$ method. In this sense more room for improvement exists. The work described in Chapter 5 has shown that using a local equilibrium distribution results in a substantial improvement in accuracy. Further improvements can come from performing a collision process using a local distribution which is as close to the actual distribution as possible. Given that stability problems and the associated KDE-induced bias are intimately linked to the low Knudsen regime, perhaps additional improvements can be achieved by using a Chapman-Enskog distribution [19]. Care however needs to be taken, because a Chapman-Enskog distribution is no longer invariant under the action of the collision operator.

Appendix A

Nearest Neighbor Routine Optimized for VRDSMC

The application of the KDE procedure to every particle accepted for collision requires finding the set S_i of all particles that are within εc_0 of particle i in velocity space. The direct method of looping over all particles in the cell and comparing their distance would make the main VRDSMC routine $O(N_{cell}^2)$, which would be too slow for practical applications. A standard method of solving the problem of finding nearest neighbors is to use a K-Dimensional (KD) tree to sort particles in a way that allows the retrieval of a particle's neighbors in $O(Log(N_{cell}))$ making VRDSMC much more practical especially for applications where N_{cell} is large. The first section will give a short overview of KD Trees; while the last section of this Appendix describes the actual procedure we used in finding neighbors in our VRDSMC implementation.

A.1 KD Trees and Finding Nearest Neighbors

A K-Dimensional tree is a standard computational geometry data structure [48, 43] that is commonly used to sort points into a tree-like structure that allows fast retrieval of neighboring points. Press et. al. [48] describe them as:

> "The defining principles of a KD tree are

- Boxes are successively partitioned into two daughter boxes.
- Each partition is along the axis of one coordinate.
- The coordinates are used cyclically in successive partitions.

In making the partition, the position of the "cut" is chosen to leave equal numbers of points on the two sides (or differing by one in the case of an odd number of points)."

We have found a number of problems when using this formulation to find nearest neighbor particles for our KDE step. To begin with, KD Trees are somewhat complicated to implement and require non-trivial algorithms to quickly sort and search for points. More importantly, there is no simple way of re-sorting particles whose velocities change as a result of collisions, without creating duplicate particles. In the end we found the method of the next section to be much simpler to understand, implement and debug. Most importantly, it also performs significantly faster than KD Trees, for the sizes of particles per cell one deals with in typical DSMC/VRDSMC calculations.

A.2 Our Nearest Neighbor Procedure

Due to the shortcomings discussed in the last section, we used a significantly simpler scheme to find nearest neighbors that is based on dividing the velocity space into bins of unequal size and sorting particles in them. The main advantage our procedure enjoys compared to a simple uniform cell gridding is that we exploit the fact that our calculation is close to equilibrium and so particle velocities are distributed in an approximately Gaussian manner. This means that with a judicious choice of bin boundaries we can create bins that on average contain the same number of particles, which results in efficient use of memory as well as the fast retrieval of neighbor particles within εc_0. Moreover, this method makes it easy for particles to change their location in velocity space without needing to create clone particles or re-sorting the particles in the cell.

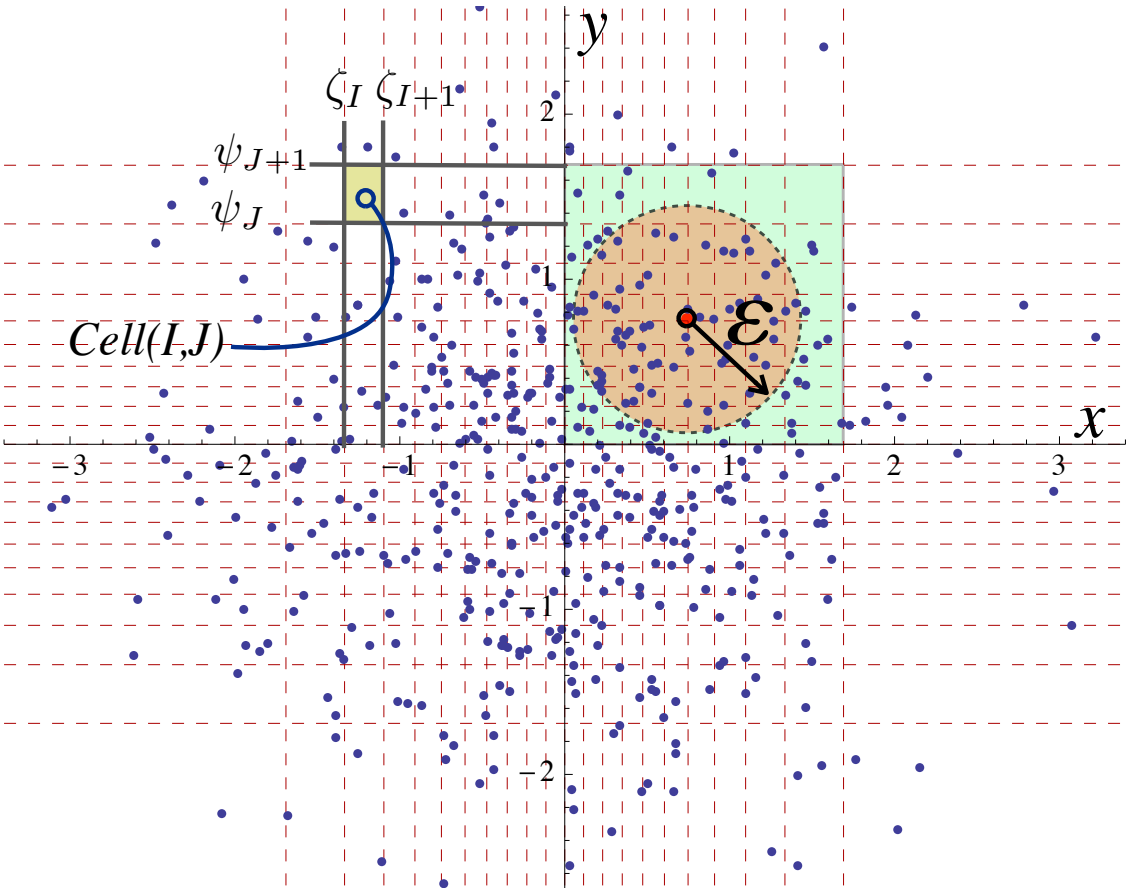

Figure A-1: Sketch illustrating our nearest neighbor procedure in 2D. The blue points in the figure are drawn from a normal distribution and the cells are chosen such that on average they contain the same number of particles.

To further detail how our procedure works we will discuss a simple 2D implementation; the 3D implementation follows directly. As a guide, the reader may refer to Figure A-1 which illustrates the basic concepts and definitions of this section.

We divide the domain into $\mathbb{M} \times \mathbb{M}$ cells such that each $Cell(I, J)$ is defined as the rectangle defined by $\zeta_I < x < \zeta_{I+1}$ and $\psi_J < y < \psi_{J+1}$ where $I = 0, 1, 2.....\mathbb{M} - 1$ and $J = 0, 1, 2.....\mathbb{M} - 1$. The cell sizes are variable; they are chosen such that, on average, they contain the same number of particles. Assuming the velocity distribution is the Gaussian g this can be achieved by using

$$\zeta_I = \mathbb{H}(I/\mathbb{M})$$

and

$$\psi_J = \mathbb{H}(J/\mathbb{M})$$

where $\mathbb{H}(x) = \mathbb{G}^{-1}(x)$ is the inverse of the cumulative distribution

$$\mathbb{G}(x) \equiv \int_{-\infty}^{x} g(c)dc$$

This result can be obtained by considering that the requirement of, on average, equal number of particles in each cell can be written as

$$g(c)dc = p(x)dx$$

where $p(x)$ is a uniform distribution on $[0, 1]$. A particle located at (x, y) will reside in $Cell(I(x), J(y))$ such that

$$I(x) = \mathbb{M}\,\lfloor \mathbb{G}(x) \rfloor$$
$$J(y) = \mathbb{M}\,\lfloor \mathbb{G}(y) \rfloor$$

where $\lfloor\rfloor$ denotes the floor function. Using these procedures we can efficiently implement the KDE reconstruction as follow:

- **Sorting:** loop through the N_{cell} particles in a cell placing particle i in $Cell(I(x_i), J(y_i))$. This operation needs $O(N_{cell})$ operations and is performed before the start of the collision step.

- **Updating particle velocities:** when a collision updates the velocity of a particle we update the lists of both (departing and arriving) cells.

- **Finding neighbors of particle i within distance ε:** We start by finding all cells that contain parts of a circle of radius ε centered at the particle i, as shown in green in Figure A-1. We then loop through these particles to see if they are in the neighborhood of i. Symbolically, this translates to looping through the

Table A.1: Simple benchmarks of VRDSMC for the same simulation but with a different number of cells in velocity space. This particular set of runs was for a $Kn = 0.1$ Couette flow simulation with $N_{cell} = 4,500$ with 30 cells in the y-direction and $\overline{\|S_i\|} = 10$. The run times were calculated on a notebook with a 2.93GHz Core 2 processor with $\mathbb{M}^3$ cells in velocity space.

Average # of particle per cell	$\mathbb{M}$	Execution Time(s)
3	12	54
6	10	55
10	8	56
20	7	58

set of particles $\{j \in cells(I,J); I_{Min} < I < I_{Max}, J_{Min} < J < J_{Max}\}$ where

$$I_{Min} = I(x_i - \varepsilon)$$

$$I_{Max} = I(x_i + \varepsilon)$$

and

$$J_{Min} = J(y_i - \varepsilon)$$

$$J_{Max} = J(y_i + \varepsilon)$$

to find the set of nearest neighbors S_i.

Since finding the nearest neighbors requires looping though all cells that may contain close particles, it is advantageous to select $\mathbb{M}$ such that an average cell has $\overline{\|S_i\|}$ particles in it. When this is done the cost of finding the set S_i becomes close to the average number of neighbors. Indeed, from our experiments (for example, see Table A.1) we have found that having on average a number of particles in a cell that is less than $\overline{\|S_i\|}$ gives a (small) performance advantage.

Appendix B

Variance-Reduced Estimates for The Common Hydrodynamic Variables

Like DSMC, hydrodynamic properties of interest in VRDSMC are calculated as expected values of moments of the distribution function that are sampled as the simulation evolves. The only difference between the two methods is that the variance-reduced property estimators are slightly more complex since they require the subtraction of the sampled equilibrium properties.

B.1 Background

Recall that in a particle simulation the distribution function is approximated as

$$f(\boldsymbol{x}, \boldsymbol{c}) = N_{Eff} \sum_{i=1}^{N} \delta(\boldsymbol{c}_i - \boldsymbol{c}) \delta(\boldsymbol{x}_i - \boldsymbol{x}) \tag{B.1}$$

and

$$f_{eq}(\boldsymbol{x}, \boldsymbol{c}) = N_{Eff} \sum_{i=1}^{N} W_i \delta(\boldsymbol{c}_i - \boldsymbol{c}) \delta(\boldsymbol{x}_i - \boldsymbol{x}) \tag{B.2}$$

where N is the number of simulation particles in the domain and N_{cell} is the number of particles in a cell.

As discussed in Section 2.2.1, integral $R = \int \int R(\boldsymbol{c}) f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x}$ can be written in

a variance-reduced form

$$R_{VR} = \int\int Rf(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x} - \int\int R(\boldsymbol{x},\boldsymbol{c})f_{eq}d\boldsymbol{c}d\boldsymbol{x} + \int\int R\, f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}$$

and can be estimated using

$$\overline{R}_{VR} = \overline{R} - \overline{R}_{eq} + \langle R\rangle_{eq} \tag{B.3}$$

which results in reduced variance when the equilibrium property is known deterministically.

In the following Sections we apply this approach and definitions B.1 and B.2 to find cell-averaged variance-reduced estimators for common hydrodynamic properties. In this Appendix we use the symbols $\mathfrak{n}$, $\boldsymbol{u}$, $\mathfrak{T}$, $\mathfrak{q}$ to denote expectation values of number density, velocity, temperature and heatflux averaged over the cell. As remarked in Section 1.3, in the main body of the Thesis we use one set of symbols, namely $n, \boldsymbol{u}, T$ and $\boldsymbol{q}$ to denote both hydrodynamic fields and their simulation (cell-averaged) estimates.

B.2 Variance-Reduced Density Estimator $\overline{\mathfrak{n}}_{VR}$

The approach can be readily applied to number density by starting with the definition of the average number density over a cell of volume V as

$$\mathfrak{n} = \frac{\int\int f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}}{V} \Rightarrow \overline{\mathfrak{n}} = \frac{N_{Eff}N_{cell}}{V} \tag{B.4}$$

and using a similar expression for the equilibrium simulation

$$\mathfrak{n}_{eq} = \frac{\int\int f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}}{V} \Rightarrow \bar{\mathfrak{n}}_{eq} = \frac{1}{V}N_{Eff}\sum_{i=1}^{N_{cell}} W_i$$

Recalling that the expected value for equilibrium is

$$\langle \mathfrak{n} \rangle_{eq} = \frac{1}{V} N_{Eff} N_{cell,eq}$$

where $N_{cell,eq} = \frac{n_{eq} V}{N_{eff}}$ is the number of particles expected in the equilibrium simulation, we obtain

$$\overline{\mathfrak{n}}_{VR} = \overline{\mathfrak{n}} - \overline{\mathfrak{n}}_{eq} + \langle \mathfrak{n} \rangle_{eq} \Rightarrow \overline{\mathfrak{n}}_{VR} = \frac{N_{Eff}}{V} \left(\sum_{i=1}^{N_{cell}} (1 - W_i) + N_{cell,eq} \right) \tag{B.5}$$

B.3 Variance-Reduced Velocity $\bar{\mathbf{u}}_{VR}$

Now proceed to treat the flow velocity, by focusing on the x-component, $\mathfrak{u}_x$, given by

$$\mathfrak{u}_x = \frac{\int \int c_x f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x}}{\int \int f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x}} \tag{B.6}$$

The denominator can be re-written using Equation B.4 as $N_{cell} N_{Eff} = \int \int f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x}$. Applying Equation B.3 on the numerator gives

$$\mathfrak{u}_x = \frac{1}{N_{cell} N_{Eff}} \left(\int \int c_x f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x} - \int \int c_x f_{eq}(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x} + \int \int c_x f_{eq}(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x} \right)$$

where

$$\int \int c_x f_{eq}(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x} = V \mathfrak{n}_{eq} \mathfrak{u}_{x,eq}$$

which is 0 in the case when the reference equilibrium state has no mean velocity.

Finally, our variance-reduced estimator for a cell becomes

$$\overline{\mathfrak{u}}_{x,VR} = \frac{1}{N_{cell} N_{Eff}} \left(N_{Eff} \sum_{i=1}^{N_{cell}} c_{x,i} (1 - W_i) + V \mathfrak{n}_{eq} \mathfrak{u}_{x,eq} \right) \tag{B.7}$$

In contrast, the estimator of $\mathfrak{u}_x$ for DSMC is

$$\mathfrak{u}_x = \frac{\int \int c_x f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x}}{\int \int f(\boldsymbol{x}, \boldsymbol{c}) d\boldsymbol{c} d\boldsymbol{x}} \Rightarrow \overline{\mathfrak{u}}_x = \frac{1}{N_{cell}} \sum_{i=1}^{N_{cell}} c_{x,i}$$

B.4 Variance-Reduced Temperature $\overline{\mathfrak{T}}_{VR}$

Turning our attention to temperature we follow a similar line of arguing. The temperature in a cell is defined as:

$$\mathfrak{T} = \frac{\frac{m}{3k}\int\int\left[(c_x-\mathfrak{u}_x)^2+(c_y-\mathfrak{u}_y)^2+(c_x-\mathfrak{u}_z)^2\right]f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}}{\int\int f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}}$$

This can be rewritten as follows

$$\begin{aligned}\mathfrak{T} = \frac{m}{3kN_{cell}N_{Eff}}\Bigg\{&\int\int\left(c_x^2+c_y^2+c_z^2\right)f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}\\&-\left(\overline{\mathfrak{u}}_{x,VR}^2+\overline{\mathfrak{u}}_{y,VR}^2+\overline{\mathfrak{u}}_{z,VR}^2\right)\int\int f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}\Bigg\}\end{aligned}$$

giving

$$\begin{aligned}\mathfrak{T} =\frac{m}{3kN_{cell}N_{Eff}}\Bigg\{&\int\int\left(c_x^2+c_y^2+c_z^2\right)f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}\\&-\left(\overline{\mathfrak{u}}_{x,VR}^2+\overline{\mathfrak{u}}_{y,VR}^2+\overline{\mathfrak{u}}_{z,VR}^2\right)N_{cell}N_{Eff}\Bigg\}\end{aligned}$$

By B.3 above we get

$$\begin{aligned}\mathfrak{T} =\frac{m}{3kN_{cell}N_{Eff}}\Bigg\{&\int\int\left(c_x^2+c_y^2+c_z^2\right)(1-W(\boldsymbol{x},\boldsymbol{c}))f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}\\&+\left\langle c^2\right\rangle_{eq}-\left(\overline{\mathfrak{u}}_{x,VR}^2+\overline{\mathfrak{u}}_{y,VR}^2+\overline{\mathfrak{u}}_{z,VR}^2\right)N_{cell}N_{Eff}\Bigg\}\end{aligned}$$

where $\langle c^2\rangle_{eq} = \int\int(c_x^2+c_y^2+c_z^2)f_{eq}d\boldsymbol{c}d\boldsymbol{x} = \frac{3k\,n_{eq}T_{eq}\,V}{m}$. Thus, the a variance-reduced estimator for $\mathfrak{T}$ can be written as

$$\begin{aligned}\overline{\mathfrak{T}}_{VR} = \frac{m}{3k}\Bigg\{&\frac{3k\,N_{cell,eq}T_{eq}}{m\,N_{cell}}-\left(\overline{\mathfrak{u}}_{x,VR}^2+\overline{\mathfrak{u}}_{y,VR}^2+\overline{\mathfrak{u}}_{z,VR}^2\right)\\&+\frac{1}{N_{cell}}\sum_{i=1}^{N_{cell}}(1-W_i)\left(c_{x,i}^2+c_{y,i}^2+c_{z,i}^2\right)\Bigg\}\end{aligned}\tag{B.8}$$

In comparison, the non-variance-reduced estimator $\mathfrak{T}$ is

$$\overline{\mathfrak{T}} = \frac{m}{3k}\left\{\frac{1}{N_{cell}}\sum_{i=1}^{N_{cell}}\left(c_{x,i}^2 + c_{y,i}^2 + c_{z,i}^2\right) - \left(\overline{\mathfrak{u}}_x^2 + \overline{\mathfrak{u}}_y^2 + \overline{\mathfrak{u}}_z^2\right)\right\} \tag{B.9}$$

The astute reader will note that B.8 is an estimator of a quantity $\mathfrak{T}$ based on an other set of quantities ($\mathfrak{u}_{VR}$) that themselves are estimated from the same dataset which means that it may be biased. Typically, a correction is introduced to make the estimator unbiased; no such correction is given here partially because it will only make a substantial difference for small N_{cell} which is not of interest in our work. The same holds for the other estimators introduced in this Appendix.

B.5 Variance-Reduced Heat Flux $\overline{\mathfrak{q}}_{y,VR}$

The cell-average heat flux in the direction y is given by

$$\mathfrak{q}_y = \frac{1}{V}\int\int\left(c_y - \mathfrak{u}_y\right)\frac{m}{2}\|\mathbf{c} - \mathfrak{u}^2\|f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}$$

If we take $\mathfrak{u}_z = 0$ we expand this equation to read

$$\begin{aligned}\mathfrak{q}_y = \frac{m}{2V}\int\int\Big(&c_x^2c_y + c_y^3 + c_yc_z^2 - 2c_xc_y\mathfrak{u}_x + c_y\mathfrak{u}_x^2 - c_x^2\mathfrak{u}_y \\ &- 3c_y^2\mathfrak{u}_y - c_z^2\mathfrak{u}_y + 2c_x\mathfrak{u}_x\mathfrak{u}_y - \mathfrak{u}_x^2\mathfrak{u}_y + 3c_y\mathfrak{u}_y^2 - \mathfrak{u}_y^3\Big)f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}\end{aligned} \tag{B.10}$$

In other words, the integral can be written as a sum of terms of the form

$$\int\int c_\eta^\alpha c_\zeta^\beta \mathfrak{u}_\zeta^\delta \mathfrak{u}_\eta^\chi f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}$$

where $\alpha,\beta,\chi,\delta \in \{0,1,2,3\}$ and $\eta,\zeta \in \{x,y,z\}$. The goal of producing a variance-reduced estimator of $\mathfrak{q}_y$ is now reduced to finding expressions for these individual terms. In the interest of brevity, we will treat here one of the terms

$$\int\int c_x^2c_yf(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}$$

with the understanding that other terms can be treated similarly. Using Equation B.3 above we obtain

$$\int\int c_x^2 c_y f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x} = \int\int c_x^2 c_y f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x} - \int\int c_x^2 c_y f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x} + \int\int c_x^2 c_y f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x}$$

and this can be written as

$$\int\int c_x^2 c_y (1 - W(\boldsymbol{x},\boldsymbol{c}))f(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x} + \int\int c_x^2 c_y f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c}d\boldsymbol{x} \tag{B.11}$$

The first term of Equation B.11 can be evaluated using

$$\begin{aligned} &N_{Eff}\sum_{i=1}^{N_{cell}}\left(c_{x,i}^2 c_{y,i}(1-W_i)\right) \\ &= N_{Eff}\left(\sum_{i=1}^{N_{cell}} c_{x,i}^2 c_{y,i} - \sum_{i=1}^{N_{cell}} W_i c_{x,i}^2 c_{y,i}\right) \end{aligned} \tag{B.12}$$

The second term of Equation B.11 can be evaluated by substituting the analytical value of f_{eq} and explicitly evaluating the integral. A relation that is useful in general is

$$\int c_\eta^\alpha c_\zeta^\beta f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c} = \frac{1}{\pi}\left((-1)^\alpha + 1\right)\left(1 + (-1)^\beta\right) n_{eq} 2^{\frac{1}{2}(\alpha+\beta-4)}\Gamma(\frac{\alpha+1}{2})\Gamma(\frac{\beta+1}{2})\left(\frac{kT_{eq}}{m}\right)^{\frac{\alpha+\beta}{2}} \tag{B.13}$$

which in our case would imply

$$\int c_x^2 c_y f_{eq}(\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c} = 0$$

and so our variance-reduced estimator of the first term is

$$\overline{c_x^2 c_y}_{VR} = N_{Eff} \left(\sum_{i=1}^{N_{cell}} c_{x,i}^2 c_{y,i} - \sum_{i=1}^{N_{cell}} W_i c_{x,i}^2 c_{y,i} \right) \tag{B.14}$$

We apply a similar approach to every term of B.10 to get a relation similar to B.14. The variance reduced estimator of the heat flux will simply be the summation of these terms.

Bibliography

[1] L. Boltzmann. Uber das warmegleich zwischen mehratomigen gasmolekulen. *Sitzungsber. Akad. Wiss. Math.-Natur. Klasse Wien,* 63:397–418, 1871.

[2] M. Knudsen. Die Gesetze der Molekularströmung und der inneren Reibungsströmung der Gase durch Rhören. *Annalen der Physik*, 333:75–130, 1909.

[3] M. Rosenblatt. Remarks on some nonparametric estimates of a density function. *Ann. Math. Statist.,* 27:832–837, 1956.

[4] W. G. Vincenti and C. H. Kruger. *Introduction to Physical Gas Dynamics,* Krieger, Florida, 1965.

[5] R. B. Bird, R. C. Armstrong and O. Hassager. *Dynamics of Polymeric Liquids,* John Wiley & Sons, New York, 1987.

[6] C. Cercignani. *The Boltzmann Equation and its Applications*, Springer-Verlag, New York, 1988.

[7] T. Ohwada, Y. Sone, and K. Aoki. Numerical analysis of the Poiseuille and thermal transpiration flows between two parallel plates on the basis of the Boltzmann equation for hard-sphere molecules. *Phys. Fluids A,* 1:2042, 1989.

[8] T. Ohwada, Y. Sone, and K. Aoki. Numerical analysis of the shear and thermal creep flows of a rarefied gas over a plane wall on the basis of the linearized Boltzmann equation for hard-sphere molecules. *Phys. Fluids A* 1:1588, 1989.

[9] P. Andreo. Monte Carlo techniques in medical radiation physics. *Phys. Med. Biol.,* 36:861–920, 1991.

[10] W. Wagner. A Convergence Proof for Bird Direct Simulation Monte Carlo Method for the Boltzmann Equation. *Journal of Statistical Physics*, 66:1011–1044, 1992.

[11] D. W. Scott. *Multivariate Density Estimation.* John Wiley & Sons, Marblehead, MA, 1992.

[12] G. A. Bird. *Molecular Gas Dynamics and the Direct Simulation of Gas Flows*, Clarendon Press, Oxford, 1994.

[13] H. C. Öttinger. *Stochastic processes in polymeric fluids,* Springer-Verlag, New York, 1995.

[14] I. D. Boyd. Conservative species weighting scheme for the direct simulation Monte Carlo method. *J. Thermophys. Heat Transfer,* 10:579–585, 1996.

[15] E. Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD Thesis, Stanford University, 1997.

[16] T. Rychlik. Randomized unbiased nonparametric estimates of nonestimable functionals. *Nonlinear Analysis,* 30:4385–4394, 1997.

[17] F. J. Alexander, A. L. Garcia and B. J. Alder. Cell size dependence of transport coefficients in stochastic particle algorithms. *Physics of Fluids*, 10:1540–1542, 1998.

[18] C. Ho and Y. Tai. Micro-Electro-Mechanical-Systems (MEMS) and fluid flows. *Annual Review of Fluid Mechanics*, 30:579–612, 1998.

[19] S. Chapman and T. G. Cowling. *Mathematical Theory of Non-uniform Gases*, Cambridge Mathematical Library, Cambridge, UK, 1999.

[20] N. G. Hadjiconstantinou. Analysis of discretization in the direct simulation Monte Carlo. *Physics of Fluids,* 12:2634–2638, 2000.

[21] A. Garcia and W. Wagner. Time step truncation error in direct simulation Monte Carlo. Physics of Fluids, 12:2621–2633, 2000.

[22] A. Garcia. *Numerical Method for Physics*. Prentice Hall, Englewood Cliffs, 2000.

[23] M. A. Gallis and J. R. Torczynski. The Application of the BGK Model in Particle Simulations. *34th AIAA Thermophysics Conference, Devver, CO*, June 2000.

[24] G. J. LeBeau and F. E. Lumpkin III. Application highlights of the DSMC Analysis Code (DAC) software for simulating rarefied flows, *Computer Methods in Applied Mechanics and Engineering,* 191:595–609, 2001.

[25] K. Aoki, Y. Sone, S. Takata, K. Takahashi and G. Bird. One-way flow of a rarefied gas induced in a circular pipe with periodic temperature distribution, *Rarefied Gas Dynamics*, ed. Bartel and Gallis, 940, AIP, 2001.

[26] M. N. Macrossan. A particle simulation method for the BGK equation, *AIP Conference Proceedings,* 585:426–433, 2001.

[27] M. A. Gallis, J. R. Torczynski and D. J. Rader. An approach for simulating the transport of spherical particles in a rarefied gas flow via the direct simulation Monte Carlo, *Physics of Fluids*, 13:3482, 2001.

[28] Q. Sun and I. D. Boyd. Simulation of Gas Flow over Micro-Scale Airfoils Using a Hybrid Continuum- Particle Approach. *33rd AIAA Fluid Dynamics Conference and Exhibit.* June 2003.

[29] N. G. Hadjiconstantinou, A. Garcia, M. Bazant and G. He. Statistical error in particle simulations of hydrodynamic phenomena. *Journal of Computational Physics,* 187:274–297, 2003.

[30] P. K. Kundu and I. M. Cohen. *Fluid Mechanics.* Elsevier, London, 2004.

[31] H. A. Al-Mohssen, N. G. Hadjiconstantinou. Arbitrary pressure chemical vapor deposition modeling using direct simulation Monte Carlo with nonlinear surface chemistry, *Journal of Computational Physics,* 198:617, 2004.

[32] P. Glasserman. *Monte Carlo methods in financial engineering*, Springer-Verlag, New York, 2004.

[33] G. Zou and R. D. Skeel. Robust Variance Reduction for Random Walk Methods. *SIAM J. Sci. Comput.,* 25:1964–1981, 2004.

[34] N. G. Hadjiconstantinou, and H. A. Al-Mohssen. A Linearized Kinetic Formulation Including a Second-Order Slip Model for an Impulsive Start Problem at Arbitrary Knudsen Numbers. *Journal of Fluid Mechanics,* 533:47–56, 2005.

[35] J. Chun and D. Koch. A direct simulation Monte Carlo method for rarefied gas flows in the limit of small Mach number. *Physics of Fluids*, 17:107107, 2005.

[36] L. L. Baker and N. G. Hadjiconstantinou. Variance reduction for Monte Carlo solutions of the Boltzmann equation. *Physics of Fluids*, 17:051703, 2005.

[37] J. Eapen, J. Li and S. Yip. Statistical field estimators for multiscale simulations. *Physical Review E*, 72:056712, 2005.

[38] A. Doucet and X. Wang. Monte Carlo methods for signal processing: a review in the statistical signal processing context. *IEEE Signal Processing Magazine,* 22:152–170, 2005.

[39] N. G. Hadjiconstantinou. The limits of Navier-Stokes theory and kinetic extensions for describing small-scale gaseous hydrodynamics. *Physics of Fluids*, 18:111301, 2006.

[40] C. Cercignani. *Slow Rarefied Flows: Theory and Application to Micro-Electro-Mechanical Systems*, Springer, New York, 2006.

[41] M. S. Ivanov, A. V. Kashkovsky, S. F. Gimelshein, G. N. Markelov, A. A. Alexeenko, Y. A. Bondar, G. A. Zhukova, S. B. Nikiforov, P. V. Vaschenkov. SMILE system for 2D/3D DSMC computations. *Proceedings of 25th International Symposium on Rarefied Gas Dynamics, St. Petersburg, Russian*, July 21–28, 2006.

[42] J. L. Barber, K. Kadau, T. C. Germann, P. S. Lomdahl, B. L. Holian and B. J. Alder. Atomistic simulation of the Rayleigh-Taylor instability. *Journal of Physics: Conference Series,* 46:58–62*,* 2006.

[43] E. Langetepe and G. Zachmann. *Geometric data structures for computer graphics*, A K Peters, Ltd., Wellesley, MA, 2006.

[44] J. Eapen. *Transport modeling of simple fluids and nano-colloids: thermal conduction mechanisms and coarse projection*, PhD Thesis, Massachusetts Institue of Technology, 2006.

[45] A. L. Garcia. Estimating hydrodynamic quantities in the presence of microscopic fluctuations. *Comm. app. Math. And Comp. Sci.,* 1:53–78, 2006.

[46] Y. Sone. *Molecular Gas Dynamics: Theory, Techniques, and Applications.* Birkhauser, Boston, 2007.

[47] H. C. Öttinger, H. van den Brule, and M. Hulsen. Brownian configuration fields and variance-reduced CONNFFESSIT. *Journal of Non-Newtonian Fluid Mechanics*, 70:255–261, 2007.

[48] W. Press, S. Teukolsky, W. Vetterling and B. Flannery. *Numerical Recipes*, Cambridge University Press, 2007.

[49] L. L. Baker. *Efficient numerical methods for solving the Boltzmann equation for small scale flows*, PhD Thesis, MIT, 2007.

[50] T. M. Homolle and N. G. Hadjiconstantinou. Low-variance deviational simulation Monte Carlo. *Physics of Fluids*, 19:041701, 2007.

[51] T. M. Homolle. *Efficiant particle methods for solving the Boltzmann equation.* SM Thesis, MIT, 2007.

[52] T. M. M. Homolle and N. G. Hadjiconstantinou. Low-variance deviational simulation Monte Carlo. *Physics of Fluids,* 19:041701, 2007.

[53] T. M. M. Homolle and N. G. Hadjiconstantinou. A low-variance deviational simulation Monte Carlo for the Boltzmann equation. *Journal of Computational Physics,* 226:2341–2358, 2007.

[54] T. M. M. Homolle and N. G. Hadjiconstantinou. A low-variance deviational simulation Monte Carlo for the Boltzmann equation. *Journal of Computational Physics*, 226, 2341–2358, 2007.

[55] H. A. Al-Mohssen, N. G. Hadjiconstantinou and I. Kevrekidis. Acceleration methods for coarse-crained numerical solution of the Boltzmann equation. *J. Fluids Eng.* 129:908–912, 2007.

[56] Z. Anwar. *Enabling Microscopic Simulators to Perform System-Level Analysis of Viscoelastic Flows.* PhD Thesis, MIT, 2008.

[57] F. Attilio. Advances in multiphysics simulation and experimental testing of MEMS. *Computational & experimental methods in structures*, Vol. 2, 2008.

[58] W. Wagner. Deviational particle Monte Carlo for the Boltzmann equation. *Monte Carlo Methods and Applications,* 14:191–268, 2008.

[59] H. A. Al-Mohssen and N. G. Hadjiconstantinou. Yet another variance reduction method for direct Monte carlo simulations of low-signal flows. *Rarified Gas Dynamics: Proceedings of the 26th International Symposium on Rarified Gas Dynamics. AIP Conference Proceedings*, 1084:257–262, 2008.

[60] L. L. Baker and N. G. Hadjiconstantinou. Variance-reduced Monte Carlo solutions of the Boltzmann equation for low-speed gas flows: A discontinuous Galerkin formulation. *International Journal for Numerical Methods in Fluids*, 58:381–402, 2008.

[61] L. L. Baker and N. G. Hadjiconstantinou, Variance-reduced particle methods for solving the Boltzmann equation. *Journal of Computational and Theoretical Nanoscience*, 5:165–174, 2008.

[62] L. L. Baker and N. G. Hadjiconstantinou. Variance-Reduced Particle Methods for Solving the Boltzmann Equation. *Journal of Computational and Theoretical Nanoscience*, 5:165–174, 2008.

[63] R. Y. Rubinstein and D. P. Kroese. *Simulation and the Monte Carlo Method,* Wiley-Interscience, Malden, 2008.

[64] B. John and M. Damodaran. Computation of head–disk interface gap micro flowfields using DSMC and continuum–atomistic hybrid methods. *Int. J. Numer. Meth. Fluids,* 61:1273–1298, 2009.

[65] S. Ramanathan and D. L. Koch. An efficient direct simulation Monte Carlo method for low Mach number noncontinuum gas flows based on the Bhatnagar–Gross–Krook model. *Phys. Fluids*, 21:033103, 2009.

[66] G. A. Radtke and N. G. Hadjiconstantinou. Variance-reduced particle simulation of the Boltzmann transport equation in the relaxation-time approximation, *Physical Review E,* 79:056711, 2009.

[67] C. D. Landon. *Weighted Particle Variance Reduction of Direct Simulation Monte Carlo with the Bhatnagar-Gross-Krook Collision Integral.* SM Thesis, MIT, 2010.

[68] H. A. Al-Mohssen and N. G. Hadjiconstantinou. Low-Variance direct Monte Carlo using importance weights. *Mathematical Modelling and Numerical Analysis*, In press, 2010.

[69] G. N. Fayad. *Untitled.* PhD Thesis, MIT, 2010.