Consistent Modeling of Hypersonic Nonequilibrium Flows using Direct Simulation Monte Carlo

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

CHONGLIN ZHANG

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Advisor: Thomas E. Schwartzentruber

August, 2013

Acknowledgements

First and foremost, I would like to acknowledge my Ph.D advisor Professor Thomas E. Schwartzentruber for his constant encouragement and support. When I first came to Minnesota, Prof. Schwartzentruber introduced me to the field of molecular gas dynamics and the direct simulation Monte Carlo method, which provided me with the opportunity to study gas flows from a completely new perspective. During the past five years of my study here, he always gives me the freedom to attack research problems using my own approach and idea. This is always a good training and experience for me and will be a valuable asset in my future career. I would also like to thank his continuous help, encouragement and many useful suggestions in writing and presentation, which are always challenging tasks for me. I would also like to thank him for providing me with the financial support during my study here, and encouraging me to attend workshops and conferences. Without such support, I could not have completely focused the attention on my study and research.

I would also like to acknowledge the help of my other dissertation committee members, including Prof. Graham Candler, Prof. Krishnan Mahesh, Prof. Chris Hogan, and Prof. Sanford Lipsky. I attended courses taught by most of them, and through these courses I gained solid background on various subjects. I would also like to acknowledge many other professors here. From their excellent course instructions, I have the opportunities to expand my interests in different topics. Many professors and staffs in the Aerospace engineering and Mechanics department have also helped me a lot during my study here, their help are much appreciated.

I would also like to acknowledge all current and former colleagues in the Molecular Gas Dynamics research group. Special thanks goes to Dr. Paolo Valentini and Dr. Paul Norman. We have had many helpful discussions on various research questions and

other topics on a daily basis. Through such discussions, each of us gained a much deeper understanding of many subtle concepts and questions. I would also like to thank all my friends in the AEM department, at the U, and back home. Without them, I could not have a balance between work and life. Special thanks goes to Dr. Xiaochuan Chai, who helped me a lot since I first arrived here, and also my long time friend Dr. Hao Dang from Columbia University, who always provided me with good suggestions.

Finally, I would like to express my deepest thanks to my parents and my sisters. They have encouraged and supported me in every possible way. They always let me do everything in my own way and according to my interest. Over the past several years, no matter where I was, I always knew I have a whole family standing on my back to support me. Knowing this makes me always have the confidence to pursue my dreams over the past years, especially during the difficult times in my study and research.

Abstract

Hypersonic flows involve strong thermal and chemical nonequilibrium due to steep gradients in gas properties in the shock layer, wake, and next to vehicle surfaces. Accurate simulation of hypersonic nonequilibrium flows requires consideration of the molecular nature of the gas including internal energy excitation (translational, rotational, and vibrational energy modes) as well as chemical reaction processes such as dissociation. Both continuum and particle simulation methods are available to simulate such complex flow phenomena. Specifically, the direct simulation Monte Carlo (DSMC) method is widely used to model such complex nonequilibrium phenomena within a particle-based numerical method. This thesis describes in detail how the different types of DSMC thermochemical models should be implemented in a rigorous and consistent manner. In the process, new algorithms are developed including a new framework for phenomenological models able to incorporate results from computational chemistry. Using this framework, a new DSMC model for rotational energy exchange is constructed. General algorithms are developed for the various types of methods that inherently satisfy microscopic reversibility, detailed balance, and equipartition of energy in equilibrium. Furthermore, a new framework for developing rovibrational state-to-state DSMC collision models is proposed, and a vibrational state-to-state model is developed along the course. The overall result of this thesis is a rigorous and consistent approach to bridge molecular physics and computational chemistry through stochastic molecular simulation to continuum models for gases in strong thermochemical nonequilibrium.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and motivation

Hypersonic flows involve strong thermal and chemical nonequilibrium due to steep gradients in gas properties in the shock layer, wake, and next to vehicle surfaces. Accurate simulation of hypersonic nonequilibrium flows requires consideration of the molecular nature of the gas including internal energy excitation (translational, rotational, and vibrational energy modes) as well as chemical reaction processes such as dissociation. Both continuum and particle simulation methods are available to simulate such complex flow phenomena. Specifically, the direct simulation Monte Carlo (DSMC) method [15] is widely used to model such complex nonequilibrium phenomena within a particle-based numerical method.

The DSMC method is a stochastic particle simulation method that simulates the Boltzmann equation and is therefore accurate for gas flows ranging from free-molecular to continuum conditions. In a DSMC simulation, millions of simulation particles are tracked through a computational grid, where each simulation particle represents a large number of (identical) real molecules. A computational grid divides space into small regions (cells that are sized approximately to the local mean-free-path) to group nearby simulation particles. Each simulation particle has a center-of-mass molecular velocity vector, a species type, and classical or quantized internal energies (rotation, vibration). In this manner, particle properties within computational cells can be sampled to obtain velocity and internal energy distribution functions as well as macroscopic average

quantities such as density, bulk velocity, and translational/rotational/vibrational temperatures. Particles are moved in a straight line along their velocity vectors without interacting for small time steps limited to a fraction (approximately 0.2) of the local mean-collision-time. After such free-flight movement (including diffuse or specular reflections from surfaces), particles are sorted into cells and collisions are considered between particles located within the same cell. These assumptions and molecular simulation procedures (separating movement and collision processes) are rigorously supported by the nature of dilute gases. The main modeling component of the DSMC method is the manner in which collisions are performed in a stochastic manner within each computational cell. Two models are required; a model to determine the collision rate (the average number of collisions performed in a given DSMC cell during a given time step), and a model to determine the outcome of those collisions.

At the core of a DSMC code is a relatively simple collision probability expression that has the general form,

$$P_{\text{coll}} = \sigma_T(g) g \Delta t W_p / V \ , \tag{1.1}$$

where g is the relative collision velocity of a collision pair, V is the volume of the DSMC collision cell, Δt is the time step, W_p is the weight of each particle (each particle represents W_p number of real molecules), and $\sigma_T(g)$ is the "collision cross-section" of the colliding pair. This cross-section is arguably the main modeling input for the DSMC method and can take many functional forms, although the most common is that of a power-law function of the relative velocity ($\sigma_T(g) \approx \sigma_{ref}/g^{2\zeta}$, where σ_{ref} is a reference hard-sphere cross-section and ζ is a model parameter). For example, collisions with higher relative velocities tend to have smaller cross-sections since the molecules interact through their potential field for shorter times. The probability expression above essentially compares the volume swept out by an interaction pair during a single time step (accounting for the particle weight, W_p) with the volume of the computational cell. Within a given collision cell that contains N simulation particles (N is typically 50-100), there are $\frac{N(N-1)}{2}$ possible collision pairs. Each possible collision pair is essentially chosen to undergo a collision with the above probability. The summation of applying P_{coll} to all pairs in the cell determines the total number of collisions that occur within a given cell during a given time step which is directly related to the theoretical gas

collision rate (number of collisions per time per volume), ν, and establishes the physical validity of the DSMC method. For an equilibrium gas, the collision rate is expressed as

$$\nu = \frac{1}{2} n^2 \int_0^{+\infty} f(g) \sigma_T(g) g dg \ , \tag{1.2}$$

where $f(g)$ is the Maxwell-Boltzmann distribution function and n is the number density. Indeed, the equation for P_{coll} (which is applied to the $\frac{N(N-1)}{2}$ collision pairs within each cell) is the stochastic equivalent of the equation for ν, except that particles within a DSMC cell are not restricted to a Maxwell-Boltzmann distribution and can take on any value in velocity space. As a result, the DSMC method can model the collision rate in a nonequilibrium gas and reduces to the theoretical collision rate under equilibrium conditions. Furthermore, the collision cross-section, $\sigma_T(g)$, can be rigorously connected to an interatomic potential that governs collision dynamics. For monatomic gases, the cross-section ($\sigma_T(g)$) is analytically linked to the gas transport properties (such as viscosity), through the well-established Chapman-Enskog theory [15, 32, 56, 89]. Thus, for monatomic gases, there is a well-established connection between interatomic potentials, the cross-section and collision rate, and macroscopic transport properties. The DSMC method has been shown over many years to accurately predict such flows under a wide range of nonequilibrium and equilibrium conditions.

However, when mixtures of diatomic species are considered that involve internal energy (rotation and vibration) relaxation and chemical reaction processes, corresponding collision probabilities or cross-sections must be introduced. The typical approach in DSMC is that collision pairs selected to collide (pairs that have already been accepted with probability P_{coll} above) have a further probability of exchanging internal energy or chemically reacting. Thus, these pairs are further tested with probabilities such as P_{rot}, P_{vib}, and P_{react}, or they may even be tested for a state-specific transition from an initial quantized rovibrational state to a final rovibrational state with a "state-to-state" probability. Such internal energy exchange and chemical reaction models and DSMC algorithms are less established than for monatomic gases. The focus of this thesis is a rigorous analysis of these physical models and associated numerical algorithms for the DSMC method.

Although a number of articles found in the literature are devoted to the modeling

of internal energy relaxation and chemical reaction processes in the DSMC method, generally, there is a lack of detail regarding how to consistently and correctly implement such models within a DSMC program, and discrepancies and inconsistencies are evident within the literature. Specifically, there are three general types of models available for internal energy relaxation processes.

1. The first type of DSMC model (termed "phenomenological") is constructed using an existing continuum level model. This includes many widely used phenomenological models for internal energy exchange/relaxation, and chemical reactions [15, 19, 21–26, 33, 34, 47, 91, 95, 100]. In the continuum model, the internal energy relaxation process (rotational, or vibrational energy relaxation) is described by a characteristic time for relaxation (for example τ_{rot} and τ_{vib}). A DSMC collision model is constructed, using values of P_{rot} and P_{vib} that result in simulated relaxation processes that are consistent with the continuum model in the near-equilibrium limit. Analogous reaction models can link P_{react} to a continuum reaction rate k_f, for chemically reacting flows. Thus, such phenomenological DSMC models are constructed to be consistent with a continuum model in the equilibrium limit, however, can be used within DSMC on a collisional basis for highly nonequilibrium flows.

2. The second type of DSMC model is one that is constructed using some detailed molecular simulation data from theoretical or computational chemistry, however, this data is incorporated into a DSMC collision model that is phenomenological in nature (using P_{vib}, P_{rot}, and P_{react} for example) [87, 100].

3. The third type of DSMC model is a state-resolved model [27, 28, 37, 38, 58, 72]. In this type of model, theoretical or computational chemistry results determine all state-to-state transition probabilities (or cross-sections), which are directly used within DSMC. This type of method directly simulates realistic quantum state transitions representative of real molecular collisions and therefore models nonequilibrium flows at the most fundamental level.

When implementing phenomenological models for rotational/vibrational relaxation and chemical reactions in DSMC, different authors (and different DSMC codes) generally use different collision selection procedures [15, 35, 44, 48]. For example, three

widely used inelastic collision selection procedures include pair selection[22, 67], particle selection permitting double relaxation[15], and particle selection prohibiting double relaxation[48]. However, it has been shown that to correctly simulate a specified relaxation rate, the inelastic collision probability expression used within the DSMC method (P_{rot}, P_{vib}, and P_{react}) must depend on, or be specific to, the selection procedure[44, 48, 67]. Only the consistent use of a selection procedure and its corresponding probability expression will result in a DSMC simulation reproducing the desired relaxation rate. This subtlety can complicate the transferability of probability expressions (collision models) between DSMC implementations and can also lead to inconsistent comparisons of DSMC simulations with continuum simulations involving internal energy relaxation in the near-equilibrium limit. Moreover, for gas mixtures, the collision selection procedure can affect the simulated relaxation process , since some selection procedures inherently couple the relaxation probabilities and internal energy redistribution processes of the different gas species.

The second type of DSMC model typically constructs the probability of internal energy transfer or chemical reaction to be a function of collision properties only, and the model is informed in some manner by more detailed molecular computations. Such models have encountered difficulty in achieving detailed balance and equipartition of energy at equilibrium conditions [1, 21, 22, 33, 88]. A systematic procedure is needed to address such issues.

Only very recently have realistic state-to-state models for DSMC been presented in the literature [27, 58, 92], and no rigorous model framework has been presented that addresses microscopic reversibility, detailed balance, the connection to gas transport properties, and the overall consistency of the state-to-state cross-section database.

This thesis describes in detail how the different types of DSMC models should be implemented in a rigorous and consistent manner. In the process, new algorithms are developed including a new framework for phenomenological models able to incorporate results from computational chemistry. Using this framework, a new DSMC model for rotational energy exchange is constructed. General algorithms are developed for the various types of methods that inherently satisfy microscopic reversibility, detailed balance, and equipartition of energy in equilibrium. Furthermore, a new framework for

developing rovibrational state-to-state DSMC collision models is proposed, and a vibrational state-to-state model is developed along the course. The overall result of this thesis is a rigorous and consistent approach to bridge molecular physics and computational chemistry through stochastic molecular simulation to continuum models for gases in strong thermochemical nonequilibrium.

1.2 Outline of the thesis

In Chapter 2, the issue of how particles are randomly selected for collisions within a given DSMC cell during a given time step is investigated. Especially for gas mixtures, the algorithm used to select collision pairs and update the internal energies of either both particles or just one particle involved in the collision, has a direct effect on the *simulated* internal energy transfer rates. The same internal energy relaxation model with the same parameter inputs may lead to different simulated energy transfer rates if different selection procedures are used. This subtle, but important, aspect of DSMC is crucial for the transferability of DSMC collision models between DSMC implementations (codes) and also for obtaining consistency with continuum simulations (CFD) in the near-equilibrium limit.

In Chapter 3, existing phenomenological models for rotational-vibrational-chemical nonequilibrium flows are analyzed. Such existing models have not been consistently compared in the literature. Often the selection procedure used is not identified and different combinations of rotation, vibration, and dissociation models have been used in the comparisons, which makes conclusions regarding the differences of a single model (for example, the vibration model only) difficult to interpret.

In Chapter 4, the physical mechanisms of rotational excitation and relaxation are analyzed and a new rotational model (for nitrogen) is developed for the DSMC method. Previously, the translational-rotational relaxation rate has been modeled as either a function of the translational temperature or simply modeled as a constant value. However, prior studies have also highlighted that the rate of rotational energy relaxation may be a strong function of the direction towards the equilibrium state (compressing flows vs. expanding flows) and also the magnitude of the deviation from equilibrium. Chapter 4 presents a new nonequilibrium-direction-dependent (NDD) rotational energy

exchange model for DSMC and multi-temperature Navier-Stokes methods. The DSMC NDD model is based only on collision-quantities and reduces to a rotational collision number model in the continuum limit, and is applicable for use with the Jeans relaxation equation in CFD calculations. The model is formulated based on recent Molecular Dynamics (MD) simulations of rotational relaxation in nitrogen [87] and accounts for the dependence of the relaxation rate on the direction to the equilibrium state. This enables a single parameterization of the model to accurately simulate rotational relaxation in nitrogen for both compressing and expanding flows, unlike the widely used Parker model.

In Chapter 5, the formulation and implementation of state-to-state models in the DSMC method is investigated. Only very recently have realistic state-to-state models for DSMC been presented in the literature [27, 58, 92], and no rigorous model framework has been presented that addresses microscopic reversibility, detailed balance, the connection to gas transport properties, and the overall consistency of the state-to-state cross-section database. Chapter 5 presents a general framework for implementing state-to-state collision models in the DSMC method. We further develop a vibrational state-to-state DSMC collision model using the transition probability data provided by the forced harmonic oscillator (FHO) model [3, 68, 77, 78, 82, 96]. State-to-state DSMC simulation results are compared with corresponding master equation simulation results.

Finally, in Chapter 6, the issue of grid generation and complex geometry associated with the DSMC method is investigated. A significant obstacle for new DSMC developers is the fact that the computational grid must be adapted to the local mean-free-path (i.e. the density field), which is a result of the simulation itself. This requires adaptive mesh refinement (AMR) in order to avoid significant user-time for iterative grid generation. The requirement of AMR and also the desirability of decoupling the flow field grid (adapted to the local mean-free-path) from the body-surface mesh naturally leads to the use of Cartesian flow grids and "cut-cell" algorithms to imbed complex geometry. Although the implementation of multi-level Cartesian flow grids is straight-forward, general algorithms for cutting complex 3D geometry from a background grid are quite challenging and tedious in terms of implementation. Chapter 6 details a complete framework of cut-cell algorithms used to handle the decoupled flow field grids and surface meshes, for general Cartesian grid based DSMC implementations.

Chapter 2

Particle Selection Procedures for DSMC Calculations of Gas Mixtures

2.1 Chapter Introduction

In order to obtain accurate numerical solutions for gas flows in thermochemical nonequilibrium involving finite-rate translational-rotational-vibrational-chemical relaxation, various DSMC models [13, 15, 19, 21–26, 29, 33, 34, 44–47, 53, 88, 91, 94, 95, 100] have been proposed to account for the energy exchange/relaxation between translational, rotational and vibrational energy modes, as well as chemical reactions. Most of these models are phenomenological [15], in that a fraction of the collisions are considered as inelastic/reactive, and are subject to transfer of energies between different energy modes (usually between translational and internal energy modes) and/or species change; all the remaining collisions are considered as elastic and only involve translational energy exchange between the colliding particles. In other words, during the DSMC simulation, every collision is considered for internal energy exchange with an inelastic collision probability P_i (P_{rot} for rotational inelastic collisions, or P_{vib} for vibrational inelastic collisions). The objective is to match the DSMC simulated relaxation rate in accordance

with that specified by the relaxation time or collision number in the continuum relaxation equation (Eq. 2.1). The same idea is used to model finite rate chemical reaction process in the DSMC method, where a chemical reaction probability P_{react} is used to determine whether each molecular collision is reactive or not. For gas mixtures involving energy relaxation of different energy modes and chemical reactions of multiple species, the overall relaxation and chemical reaction processes are very complex; on the other hand, the collision probabilities simulated by DSMC will also depend on the procedure used to implement these models. As a result, we need a systematic approach to implement all the different phenomenological models, to ensure that the correct relaxation rate, and chemical reaction rate are simulated in the DSMC method.

When implementing phenomenological models for rotational/vibrational relaxation and chemical reactions in the DSMC method, different authors (and different DSMC codes) generally use different inelastic collision selection procedures [15, 35, 44, 48]. For example, three widely used inelastic collision selection procedures include pair selection [22, 67], particle selection permitting double relaxation [15], and particle selection prohibiting double relaxation [48]. However, it has been shown that to correctly simulate a specified relaxation rate, the inelastic collision probability expression used within the DSMC method must depend on, or be specific to, the selection procedure [44, 48, 67]. Only the consistent use of a selection procedure and its corresponding probability expression will result in a DSMC simulation reproducing the desired relaxation rate. This subtlety can complicate the transferability of probability expressions (collision models) between DSMC implementations and can also lead to inconsistent comparisons of DSMC simulations with continuum simulations involving internal energy relaxation in the near-equilibrium limit.

In continuum simulations, relaxation processes are usually modeled by the Jeans equation or Landau-Teller equation which have the same form [48],

$$\frac{dE}{dt} = \frac{E^*(t) - E(t)}{\tau} \quad , \tag{2.1}$$

where E(t) is the average energy at time t of either the rotational or vibrational mode associated with ζ degrees of freedom, and τ is the characteristic relaxation time of the energy mode. $E^*(t)$ is the instantaneous equilibrium energy of the energy mode, which

is defined according to the instantaneous translational temperature $T_t(t)$, as,

$$E^*(t) = \frac{\zeta}{2} k_B T_t(t) \quad . \tag{2.2}$$

The characteristic relaxation time τ in Eq. 2.1, is determined as a function of the equilibrium gas temperature in previous theoretical and experimental studies [69, 74]. τ is usually expressed as a function of the mean collision time τ_c, and an inelastic collision number Z in the following manner,

$$\tau = \tau_c Z \quad . \tag{2.3}$$

Thus, a rotational or vibrational inelastic collision number (Z_{rot}, Z_{vib}) is used to specify the relaxation rate, which in general, may be a function of temperature. To simulate these relaxation processes in DSMC, each particle involved in a collision is considered for internal energy exchange with an inelastic collision probability,

$$P_{rot} = f(\zeta_{rot,A}, \zeta_{rot,B}, \zeta_t, Z_{rot}) \quad \text{or} \quad P_{vib} = f(\zeta_{vib,A}, \zeta_{vib,B}, \zeta_t, Z_{vib}) \ . \tag{2.4}$$

Here ζ_t represents the translational degrees of freedom of the collision pair. ζ_{rot} and ζ_{vib} are the effective internal degrees of freedom of the rotational and vibrational energy modes (corresponding to collision partners A and B) participating in the inelastic collision [13, 15, 44, 48]. As an example, for the variable hard sphere (VHS) molecular model [15] with a temperature dependent viscosity exponent of ω, the translational degrees of freedom participating in a collision is $\zeta_t = 5 - 2\omega$. In general, the exact form of Eq. 2.4 is specific to the inelastic collision *selection procedure* used to model the relaxation process [15, 35, 44, 48]. A number of these inelastic collision selection procedures have been discussed in several papers, including Lumpkin *et al.* [67], Haas *et al.* [48] and Gimelshein *et al.* [44].

The effect of the collision selection procedure on the simulated relaxation process is most significant for gas mixtures, since some selection procedures inherently couple the relaxation probabilities and internal energy redistribution processes of the different species. To remedy this, Haas *et al.* [48] constructed a framework for rotational and vibrational relaxation suitable for mixtures. However, to relate the DSMC inelastic

collision probabilities (P_{rot}, P_{vib}) with the corresponding collision numbers (Z_{rot}, Z_{vib}), a set of quadratic equations must be solved during each collision for the case when the collision numbers are temperature dependent. In an article on vibrational relaxation, Gimelshein *et al.* [44] used a modified version of the framework by Haas *et al.* [48] that uses a single random number to determine the probabilities of rotational and vibrational energy exchange between the colliding partciles and does not require the solution of quadratic equations. However, the *simulated* probabilities do not explicitly appear in the algorithm by Gimelshein *et al.* [44], rather they result from inequalities evaluated using the random number. As a result, it is not clear that the techniques of Haas *et al.* [48] and Gimelshein *et al.* [44] produce the same simulated relaxation for gas mixtures and, furthermore, the modified algorithm by Gimelshein *et al.* [44] was not tested on a gas mixture in the article.

In this chapter, we present a modification to the approach taken by Haas *et al.* [48], which also removes the requirement of solving a set of quadratic equations during each collision, but where the simulated probabilities of rotational and vibrational energy relaxation appear explicitly in the algorithm. Ultimately, through this new algorithm, we are able to analytically prove the equivalence of all three algorithms, and demonstrate the ability to accurately simulate specified internal energy relaxation rates in gas mixtures. Section 2.2.1 summarizes the most widely used inelastic collision selection procedures with specific discussion regarding simulated relaxation processes in gas mixtures. In section 2.2.2, the sequential probability selection procedure is described and the simulated relaxation process for gas mixtures is analytically proven to be equivalent to that produced by the procedures of Haas *et al.* [48] and Gimelshein *et al* [44]. A discussion of the computational efficiency of the three procedures is also presented in this section. The ability of the proposed selection procedure to simulate specified internal energy relaxation rates is demonstrated in section 2.2.3 through comparison with analytical solutions for isothermal relaxation, and conclusions are drawn in section 2.3.

2.2 A Sequential Probability Collision Selection Procedure Suitable for Gas Mixtures

2.2.1 Existing inelastic collision selection procedures

For clarity, we summarize the three widely used inelastic collision selection procedures as they apply to rotational relaxation. For inelastic collisions involving vibrational energy exchange, the procedures are identical, only the internal energy mode is altered.

(A) **Pair selection** [22, 67]: The collision pair is tested for rotational inelastic collision with a probability, and once the collision pair is selected, the energy of both particles in the pair is redistributed. Specifically, the total collision energy $E_T = E_t + E_{r,A} + E_{r,B}$ is redistributed between translational and rotational modes, as E'_t and E'_r, using the Borgnakke-Largsen (BL) model [19], with $E_T = E'_r + E'_t$. The post collision rotational energy E'_r is then distributed between the two collision partners as $E'_{r,A}, E'_{r,B}$ again using the BL model with $E'_r = E'_{r,A} + E'_{r,B}$ [15]. Each of the three inelastic collision selection procedures has a limit on the smallest collision number that can be simulated in DSMC, which corresponds to a simulated rotational inelastic collision probability of 1. Suppose the translational degrees of freedom of the collision pair is $\zeta_t = \zeta_{t_{A|B}}$, and the rotational degrees of freedom is ζ_{rot}, with $\zeta_{rot} = \zeta_{rot,A} + \zeta_{rot,B}$, where $\zeta_{rot,A}$ and $\zeta_{rot,B}$ are the rotational degrees of freedom of the two participating molecules A and B. For the pair selection procedure, as developed in previous studies [48, 67], the probability should be set as $P_{rot} = \frac{\zeta_t + \zeta_{rot}}{\zeta_t} \frac{1}{Z_{rot}}$. For rotational relaxation, this means the smallest rotational collision number is $Z_{rot}^{limit} = \frac{\zeta_t + \zeta_{rot}}{\zeta_t}$, which is approximately 2 for nitrogen. Similarly, there is a smallest vibrational collision number that can be simulated.

(B) **Particle selection permitting double relaxation** [15]: Each particle in the collision pair is tested with a probability for rotational inelastic collision individually. If the first particle is selected for an inelastic collision, the BL model is used to redistribute the total collision energy between the translational energy of the *pair* and the rotational energy of only the *selected particle*. The second particle in the pair is then tested for a rotational inelastic collision. If selected, the

total collision energy used in the BL procedure includes the redistributed translational energy of the pair from the first collision and only the rotational energy of the second particle. Thus, if both particles are selected for an inelastic collision, there is some degree of coupling between their relaxation processes. For this selection procedure, suppose the molecule (particle) under consideration has rotational degrees freedom ζ_{rot} ($\zeta_{rot} = \zeta_{rot,A}$ or $\zeta_{rot,B}$), similarly, the smallest rotational collision number that can be simulated in DSMC is $Z_{rot}^{limit} = \frac{\zeta_t+\zeta_{rot}}{\zeta_t}$. For molecule-molecule collisions, this number is slightly smaller compared to the first selection procedure, since ζ_{rot} only includes the degrees of freedom of one molecule. Similarly, there is a smallest vibrational collision number that can be simulated in DSMC.

(C) **Particle selection prohibiting double relaxation** [48]: The two particles in the collision pair are tested with a probability for rotational inelastic collision individually. In this case, the total collision energy is always the sum of the relative translational energy of the collision pair and the rotational energy of only the particle being considered. However, if one particle is selected to undergo a rotational inelastic collision, the other particle is not tested for an inelastic collision, and the relaxation process for the collision pair ends. Otherwise, the same procedure is then applied to the second particle in the pair. For this selection procedure, to have all types of inelastic collisions correctly simulated in DSMC, we need to satisfy $P_{rot,1}+P_{rot,2}+P_{vib,1}+P_{vib,2} < 1$ as given later in Eq. 2.9. Compared to the two previous selection procedures this will correspond to a larger Z_{rot}^{limit} and Z_{vib}^{limit} that can be simulated in DSMC. Moreover, since the inequality needs to be satisfied, all values in the sum must be determined as a whole.

Selection procedure (A) couples the relaxation probabilities and energy redistribution processes for collision pairs of different species, and thus couples the simulated relaxation process. Although not as direct as procedure (A), procedure (B) also couples the energy redistribution processes of species when both particles are selected for rotational relaxation (i.e. double relaxation). Furthermore, when both rotational and vibrational relaxation processes are considered, a sequential test for rotational followed by vibrational inelastic collision is required for selection procedures (A) and (B). This

will inherently couple the rotational and vibrational relaxation processes. Although such coupling may seem physically realistic, it is stressed that the DSMC collision models discussed here are phenomenological and are constructed to reproduce specified internal energy relaxation rates (collision numbers, Z) for a given energy mode and species interaction. For this reason, in order to accurately simulate the relaxation process of mixtures, Haas *et al.* [48] and Gimelshein *et al.* [44] recommend using selection procedure (C) to decouple the rotational and vibrational relaxation of different species.

The logical steps followed by selection procedure (C) are depicted in Fig. 2.1 (this is constructed by Haas *et al.* [48], and is later adapted by Gimelshein *et al.* and used in a modified form [44]). First, one of the two particles (from the collision pair) is selected with equal chance and is tested for rotational relaxation with a specified collision probability $P_{rot,1}$ (suppose particle 1 is selected first). As discussed above, if the particle is chosen to undergo an inelastic collision, the BL model is used to redistribute the post collision energy and the relaxation of the current pair will finish. Only when the first particle does not undergo a rotational inelastic collision, will the second particle in the collision pair be tested for a rotational inelastic collision with probability $P_{rot,2}$. Again, the relaxation of the current pair will end if the second particle undergoes an inelastic collision. Only when the second particle is not chosen for a rotational inelastic collision, will the two particles be tested individually for a vibrational inelastic collision. In the same manner, the first particle will be tested with a specified probability $P_{vib,1}$ and only if not selected will the second particle be tested with probability $P_{vib,2}$.

As initially proposed by Haas *et al.* [48], the probabilities $P_{rot,1}$, $P_{rot,2}$, $P_{vib,1}$ and $P_{vib,2}$, are obtained by solving the set of quadratic equations listed in Eq. 2.5. These equations relate the probabilities used within DSMC (P_{rot} and P_{vib}) to specified continuum rotational collision numbers (Z_{rot} and Z_{vib}). If the collision numbers are not constant (for example they may be temperature dependent), then the quadratic equation must be solved for every collision. We also note that if the order of testing for rotational and vibrational inelastic collisions is reversed, then the quadratic equations

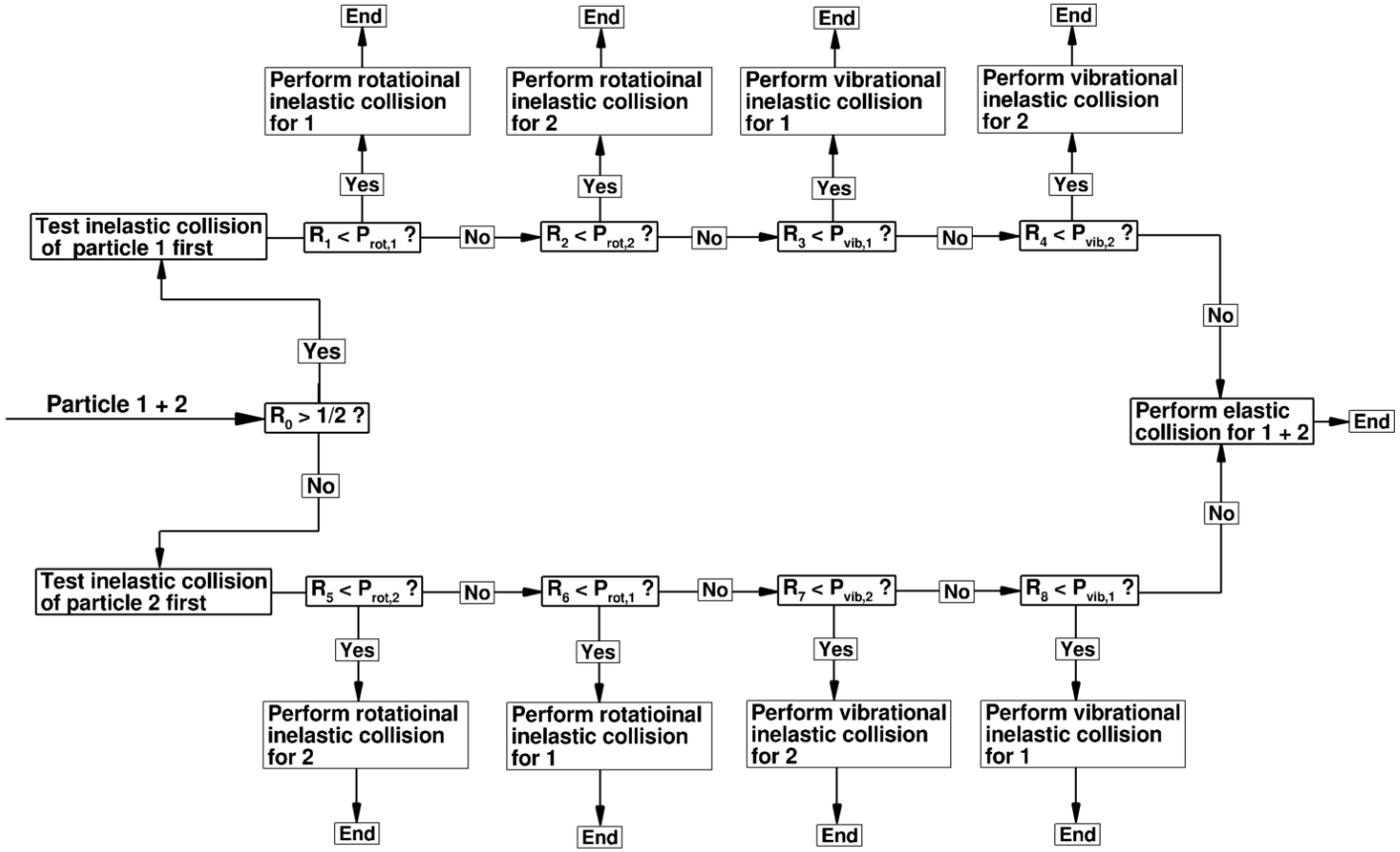


Figure 2.1: The process to select an inelastic collision using selection procedure (C) as constructed by Haas *et al.*. In the figure, $R_n (n = 0, 1, ..., 8)$ are uniform random numbers between 0 and 1, and $P_{rot,i}, P_{vib,i}$ ($i = 1$ or 2) are the rotational and vibrational inelastic collision probabilities used in DSMC for particle i.

and instances of $P_{rot,1}$, $P_{rot,2}$, $P_{vib,1}$, and $P_{vib,2}$ must be changed accordingly.

$$\frac{1}{2}P_{rot,1}^2 - \left(\frac{1}{2}\frac{\zeta_t+\zeta_{rot,1}}{\zeta_t}\frac{1}{Z_{rot,1}} - \frac{1}{2}\frac{\zeta_t+\zeta_{rot,2}}{\zeta_t}\frac{1}{Z_{rot,2}} + 1\right)P_{rot,1} + \frac{\zeta_t+\zeta_{rot,1}}{\zeta_t}\frac{1}{Z_{rot,1}} = 0 \tag{2.5a}$$

$$\frac{1}{2}P_{rot,2}^2 - \left(\frac{1}{2}\frac{\zeta_t+\zeta_{rot,2}}{\zeta_t}\frac{1}{Z_{rot,2}} - \frac{1}{2}\frac{\zeta_t+\zeta_{rot,1}}{\zeta_t}\frac{1}{Z_{rot,1}} + 1\right)P_{rot,2} + \frac{\zeta_t+\zeta_{rot,2}}{\zeta_t}\frac{1}{Z_{rot,2}} = 0 \tag{2.5b}$$

$$\frac{1}{2}(1-P_{rot,1})(1-P_{rot,2})P_{vib,1}^2 - \left(\frac{1}{2}\frac{\zeta_t+\zeta_{vib,1}}{\zeta_t}\frac{1}{Z_{vib,1}} - \frac{1}{2}\frac{\zeta_t+\zeta_{vib,2}}{\zeta_t}\frac{1}{Z_{vib,2}} + (1-P_{rot,1})(1-P_{rot,2})\right)P_{vib,1} + \frac{\zeta_t+\zeta_{vib,1}}{\zeta_t}\frac{1}{Z_{vib,1}} = 0 \tag{2.5c}$$

$$\frac{1}{2}(1-P_{rot,2})(1-P_{rot,1})P_{vib,2}^2 - \left(\frac{1}{2}\frac{\zeta_t+\zeta_{vib,2}}{\zeta_t}\frac{1}{Z_{vib,2}} - \frac{1}{2}\frac{\zeta_t+\zeta_{vib,1}}{\zeta_t}\frac{1}{Z_{vib,1}} + (1-P_{rot,2})(1-P_{rot,1})\right)P_{vib,2} + \frac{\zeta_t+\zeta_{vib,2}}{\zeta_t}\frac{1}{Z_{vib,2}} = 0 \tag{2.5d}$$

The specific notation used in Eq. 2.5 and throughout the remainder of this chapter requires a careful description. As depicted in Fig. 2.1, before starting the collision procedure, the two particles in the pair must be assigned a number (either particle 1 or 2). We use a subscript, i, to denote the particle numbering of the pair ($i = 1$ or $i = 2$). We further note that i does not denote a specific particle type, thus particles $i = 1$ and $i = 2$ may be the same, or different, particle type (monatomic, diatomic, or polyatomic). In this manner, all parameters denoted by a subscript i (such as $P_{rot,i}$, $Z_{rot,i}$, $\zeta_{rot,i}$, etc.) are specific to the relaxation process of particle i. For example, $\zeta_{rot,i}$ and $\zeta_{vib,i}$ are the rotational and vibrational degrees of freedom of only particle i, whereas ζ_t represents the available translational degrees of freedom of the collision pair (and thus has no subscript). Furthermore, $Z_{rot,i}$ and $Z_{vib,i}$ are the rotational and vibrational inelastic

collision numbers specific to the relaxation of particle i during a collision with the other particle in the pair. For example, if both particles are of the same type (A) then the collision numbers for the two particles would be equal (i.e. $Z_{rot,1} = Z_{rot,2} = Z_{rot,A|A}$). However, if the two particles were of different types (A for $i = 1$, and B for $i = 2$, as an example), then the collision numbers would be $Z_{rot,1} = Z_{rot,A|B}$ and $Z_{rot,2} = Z_{rot,B|A}$ where in general, $Z_{rot,A|B}$ may be specified as not equal to $Z_{rot,B|A}$.

2.2.2 Formulation of the sequential probability selection procedure

2.2.2.1 Proposed Modification to the Selection Procedure of Haas *et al.* [48]

In this section, we propose a modification to the particle selection prohibiting double relaxation procedure. We keep the structure of selection procedure (C) as shown in Fig. 2.1 unchanged, while using different expressions to calculate $P_{rot,i}$ and $P_{vib,i}$ appearing in the figure. With the modified procedure, we no longer need to solve the quadratic equations in Eq. 2.5 during each collision when the collision numbers are not constant. In the original procedure [48], the values of $P_{rot,i}$ appearing in both top and bottom branches of Fig. 2.1 are exactly equal (the same applies for $P_{vib,i}$), and this is what necessitates the solution of a quadratic equation. In our modified selection procedure (called the sequential probability selection procedure) the values of $P_{rot,i}$ and $P_{vib,i}$ appearing in the upper and lower branches of Fig. 2.1 are not the same. Rather, we calculate the appropriate probability for the specific particle currently under consideration for inelastic relaxation, taking into account the previous branching steps already completed and their associated probabilities.

Specifically, for the sequential probability selection procedure, when particle $i = 1$ is tested first (the upper branch of Fig. 2.1), the probabilities $P_{rot,1}$, $P_{rot,2}$, $P_{vib,1}$, $P_{vib,2}$ used in the acceptance-rejection technique for DSMC inelastic collision selection, are calculated from the following expressions:

$$A = 1, \qquad P_{rot,1} = AF_{rot,1} \ ; \tag{2.6a}$$

$$B = \frac{A}{1 - P_{rot,1}}, \qquad P_{rot,2} = BF_{rot,2} \ ; \tag{2.6b}$$

$$C = \frac{B}{1 - P_{rot,2}}, \qquad P_{vib,1} = CF_{vib,1} \ ; \tag{2.6c}$$

$$D = \frac{C}{1 - P_{vib,1}}, \qquad P_{vib,2} = DF_{vib,2} \ , \tag{2.6d}$$

where

$$F_{rot,i} = \frac{\zeta_t + \zeta_{rot,i}}{\zeta_t} \frac{1}{Z_{rot,i}} \ , \tag{2.7}$$

and

$$F_{vib,i} = \frac{\zeta_t + \Gamma_i}{\zeta_t} \frac{1}{Z_{vib,i}}. \tag{2.8}$$

For vibration, if a continuous energy distribution is used [15], then $\Gamma_i = \zeta_{vib,i}$. Whereas for the simple harmonic oscillator (SHO) discrete energy level model [15], as discussed by Gimelshein *et al.* in Ref. 44, $\Gamma_i = \xi_v(T)^2 exp(\theta_v/T)/2$, where $\xi_v(T) = \frac{2\theta_v/T}{exp(\theta_v/T)-1}$, T is the temperature, and is usually set as the cell averaged translational temperature, i.e., $T = T_t$, and θ is the characteristic temperature of vibration (where all parameters are specific to particle i).

Alternatively, when particle $i = 2$ is tested first (the lower branch of Fig. 2.1), the probabilities $P_{rot,1}$, $P_{rot,2}$, $P_{vib,1}$, $P_{vib,2}$ in Fig. 2.1 are still calculated using Eq. 2.6, only now with subscripts 1 and 2 interchanged.

For such particle selection prohibiting double relaxation selection procedure, the constraint [44]

$$F_{rot,1} + F_{rot,2} + F_{vib,1} + F_{vib,2} < 1 \tag{2.9}$$

must be satisfied. While Eq. 2.9 is satisfied for the majority of nonequilibrium flow problems, as evident from Eqs. 2.5 and 2.6, if $F_{rot,i}$ were to approach 0.5 there would be a vanishing number of particles available to be tested for vibrational relaxation and $P_{vib,i}$ may become larger than unity. Thus for generality, it is recommended to test for vibrational relaxation first, followed by rotational relaxation, since $P_{vib,i}$ is typically much smaller than $P_{rot,i}$. This ensures that vibrational relaxation remains accurate even in extreme cases with very fast rotational relaxation. Changing the order of rotational and vibrational relaxation only requires interchanging the subscripts (*rot*, *vib*) in Eqs. 2.5 and 2.6. Similarly, when chemical reactions are considered, the collision pair should be tested for a chemical reaction first. In this situation, to avoid the slight bias introduced to the rotational and vibrational relaxation rate by the chemical reaction

probability P_{react}, the value A in Eq. 2.6 can be modified to $A = 1/(1 - P_{react})$.

It should be noted that, we can further simplify Eq. 2.6 depending on the value of A. In the case $A = 1$, we have $P_{rot,1} = F_{rot,1}$, $P_{rot,2} = \frac{F_{rot,2}}{1-F_{rot,1}}$, $P_{vib,1} = \frac{F_{vib,1}}{1-F_{rot,1}-F_{rot,2}}$ and $P_{vib,2} = \frac{F_{vib,2}}{1-F_{rot,1}-F_{rot,2}-F_{vib,2}}$. While in the case $A = 1/(1 - P_{react})$, we have $P_{rot,1} = \frac{F_{rot,1}}{1-P_{react}}$, $P_{rot,2} = \frac{F_{rot,2}}{1-F_{rot,1}-P_{react}}$, $P_{vib,1} = \frac{F_{vib,1}}{1-F_{rot,1}-F_{rot,2}-P_{react}}$ and $P_{vib,2} = \frac{F_{vib,2}}{1-F_{rot,1}-F_{rot,2}-F_{vib,1}-P_{react}}$. These expressions can be used in the actual implementation of sequential probability selection procedure.

Finally, collision quantity dependent models [21–23, 33, 88, 95, 100] do not have a direct relationship between the collision probability P and a collision number Z. For such models, F_i is now a function of some collision quantities (collision energies for example) and can simplify replace the corresponding F_i in Eq. 2.6. In this manner, the prescribed relaxation behavior (collision quantity based) can be exactly simulated with no coupling between internal energy modes or species. Ultimately, the sequential probability particle selection procedure (detailed in Eqs. 2.6 to 2.8) is aimed at phenomenological DSMC models (either collision-number or collision-quantity based) which combine collision probabilities with the Borgnakke-Larsen model for energy redistribution. For state-to-state DSMC models, such as the recent model presented by Boyd *et al.* [27], further consideration or perhaps a different strategy may be required to correctly simulate the specified state-to-state processes within a DSMC simulation.

2.2.2.2 Equivalence of Sequential Probability and Original Haas *et al.* [48] Selection Procedures

The sequential probability particle selection procedure (Eq. 2.6) achieves the same relaxation rate as the original Haas *et al.* selection procedure [48] (Eq. 2.5). To show this, we start with the Eq. B11 from Ref. 48,

$$F_{A|B} = \frac{\zeta t_{A|B} + \zeta r_A}{\zeta t_{A|B}} \frac{1}{Z_{A|B}} \quad , \tag{2.10}$$

which represents the fraction of collisions (on average) that are required to be inelastic for A particles in order to achieve a relaxation rate consistent with Jeans equation using a collision number $Z_{A|B}$. This is also elaborated in Appendix A.2. Given the notation described at the end of section 2.2.1, Eq. 2.10 is identical to Eq. 2.7 (and Eq. 2.8 in the

case of vibration). The objective is to ensure that the collision probabilities (P) used within the acceptance-rejection portion of the DSMC algorithm actually result in the correct inelastic collision fraction (F) being simulated.

In the original Haas *et al.* selection procedure [48], $F_{A|B}$ should be the sum of the probability of two types of inelastic collisions: the relaxation of particle A in the upper branch of Fig. 2.1, and the relaxation of particle A in the lower branch of Fig. 2.1. For the case of rotation, the expression takes the following form (Eq. B2 of Ref. 48 with the right hand side written in current notation):

$$F_{A|B} = \frac{1}{2}P_{rot,1} + \frac{1}{2}(1 - P_{rot,2})P_{rot,1} = P_{rot,1}(1 - \frac{1}{2}P_{rot,2}) \quad . \tag{2.11}$$

Equating Eqs. 2.10 and 2.11 (Eqs. B11 and B2 from Ref. 48), along with the corresponding equation for $F_{B|A}$, the quadratic equations for collision probabilities ($P_{rot,1}$ and $P_{rot,2}$) are obtained as shown in Eq. 2.5 (equivalently Eq. B13 of Ref. 48).

For the sequential probability selection procedure, the collision probabilities appearing in the upper and lower branches of Fig. 2.1 are not equal. We denote probabilities in the upper branch as $P^U_{rot,1}$, $P^U_{rot,2}$, $P^U_{vib,1}$, and $P^U_{vib,2}$, and probabilities in the lower branch as $P^L_{rot,1}$, $P^L_{rot,2}$, $P^L_{vib,1}$, and $P^L_{vib,2}$. Consider the rotational relaxation of particle $i = 1$ (of type A) through a collision with particle $i = 2$ (of type B). Using Fig. 2.1 as a reference, Eq. 2.6 gives the following probability expressions: $P^U_{rot,1} = F_{rot,1}$, $P^L_{rot,1} = \frac{F_{rot,1}}{1-P^L_{rot,2}}$, $P^L_{rot,2} = F_{rot,2}$, and $P^U_{rot,2} = \frac{F_{rot,2}}{1-P^U_{rot,1}}$. Hence, using the sequential probability selection procedure, the DSMC simulated inelastic collision fraction $F^{DSMC}_{A|B}$ is (by reference to Fig. 2.1 and Eq. 2.11):

$$\begin{aligned} F^{DSMC}_{A|B} =& \frac{1}{2}P^U_{rot,1} + \frac{1}{2}(1 - P^L_{rot,2})P^L_{rot,1} \\ =& \frac{1}{2}F_{rot,1} + \frac{1}{2}(1 - P^L_{rot,2})\frac{F_{rot,1}}{1 - P^L_{rot,2}} \\ =& F_{rot,1} \ . \end{aligned} \tag{2.12}$$

Thus the simulated inelastic collision fraction resulting from the original Haas *et al.* [48] selection procedure and the modified version (the proposed sequential probability selection procedure) are the same, and both are consistent with the Jeans equation using a

specified collision number $Z_{A|B}$.

2.2.2.3 Equivalence of Sequential Probability and Gimelshein *et al.* [44] Selection Procedures

In the selection procedure by Gimelshein *et al.* [44], the calculation of the simulated collision probability for inelastic collisions and the subsequent use of the acceptance-rejection technique are combined together, and it is not immediately evident that the simulated relaxation is equivalent to that produced by the other two selection procedures. In this subsection, we present an analytical proof that this selection procedure is equivalent to the sequential probability selection procedure, and therefore the three selection procedures are equivalent in that they produce the same macroscopic relaxation rate.

We start with the equations shown in Fig. 1 of Ref. 44,

$$A_1 = F_{rot,1} \tag{2.13a}$$

$$A_2 = F_{rot,1} + F_{rot,2} \tag{2.13b}$$

$$A_3 = F_{rot,1} + F_{rot,2} + F_{vib,1} \tag{2.13c}$$

$$A_4 = F_{rot,1} + F_{rot,2} + F_{vib,1} + F_{vib,2} \tag{2.13d}$$

Based on our understanding, the four terms $P_{r,A}, P_{r,B}, P_{v,A}, P_{v,B}$ in Fig. 1 of Ref. 44, should correspond to the four terms $F_{rot,1}, F_{rot,2}, F_{vib,1}, F_{vib,2}$ in the current chapter. Therefore, we have changed the notation of these four terms to the current chapter notation when rewriting the above equations. In the Gimelshein *et al.* procedure [44], a random number R_n between [0, 1] is first selected, and then subsequently used to test each type of relaxation collision according to the acceptance-rejection technique. In these tests, the inequalities of the form $R_n > A_1$ and $A_i < R_n < A_{i+1} (i = 1, 2, 3)$ are sequentially tested. If one inequality is true, then the corresponding inelastic collision relaxation is performed, and the procedure ends for the current collision pair. Only if one inequality does not hold, will the subsequent inequality be tested.

(1) $R_n < A_1$

With the random number $R_n \in [0, 1]$ and the acceptance-rejection technique, we

should have the simulated DSMC collision probability for rotational relaxation of particle 1 (particle A in Ref. 44) as,

$$P_{rot,1}^{sim} = A_1 = F_{rot,1} \tag{2.14}$$

(2) $A_1 < R_n < A_2$

If $A_1 < R_n$, then the random number is $R_n \in [A_1, 1]$. If further $R_n < A_2$, we should have,

$$R_n - A_1 < A_2 - A_1 \ . \tag{2.15}$$

Denote $R_1 = R_n - A_1$, then $R_1 \in [0, 1 - A_1]$, and we have,

$$A_2 - A_1 > R_1 \in [0, 1 - A_1] \ . \tag{2.16}$$

This is equivalent to

$$\frac{A_2 - A_1}{1 - A_1} > \frac{R_1}{1 - A_1} = R_1^* \in [0, 1] \ , \tag{2.17}$$

where we have defined a new variable $R_1^* = \frac{R_1}{1-A_1}$. As a result, the DSMC simulated collision probability for rotational relaxation of particle 2 (particle B in Ref. 44) is,

$$P_{rot,2}^{sim} = \frac{A_2 - A_1}{1 - A_1} = \frac{F_{rot,2}}{1 - F_{rot,1}} \ . \tag{2.18}$$

(3) $A_2 < R_n < A_3$

If $A_2 < R_n$, then we have,

$$R_1^* > P_{rot,2}^{sim} = \frac{A_2 - A_1}{1 - A_1} \ , \tag{2.19}$$

and $R_1^* \in [\frac{A_2 - A_1}{1 - A_1}, 1]$.

If $R_n < A_3$, then it corresponds to,

$$R_1^* < \frac{A_3 - A_1}{1 - A_1} \ . \tag{2.20}$$

Denote $R_2 = R_1^* - \frac{A_2-A_1}{1-A_1}$, then $R_2 \in [0, 1-\frac{A_2-A_1}{1-A_1}]$. With $R_n < A_3$, we then have,

$$R_2 < \frac{A_3-A_1}{1-A_1} - \frac{A_2-A_1}{1-A_1} = \frac{A_3-A_2}{1-A_1} \ , \tag{2.21}$$

i.e.,

$$\frac{A_3-A_2}{1-A_1} > R_2 \in [0, 1-\frac{A_2-A_1}{1-A_1}] \ . \tag{2.22}$$

This is equivalent to

$$\frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} > \frac{R_2}{1-\frac{A_2-A_1}{1-A_1}} = R_2^* \in [0,1] \ , \tag{2.23}$$

where we have defined a new variable $R_2^* = \frac{R_2}{1-\frac{A_2-A_1}{1-A_1}} = \frac{R_2}{1-\frac{F_{rot,2}}{1-F_{rot,1}}}$. As a result, the DSMC simulated collision probability for vibrational relaxation of particle 1 (particle A in Ref. 44) is,

$$P_{vib,1}^{sim} = \frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} = \frac{\frac{F_{vib,1}}{1-F_{rot,1}}}{1-\frac{F_{rot,2}}{1-F_{rot,1}}} \ . \tag{2.24}$$

(4) $A_3 < R_n < A_4$

If $A_3 < R_n$, then we have,

$$R_2^* > P_{vib,1}^{sim} = \frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} \ , \tag{2.25}$$

and $R_2^* \in [P_{vib,1}^{sim}, 1]$.

If $R_n < A_4$, then we should have,

$$R_2^* < \frac{\frac{A_4-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} \ . \tag{2.26}$$

Denote $R_3 = R_2^* - P_{vib,1}^{sim} = R_2^* - \frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}}$, then $R_3 \in [0, 1-P_{vib,1}^{sim}] = [0, 1-\frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}}]$. With $R_n < A_4$, we then have,

$$R_3 < \frac{\frac{A_4-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} - \frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} = \frac{\frac{A_4-A3}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} \ , \tag{2.27}$$

i.e.,

$$\frac{\frac{A_4-A3}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} > R_3 \in [0, 1-P_{vib,1}^{sim}] = [0, 1-\frac{\frac{A_3-A_2}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}}] \ . \tag{2.28}$$

This is equivalent to

$$\frac{\frac{A_4-A3}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} \Bigg/ (1-P_{vib,1}^{sim}) > \frac{R_3}{1-P_{vib,1}^{sim}} = R_3^* \in [0,1] \ , \tag{2.29}$$

where we have defined a new variable $R_3^* = \frac{R_3}{1-P_{vib,1}^{sim}}$. As a result, the DSMC simulated collision probability for vibrational relaxation of particle 2 (particle B in Ref. 44) is,

$$P_{vib,2}^{sim} = \frac{\frac{A_4-A3}{1-A_1}}{1-\frac{A_2-A_1}{1-A_1}} \Bigg/ \left(1-P_{vib,1}^{sim}\right) = \frac{\frac{F_{vib,2}}{1-F_{rot,1}}}{1-\frac{F_{rot,2}}{1-F_{rot,1}}} \Bigg/ \left(1-\frac{\frac{F_{vib,1}}{1-F_{rot,1}}}{1-\frac{F_{rot,2}}{1-F_{rot,1}}}\right) \ . \tag{2.30}$$

By simplification, we have $P_{rot,1}^{sim} = F_{rot,1}$, $P_{rot,2}^{sim} = \frac{F_{rot,2}}{1-F_{rot,1}}$, $P_{vib,1}^{sim} = \frac{F_{vib,1}}{1-F_{rot,1}-F_{rot,2}}$ and $P_{vib,2}^{sim} = \frac{F_{vib,2}}{1-F_{rot,1}-F_{rot,2}-F_{vib,2}}$, which are the same as the collision probabilities given by the sequential probability selection procedure. Therefore, we have shown that the three approaches to select potential inelastic collisions according to the particle selection prohibiting double relaxation procedure are equivalent and will give the same macroscopic relaxation rate.

2.2.2.4 Time cost comparison of the three approaches

In this section, we give a rough estimation of the time cost of the three different approaches for particle selection prohibiting double relaxation. It should be noted, however, that the time cost of selecting appropriate particles to relax contributes a negligible portion of the total DSMC simulation time.

All operations required to calculate $P_{rot,1}$ etc., are floating point. For the purpose of comparison, we prescribe that the time cost of summation/subtraction is 1, the time cost of multiplication/division is 2, the time cost of a square root is 3, the time cost of an IF statement is 4, and the time cost for generating one random number is 5. As an example, assume that the rotational collision numbers are $Z_{rot,1} = 5$, $Z_{rot,2} = 10$, and the vibrational collision numbers are $Z_{vib,1} = 100$, $Z_{vib,2} = 200$. By counting the

total number of operations involved in each selection procedure, we arrive at an approximate total time cost for the three different particle selection prohibiting double relaxation procedures. The results are shown in Table. 2.1, for both the constant collision number and temperature dependent collision number cases. It should be noted, the time costs shown are rough estimations, and only account for the operations associated with the collision probability calculation, generating random numbers for acceptance-rejection techniques, and IF statements used to determine potential inelastic collision types. Again it is stressed that these time costs represent a negligible amount of the total DSMC simulation time cost.

Table 2.1: Time cost of the three approaches.

Approach	Haas *et al.*[48]	Current approach	Gimelshein *et al.*[44]
Constant Z_{rot}, Z_{vib}	1	1	0.47
Variable Z_{rot}, Z_{vib}	1	0.41	0.27

2.2.3 Verification of the sequential probability selection procedure

To test the sequential probability selection procedure, we conduct an isothermal relaxation simulation for a mixture of two species, where the translational temperature of the system is maintained at a constant value of $T_t = 10,000$ K. To maintain the translational temperature of the system at a constant value, each time step, we regenerate the velocities of all particles contained in the simulation domain, following a Maxwell-Boltzmann distribution at $T_t = 10,000K$. The rotational and vibrational energies of the particles are not changed during this process. Since the translational temperature is constant during an isothermal relaxation simulation, the mean collision time is also a constant, and the resulting Jeans equation has an analytical solution.

As outlined in Ref. 48, for a multi-species gas mixture, the internal energy relaxation rate of a species j is determined through summing all inelastic contributions due to collisions with all possible collision partners k in the system,

$$\frac{dE_{rot,j}}{dt} = \sum_k \frac{E^*_{rot,j}(t) - E_{rot,j}(t)}{\tau_{rot,j|k}} = \sum_k \frac{E^*_{rot,j}(t) - E_{rot,j}(t)}{\tau_{c,j|k} Z_{rot,j|k}} \tag{2.31a}$$

$$\frac{dE_{vib,j}}{dt} = \sum_k \frac{E^*_{vib,j}(t) - E_{vib,j}(t)}{\tau_{vib,j|k}} = \sum_k \frac{E^*_{vib,j}(t) - E_{vib,j}(t)}{\tau_{c,j|k} Z_{vib,j|k}} \ . \tag{2.31b}$$

For isothermal relaxations, $E^*_{rot,j}(t) = E_{rot,j}(\infty)$, and $E^*_{vib,j}(t) = E_{vib,j}(\infty)$. If rotational and vibrational relaxation times are assumed to depend only on translational temperature, then $\tau_{rot} = \tau_c Z_{rot}$ and $\tau_{vib} = \tau_c Z_{vib}$ are constant, and Eqs. 2.31a and 2.31b have the following analytical solution:

$$\frac{E_{rot,j}(\infty) - E_{rot,j}(t)}{E_{rot,j}(\infty) - E_{rot,j}(0)} = exp\left(- \sum_k \frac{t}{\tau_{c,j|k} Z_{rot,j|k}} \right) \tag{2.32a}$$

$$\frac{E_{vib,j}(\infty) - E_{vib,j}(t)}{E_{vib,j}(\infty) - E_{vib,j}(0)} = exp\left(- \sum_k \frac{t}{\tau_{c,j|k} Z_{vib,j|k}} \right) \ . \tag{2.32b}$$

Using $E_j = \frac{\zeta_j}{2} k_B T$, the above two equations can be written in terms of temperature, as,

$$\frac{T_{rot,j}(\infty) - T_{rot,j}(t)}{T_{rot,j}(\infty) - T_{rot,j}(0)} = exp\left(- \sum_k \frac{t}{\tau_{c,j|k} Z_{rot,j|k}} \right) \tag{2.33a}$$

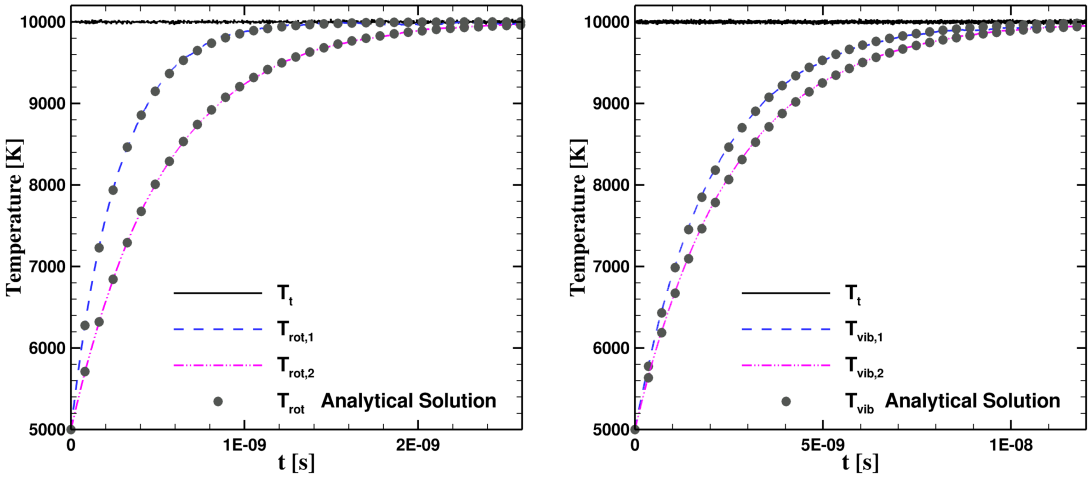
$$\frac{\zeta_{vib,j}(\infty) T_{vib,j}(\infty) - \zeta_{vib,j}(t) T_{vib,j}(t)}{\zeta_{vib,j}(\infty) T_{vib,j}(\infty) - \zeta_{vib,j}(0) T_{vib,j}(0)} = exp\left(- \sum_k \frac{t}{\tau_{c,j|k} Z_{vib,j|k}} \right), \tag{2.33b}$$

where $\zeta_{vib,j}(0)$, $\zeta_{vib,j}(\infty)$ and $\zeta_{vib,j}(t)$ are the effective vibrational degrees of freedom at time 0, ∞, and t, respectively.

The specified rotational and vibrational collision numbers for the two species are listed in Table. 2.2, together with the variable hard sphere (VHS) parameters used in the DSMC simulations. A characteristic vibrational temperature of $\theta_v = 3390K$ is assumed for both species. The two species have mole fractions of 0.3 and 0.7, respectively. The VHS model parameters used for the two species (ω, d_{ref} in Table 2.2) correspond to those of N_2 and O_2, however, the rotational and vibrational collision number Z_{rot} and Z_{vib} do not correspond to the values for those gas species, and are set here for demonstration purpose only . The simulation results are shown in Fig. 2.2 for the relaxation history of the rotational and vibrational temperatures of each species in the mixture. It is evident that the sequential probability selection procedure is able to accurately simulate the specified relaxation rate.

Table 2.2: Simulation parameters specific to each collision pair.

$j\|k$	Z_{rot}	Z_{vib}	$\omega^{j,k}$	$d_{ref}^{j,k}(\times 10^{-10}m)$	$T_{ref}(K)$
1|1	5	40	0.74	4.17	273
1|2	8	60	0.755	4.12	273
2|1	10	80	0.755	4.12	273
2|2	15	60	0.77	4.07	273

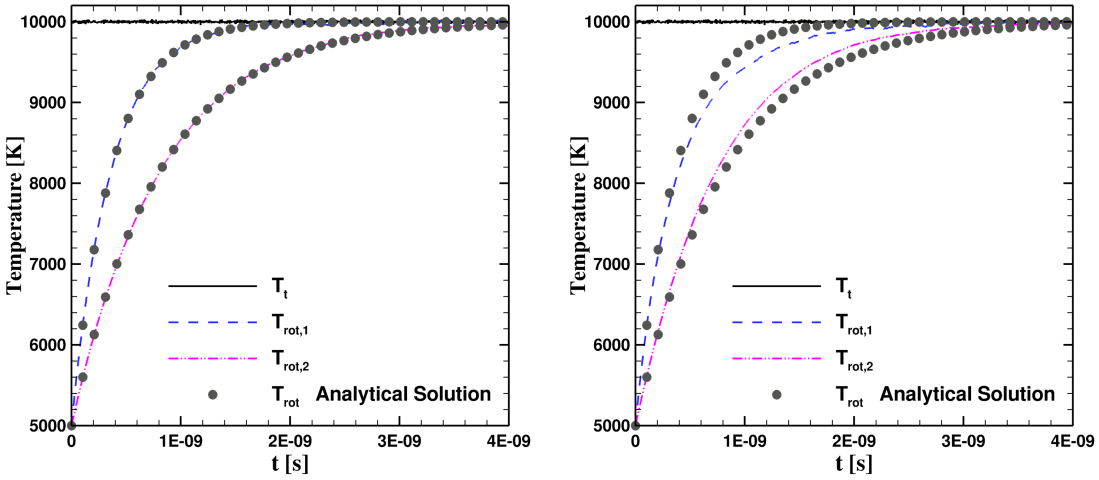


(a) Rotational temperature

(b) Vibrational temperature

Figure 2.2: The rotational and vibrational relaxation temperature in an isothermal reservoir simulation using the sequential selection procedure.

As a further demonstration, a similar isothermal relaxation simulation is conducted using the pair selection procedure (selection procedure (A)). Specifically the form of Eq. 2.4 as given by both Lumpkin *et al.* [67] and Haas *et al.* [48] is used. This simulation considers only rotational relaxation and the rotational collision numbers are modified from Table. 2.2 to be $Z_{rot} = 10$ for collisions 1|2 and 2|1, and to be $Z_{rot} = 20$ for collision 2|2. The results using the sequential probability selection procedure are shown in Fig. 2.3(a) and the results from the pair selection procedure (selection procedure (A)) are shown in Fig. 2.3(b). Clearly, the results using the pair selection procedure do not agree with the analytical solution for the mixture, whereas the sequential probability selection procedure exactly reproduces the analytical solution in the same manner as the original procedure of Haas *et al.* [48].



(a) Sequential selection procedure result

(b) Pair selection procedure result

Figure 2.3: Comparison of two different selection procedures for rotational relaxation in a two species mixture in an isothermal reservoir simulation.

2.3 Chapter Conclusions

A modification to an existing inelastic collision selection procedures is presented in this chapter, which is referred to as the sequential probability selection procedure. Simple expressions for the inelastic collision probabilities used in DSMC simulations are detailed that do not require the solution of a set of quadratic equations for each collision. This modified procedure is analytically proven to be equivalent to both the original procedure of Haas *et al.* [48] and the modified framework of Gimelshein *et al.* [44]. Thus, all three procedures (for particle selection prohibiting double relaxation) simulate the same internal energy relaxation processes in gas mixtures and the ability to accurately simulate prescribed phenomenological relaxation rates for mixtures is demonstrated. Accuracy of the modified selection procedure is verified through comparison with analytical solutions for rotational and vibrational isothermal relaxations.

The modified procedure of Gimelshein *et al.* [44], that achieves specified relaxation rates using a single random number for each collision, has the simplest implementation and is more computationally efficient than the sequential probability selection procedure presented in this chapter. However, the simulated probabilities do not explicitly appear in the algorithm by Gimelshein *et al.* [44], rather they result from inequalities evaluated

using the random number. Whereas the sequential probability selection procedure uses the simulated probabilities directly within the algorithm.

It is noted that the computational cost of any of the three selection procedures is negligible compared to the overall cost of a DSMC simulation. Thus, the purpose of this chapter is to convey a clear understanding of the accuracy implications of the inelastic collision selection process for DSMC simulations of gas mixtures, to prove the equivalence of various selection procedures, to verify their ability to reproduce phenomenological internal energy relaxation rates in gas mixtures, and finally to detail the implementation of these procedures so that no inconsistencies are introduced when probability expressions (collision models) are transferred between DSMC implementations and when DSMC simulations are compared with continuum simulations.

Chapter 3

Phenomenological Models for Flows in Thermochemical Nonequilibrium

3.1 Chapter Introduction

For hypersonic flow occurring at high flight altitude (i.e., at low density), the flow field usually contains regions of strong gradient due to the high flight speed, and the interaction of flow and object inside the flow field. This includes strong shock waves and expansion zones. Such regions are usually associated with different characteristic relaxation time for different energy modes and flow phenomena (translational, rotational, vibrational relaxation, chemical reaction, and radiation phenomena), because of the dramatic temperature changes involved. As a result, the flows are usually in thermochemical nonequilibrium condition, involving finite-rate translational-rotational-vibrational-chemical relaxation. To model such flow phenomena, the DSMC method [15] has been widely used.

As already discussed in Chapter 2, the DSMC method is usually coupled with the phenomenological models for internal energy exchange/relaxation, and chemical reactions [15, 19, 21–26, 33, 34, 47, 91, 95, 100]. Specifically, the internal energy relaxation

process (the rotational, or vibrational energy relaxation) is described by a characteristic time for relaxation, or equivalently a collision number for each relaxation process, as shown in the Jeans or Landau-Teller equation (Eqs. 2.1 and 2.3 of Chapter 2). In the phenomenological model approach, the characteristic relaxation time or the collision number for each relaxation process is then linked to a corresponding collision probability P_i (P_{rot} or P_{vib}) in the DSMC simulation, according to Eq. 2.4 of Chapter 2 [15, 48, 67]. The objective is to match the DSMC simulated relaxation rate in accordance with that specified by the relaxation time or collision number in the Jeans or Landau-Teller equation (Eqs. 2.1 and 2.3 of Chapter 2). P_i is generally a function of the inelastic collision number Z_i, and the specific inelastic collision *selection procedure* used to model the relaxation process [44, 48] as discussed in Eq. 2.4 of Chapter 2. The equation is re-written here as,

$$P_i = f(\zeta_{i,A}, \zeta_{i,B}, \zeta_t, Z_i) \quad , \tag{3.1}$$

where ζ_t is the translational degrees of freedom of the collision pair (for the variable hard sphere (VHS) molecular model with a temperature dependent viscosity exponent of ω, $\zeta_t = 5 - 2\omega$), and $\zeta_{i,A}$, $\zeta_{i,B}$ are the effective internal degrees of freedom of the energy mode participating in the inelastic collision for the colliding partners A and B, respectively. The exact form of $f(\zeta_{r,A}, \zeta_{r,B}, \zeta_t, Z_i)$ for some typical inelastic collision selection procedures have been discussed in Chapter 2.

The same idea is used for modelling finite rate chemical reaction processes in the DSMC method, where a chemical reaction probability P_{react} is used to determine whether each molecular collision will be reactive or not. The chemical reaction probability P_{react} is usually expressed as a function of the collision energy, and the overall objective is to link such probability to the Arrhenius type chemical reaction rate [15, 47]. In other words, with the properly specified P_{react}, the temperature dependent Arrhenius reaction rate are simulated under equilibrium conditions in the DSMC method.

Since the collision number Z_i (Z appearing in Eq. 2.3 of Chapter 2) is usually temperature dependent, P_i obtained from Eq. 3.1 also has a temperature dependence. Such temperature dependence should be maintained in the DSMC method. There are usually three types of approaches to achieve the overall temperature dependence of P_i. In the first two approaches, P_i is directly calculated using Eq. 3.1, where the temperature

dependence is achieved by Z_i. In the first approach, Z_i is calculated using a macroscopic temperature, which is either the cell-averaged temperature (averaged over all the particles in one collision cell or sampling cell) or the collision-averaged temperature (averaged over all the collision pairs in one cell). In the second approach, Z_i is calculated using an effective collision temperature (discussed later in Eq. 3.9), which is obtained from dividing the available energy participating in the collision with the corresponding degrees of freedom [15, 16]. In the third approach, a collision quantity dependent p_i is used (it is denoted as p_i to distinguish from P_i), where p_i usually depends on the relative translational energy and the internal energy of the collision pair. Upon integration of p_i over the corresponding equilibrium distribution function (the relative translational energy and internal energy distribution function), the temperature dependent collision probability P_i, described by Eq. 3.1, will be achieved [2, 23, 24, 88] (this will be discussed in more detail in Chapter 4 and Appendix A). In the first two approaches, because the inelastic collision probability P_i is related to some macroscopic temperature through Eq. 3.1, it can be directly related to Eq. 2.1 – 2.3. The implementation should be such that equipartition of energy between different energy modes is achieved at equilibrium, and detailed balance is satisfied [16, 33]. Particularly for the third approach, the equipartition of energy between different energy modes at equilibrium may not be achieved without proper treatment of the bias introduced by the collision quantity dependent p_i [1, 88]. This has been shown previously for both the rotational and vibrational relaxation processes [1, 2, 22, 34, 88], and will be discussed later in this chapter and in more detail in Chapter 4.

The post collision energies of the colliding particles are usually determined by the Borgnakke-Larsen (BL) model [19]. In this model, the post collision energies of the colliding particles are sampled from an equilibrium distribution corresponding to the available energy modes participating in the inelastic collision. The original BL model assumes a continuous internal energy distribution for both rotation and vibration energy. The continuous internal energy assumption, works well for rotational mode, as rotation energy is usually fully excited even at relatively low temperature, and can be treated continuously. Vibration energy, however, is not fully excited until at very high temperature [15]. As a result, the vibration energy is usually considered as discrete energy levels, and a modified BL model is used for post collision vibrational energy

redistribution [13]. The phenomenological relaxation models are used to set the correct relaxation rate for rotational or vibrational relaxation, meanwhile, the BL model ensures that the desired equilibrium states are reached. The combined use of these models achieve the correct relaxation rate and energy equipartition at equilibrium.

When the flow temperature increases, chemical reactions will occur, and molecules will start to dissociate. To simulate such chemically reacting flows, several phenomenological models have been proposed. Among these models are Bird's total collision energy (TCE) model [15], the vibrational favored dissociation (VFD) model by Haas et al. [47], and the recently proposed quantum-kinetic (QK) chemical reaction model by Bird [17]. The VFD model was developed specifically to model dissociation reactions accounting for the coupled vibration-dissociation behavior at high temperatures [47, 83]. The TCE model and the QK model can be used for general chemical reaction simulations including exchange reactions and recombination reactions. The idea behind all of these models are similar to those for rotation and vibration relaxation, and the post reaction energies of each species are usually determined using the BL model. The overall objective of these models is to maintain the correct macroscopic reaction rate as a function of temperature. Such phenomenological models depend on the availability of existing experimental, theoretical, and computational data on reaction rates, which are usually expressed as a function of temperature.

In this chapter, we first discuss the existing phenomenological models for rotational, vibrational relaxation and chemical reaction processes in the DSMC method. The models selected in this chapter are then implemented in a consistent and modular fashion, in a three dimensional Cartesian grid based DSMC code [43], using the sequential probability collision selection procedure outlined in Chapter 2. Model implementations are validated using problems with analytical solutions, and using simple zero dimensional adiabatic and isothermal reservoir simulations. As representative cases, we further conduct a detailed comparison of existing phenomenological models for vibrational relaxation and dissociation reactions, using both the above mentioned simulations and one dimensional stagnation line simulations of hypersonic flow. The comparisons of existing rotational relaxation model will be given in Chapter 4, together with the discussion of a new rotational relaxation model named NDD model. The phenomenological models discussed in this chapter are generally based on different procedures and ideas, and

contain different types of model parameters that are determined either through experimental results or through physical arguments. Because of this, the comparisons should be conducted carefully to avoid any misleading conclusions that may be due to improper selection of the parameters used in these models.

Specifically, section 3.2 discuss the existing phenomenological rotational relaxation models commonly used in the DSMC method. Section 3.3 discuss the implementation and consistent comparison of existing phenomenological vibrational relaxation models in the DSMC method. Section 3.4 discuss the existing chemical reaction models in the DSMC method, and their implementation for dissociation reactions (TCE, VFD, and QK models) [15, 17, 47]. Section 3.5 discuss the consistent comparisons of the various phenomenological vibrational relaxation and dissociation models for one dimensional stagnation line simulations [15] of hypersonic flow. The conclusions of this chapter are then drawn in section 3.6.

3.2 Phenomenological Models for Rotational Relaxation

As described by the Jeans equation (Eqs. 2.1 and 2.3) and discussed in Chapter 2, the rotational relaxation in the continuum simulation, is usually modeled by the rotational collision number Z_{rot}, which is defined as

$$Z_{rot} = \frac{\tau_{rot}}{\tau_c} \tag{3.2}$$

where τ_{rot}, τ_c are the rotational relaxation time, and the molecular mean collision time, respectively. Depending on the gas species and the temperature range under consideration, Z_{rot} may be considered as a constant, or assumed to be temperature dependent [15].

For the constant Z_{rot} case, the rotational inelastic collision probability in the DSMC simulation can be obtained according to Eq. 3.1 directly, with $Z_i = Z_{rot} =$ constant, and the inelastic collision selection procedure used.

Usually, the rotational collision number Z_{rot} is temperature dependent. Z_{rot} can be correlated with existing experimental results, expressed as a general temperature dependent function. Alternatively, the Parker temperature dependent rotational collision number model could be used, which has an analytical expression for Z_{rot} derived from theoretical analysis, and may be further parametrized with existing experimental

results. Generally, in the Parker model, Z_{rot} has the form,

$$Z_{rot}(T) = \frac{Z_r^{\infty}}{1 + a(T^*/T) + b(T^*/T)^{\frac{1}{2}}} \ , \tag{3.3}$$

where Z_r^{∞}, a, b, T^* are model constants, and T is the translational temperature [21, 66]. Specifically for N_2–N_2 collision, $Z_r^{\infty} = 23.5, a = \pi(1 + \pi/4), b = \pi^{3/2}/2, T^* = 91.5$ K. In this case, as discussed in section 3.1 (the chapter introduction), a cell-averaged temperature may be defined for each DSMC collision cell or sampling cell. Then for each DSMC collision/sampling cell, a Z_{rot} may then be calculated using Eq. 3.3 for example. The calculation of rotational inelastic collision probability is then the same as the constant Z_{rot} situation.

Alternatively for the temperature dependent Z_{rot} case, a collision energy dependent probability $p_{rot}(\varepsilon_t, \varepsilon_r)$ may be used. Various rotational models of this type have been developed [21, 22, 25, 26, 33, 95]. The derivation of such models, and the connection between p_{rot} and Z_{rot} are discussed in detail in Chapter 4 and Appendix A. The post collision energy redistribution using the BL model is also discussed in Chapter 4 and Appendix A.

3.3 Phenomenological Models for Vibrational Relaxation

3.3.1 Existing vibrational relaxation models

Similar as rotational collision number Z_{rot}, the vibrational collision number Z_{vib}, is defined as,

$$Z_{vib} = \frac{\tau_{vib}}{\tau_c} \tag{3.4}$$

where τ_{vib}, τ_c are the vibrational relaxation time, and the molecular mean collision time, respectively. τ_{vib} is usually a function of temperature, and according to the experimental correlation by Millikan and White [69, 88], it usually has the following form,

$$p\tau_{vib} = \exp(\frac{A}{T^{1/3}} + B) \tag{3.5}$$

where p is pressure, T is the translational temperature, and A, B are constants related to molecular properties. As a result, Z_{vib} has a temperature dependence. At temperatures

higher than 8,000 K (beyond the range of Millikan and White correlation for Z_{vib}), Park's high temperature correction is usually added [88]. The inelastic collision probability P_{vib} is then related to Z_{vib} based on Eq. 3.1, conditional on the inelastic collision selection procedure and the vibrational energy description used.

To account for the temperature dependence of P_{vib} in DSMC simulations, three types of relaxation models are used. One approach uses the cell-averaged temperature to calculate P_{vib}, similarly as that described for rotational relaxation. In the approach proposed by Boyd [13, 23, 24, 88], a relative collision velocity dependent inelastic collision probability $p_{vib}(g)$ is used, where $p_{vib}(g)$ has the form,

$$p_{vib}(g) = \frac{1}{Z_0} g^{\alpha} \exp(-\frac{g^*}{g}) \tag{3.6}$$

Upon integration of $p_{vib}(g)$ over the equilibrium relative velocity distribution function $f(g)$,

$$P_{vib} = \int_0^{\infty} p_{vib}(g) f(g) dg \quad , \tag{3.7}$$

the macroscopic temperature dependent P_{vib} is obtained, and the constants in Eq. 3.6 are determined according to Eq. 3.7 and the macroscopic P_{vib} expression.

Another type of model proposed by Bird [15, 16], uses the collision energy to set the vibrational collision probability for each colliding pair. In this approach, the vibrational energy is considered as discrete [13], and Z_{vib} is expressed as,

$$Z_{vib} = \left(\frac{\theta_d}{T}\right)^{\omega} \left[Z_{REF} \left(\frac{\theta_d}{T_{REF}}\right)^{-\omega} \right]^{\left[\left(\frac{\theta_d}{T}\right)^{1/3} - 1\right] / \left[\left(\frac{\theta_d}{T_{REF}}\right)^{1/3} - 1\right]} \quad , \tag{3.8}$$

where θ_d is the characteristic temperature for dissociation, Z_{REF} is the vibrational collision number at the reference temperature T_{REF}, and ω is the temperature dependence of cross-section in the variable hard sphere (VHS) collision model. Such a Z_{vib} approximates the Millikan and White correlation and is also corrected for high temperatures. For a collision pair with translational energy E_t and vibrational energy E_v, the collision energy is $E_c = E_t + E_v$. An effective collision temperature T_c is calculated as,

$$T_c = \frac{2E_c}{k_B \zeta_c} \tag{3.9}$$

where ζ_c represents the degrees of freedom of the energy modes participating in the inelastic collision, which usually includes both the vibrational and the translational energy modes. Then the vibrational collision probability for the collision pair is calculated according to Eq. 3.8 and Eq. 3.1.

For all the different vibrational models discussed here, to ensure equipartition of energy at equilibrium condition, the post collision energies of each collision pair selected for inelastic collision, are then sampled using the general BL model [15, 19]. Depending on the description of the vibrational energy mode, the details of the energy redistribution processes are different. For continuous vibrational energy description, the BL model process is the same as that for rotational relaxation model. While for vibrational mode using the discrete quantized vibrational energy description, the BL model procedure is discussed in Ref. [13].

3.3.2 Comparison of existing vibrational relaxation models

In this section, we test the following four vibrational relaxation models using an adiabatic reservoir simulation. The four models are,

1. Bird's expression of Z_{vib} based on the effective collision temperature.
2. A model using Bird's effective collision temperature, however, with a Z_{vib} based on the Millikan-White correlation with Park's high temperature correction.
3. Boyd's velocity dependent collision probability model, where the probability is averaged over all collision pairs within a cell (cell-averaged). The high temperature correction proposed by Boyd is also used in this model.
4. Boyd's velocity dependent collision probability model, where p_{vib} is directly calculated using the collision relative velocity. The high temperature correction proposed by Boyd is also used in this model.

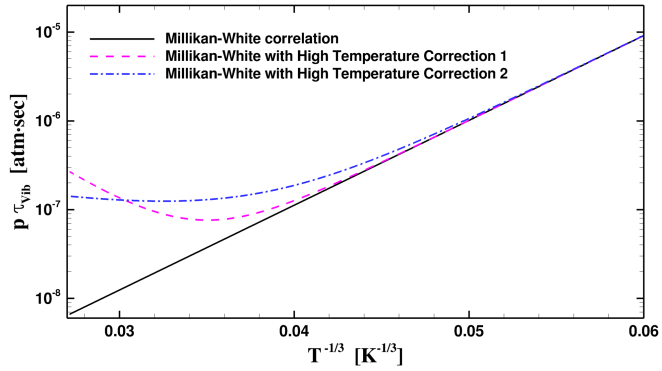
The different expressions for vibrational collision number are shown in Fig. 3.1. The high temperature correction 1 and 2 refers to the two different expressions used to account for vibrational collision number and characteristic relaxation time at high temperature. 1 refers to the original expression by Park [73], and 2 refers to the expression proposed to use in DSMC simulation by Haas et al. [47] and Boyd [27]. At

temperatures below around 8,000K, all expressions for the vibrational collision number agree with each other except Eq. 3.8. It should be noted, however, with a different choice of reference temperature T_{REF} and its corresponding collision number Z_{REF}, this expression by Bird can give good agreement with the Millikan-White correlation and Park high temperature correction in a wide temperature range. The collision velocity based vibrational inelastic collision probability in model 4 is also integrated numerically in DSMC simulation to get the corresponding macroscopic vibrational collision number. As can be seen from Fig. 3.1(c), overall, it agrees well with the Millikan-White correlation over the whole temperature range.

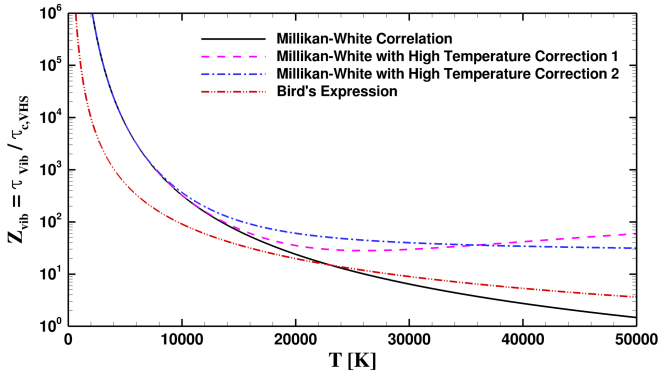
When the velocity based probably p_{vib} of Eq. 3.6 is used directly in DSMC simulation, due to the bias introduced by the selection of collision pairs, equipartition of energy is not satisfied at equilibrium, as can be seen from Fig. 3.2(a), where an adiabatic reservoir simulation of vibrational relaxation is conducted. While using the correction proposed by Abe [1], equipartition is satisfied and the correct equilibrium can be established as seen in Fig. 3.2(a) (this phenomena is more generally discussed in Chapter 4). To further test the velocity based vibrational relaxation model (model 4) with the correction from Abe, an isothermal relaxation simulation is conducted, and the results are shown in Fig. 3.2(b). As can be seen in the figure, the correct relaxation rate is reached when compared to an analytical solution.

In order to further compare model 3 against model 4, the same isothermal relaxation simulation is conducted. Although not shown, model 3 (which uses a cell-averaged collision velocity based probability) exactly reproduces the analytical solution in Fig. 3.2(b) and thus gives a slightly different relaxation result than model 4. To further investigate this difference the vibrational energy distribution functions are computed. In Fig. 3.3(a) the vibrational energy distribution functions computed using model 3 are compared to Boltzmann distributions at various times, showing highly nonequilibrium behavior at early times. As seen in Fig. 3.3(b), the vibrational distribution functions for both model 3 and model 4 are virtually identical, for this isothermal relaxation process.

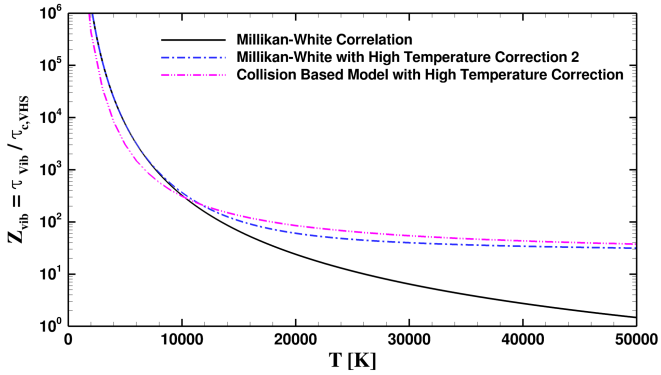
We further test the four vibrational relaxation models using an adiabatic reservoir simulation. The results for the four models are shown in Fig. 3.4. As can be seen from the figure, when using Eq. 3.8 (model 1) to calculate the vibrational collision number during the DSMC simulation, the vibrational relaxation rate is faster compared to the



(a) The vibrational characteristic relaxation time as a function of temperature

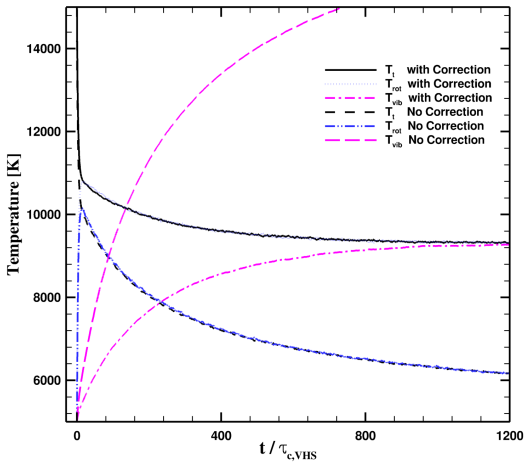


(b) Temperature dependence of the vibrational collision number $Z_{vib} = \tau_{vib}/\tau_{c,VHS}$

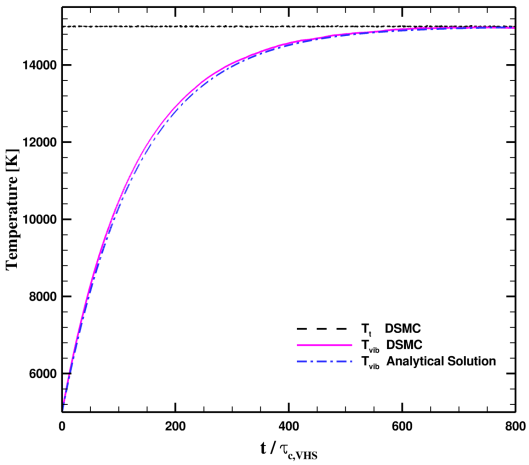


(c) Temperature dependence of the vibrational collision number $Z_{vib} = \tau_{vib}/\tau_{c,VHS}$, collision based probability

Figure 3.1: The temperature dependence of the characteristic vibrational relaxation time τ_{vib} and vibrational inelastic collision number Z_{vib}.



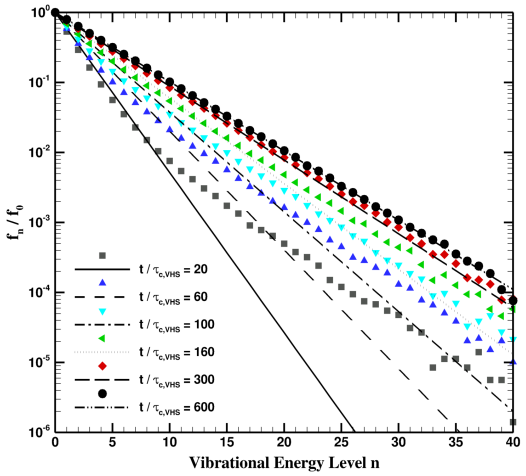
(a) Vibrational relaxation in an adiabatic reservoir using the collision based vibrational collision probability (model 4), with and without correction of bias introduced by the vibrational collision selection rule.



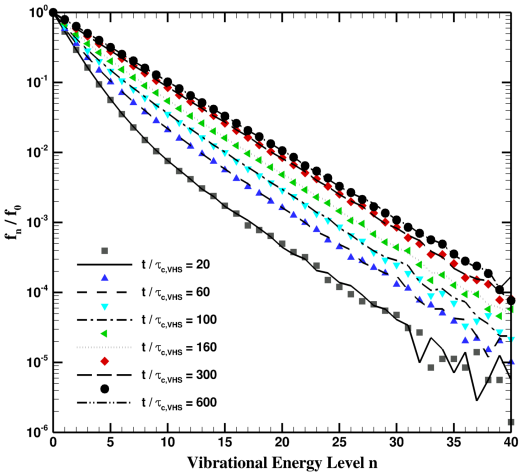
(b) Vibrational relaxation in an isothermal reservoir using the collision based vibrational collision probability (model 4).

Figure 3.2: The adiabatic and isothermal relaxation of the vibrational mode using Boyd's collision based vibrational relaxation model.

Millikan-White result. When the Millikan-White expression is used with Bird's effective collision temperature (model 2), the vibrational temperature profiles agrees well with model 3. Although model 3 and model 4 have the same relaxation rate for isothermal relaxation simulations, as can be seen from Fig. 3.2(b), there is noticeable difference in the vibrational temperature profile between these two models for the adiabatic relaxation. To further investigate this difference, the vibrational energy distributions for model 3 and model 4 are compared for the adiabatic relaxation. Fig. 3.4(c) shows the highly nonequilibrium energy distributions by plotting the results of model 3 against corresponding Boltzmann energy distributions. In Fig. 3.4(b), the vibrational energy distributions resulting from model 3 and model 4 are overlaid at specific times during the relaxation, where some discrepancies are evident. However, when the distributions are compared at the same vibrational temperature (not the same time), the results produced by model 3 and model 4 are in very close agreement. This suggests that for adiabatic relaxation, the same *structure* of the vibrational energy distribution functions result from both model 3 and model 4, however, the overall relaxation rate itself is



(a) The vibrational energy distribution at different times when using model 3. The solid lines are Boltzmann distributions at the corresponding vibrational temperature.



(b) Comparison of vibrational energy distribution functions for model 3 (symbols) and model 4 (solid lines).

Figure 3.3: Comparison of the vibrational energy distribution function using the velocity dependent vibrational collision probability averaged over the cell, and the direct velocity dependent probability

slightly different. This could be attributed to the fact that both methods employ the BL model for post-collision energy redistribution.

3.4 Phenomenological Models for Chemical Reactions

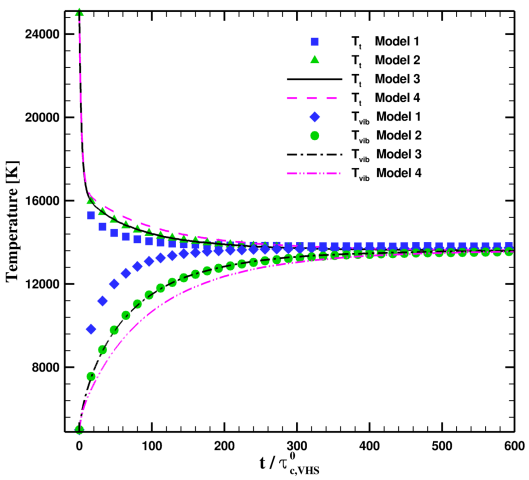
3.4.1 Existing models for dissociation reactions in DSMC

The models we compare for dissociation reactions are the following three models:

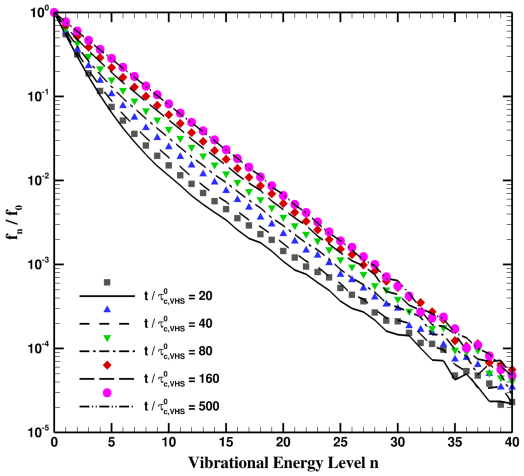
1. The TCE model of Bird [15].
2. The vibrational favored dissociation (VFD) model by Haas and Boyd [47].
3. The Q-K quantum chemical reaction model proposed recently by Bird [17].

The dissociation reaction rate constant k_f is usually expressed in the Arrhenius form, as

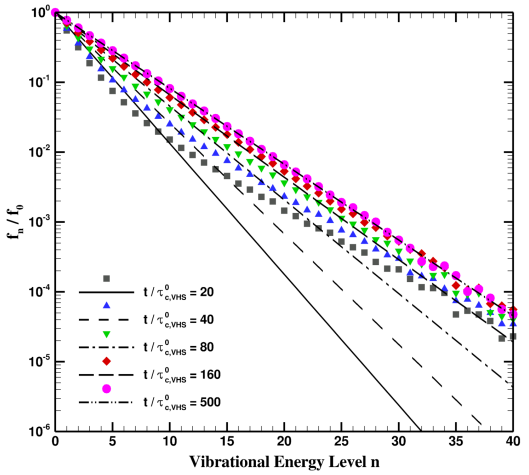
$$k_f = AT^{\eta} \exp(-\frac{E_d}{k_b T}) \tag{3.10}$$



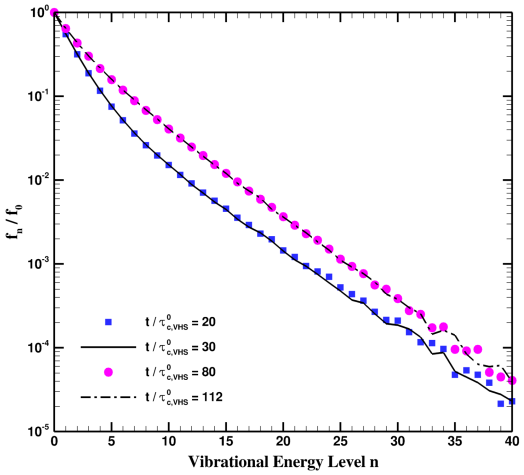
(a) Temperature history of the adiabatic reservoir simulation for four different models.



(b) Vibrational energy distribution functions for model 3 (symbols) and model 4 (lines) at different times.



(c) Vibrational energy distribution functions for model 3 (symbols) compared to Boltzmann distributions (lines) at different times.



(d) Vibrational energy distribution functions for model 3 (symbols) and model 4 (lines) at times corresponding to the same vibrational temperature T_{vib}.

Figure 3.4: The temperature history and the vibrational energy distribution function of the adiabatic reservoir simulation for vibrational excitation.

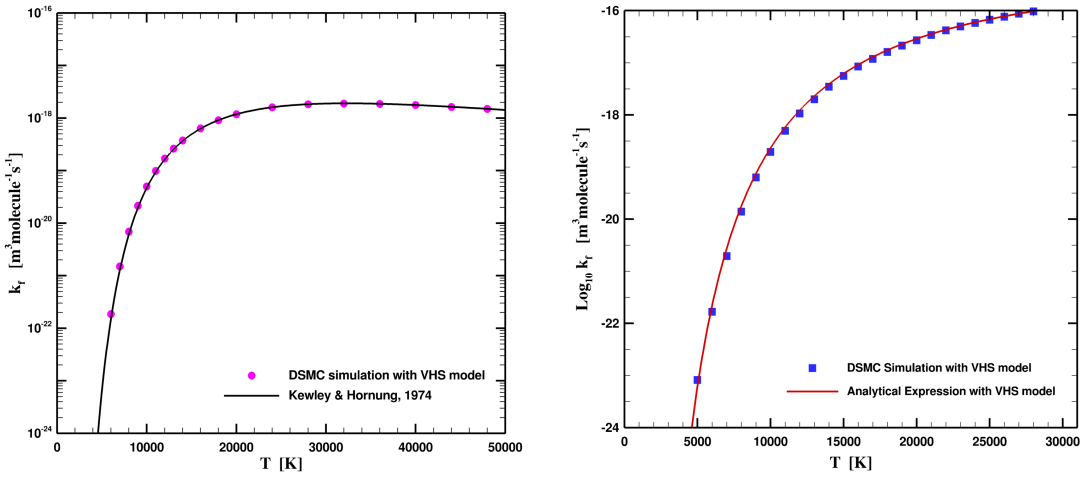
where A, η are some constants, and E_d is the dissociation energy.

In DSMC, the dissociation reaction probability is generally expressed as a function of the collision energy. Upon integration of the dissociation probability over the corresponding equilibrium collision energy distribution function, the macroscopic equilibrium dissociation reaction rate constant k_f as specified by Eq. 3.10 is obtained. For dissociation reaction, the dissociation probability is expressed as Eq. 3.11 and Eq. 3.12 for TCE and VFD model, respectively.

$$P_{diss,TCE} = A_1(E_c - E_d)^{A_2}(1 - E_d/E_c)^{A_3} \tag{3.11}$$

$$P_{diss,VFD} = C_1 \frac{(E_c - E_d)^{C_2}}{(E_c - E_{vib})^{C_3}} E_{vib}^{\phi} \tag{3.12}$$

where $A_1, A_2, A_3, C_1, C_2, C_3$ are some constants related to the species collision parameters and the dissociation reaction rate constant k_f. ϕ is the vibrational favoring parameters used in VFD model.



(a) Comparison of the dissociation reaction rate constant k_f for $N_2 + N_2 \xrightarrow{k_f} N + N + N_2$ between the specified values and results obtained from DSMC simulation using TCE model.

(b) Comparison of the dissociation reaction rate constant k_f for $N_2 + N_2 \xrightarrow{k_f} N + N + N_2$ between the analytical values of QK model and the results obtained from DSMC simulation using Q-K model.

Figure 3.5: Validation of the DSMC simulated dissociation rate constant k_f for TCE model and QK model

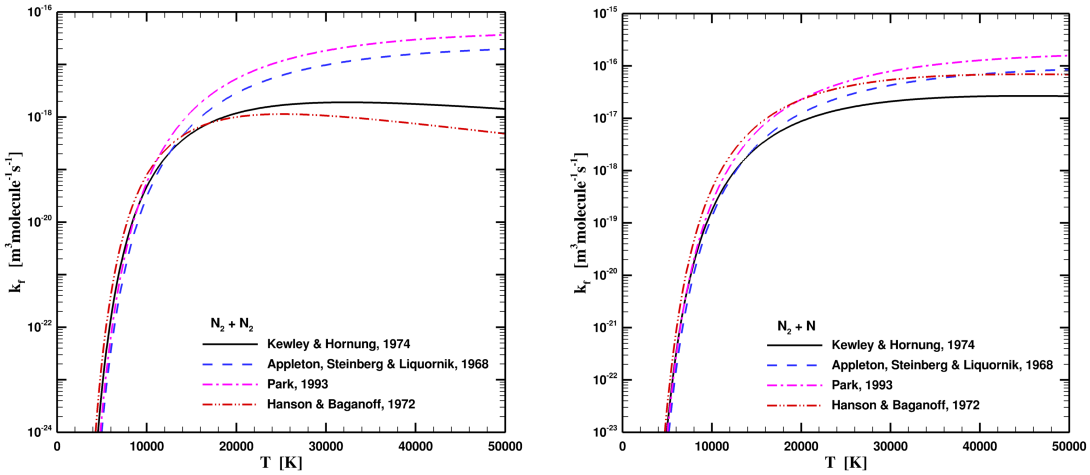
In order to validate the implementations of these dissociation models, the reaction rate simulated by DSMC is compared with the corresponding analytical expression and

the specified reaction rate constant. Results are shown for the TCE model in Fig. 3.5(a) and for the QK model in Fig. 3.5(b). Since the VFD model with a vibrational favoring parameter of $\phi = 0$ is the same as the TCE model, the results for the VFD model are not shown. From the figures we can see that the reaction rate constant achieved in DSMC simulation agrees well with the specified k_f for the TCE model, and the DSMC result agrees well with the analytical expression [17, 40, 41] for the QK model.

Different experimental results exist regarding the N_2 dissociation rate constant at different temperatures. To get an overall feeling of the existing dissociation rate constant expressions, representative data are plotted in Fig. 3.6 for the following two dissociation reactions using existing experimental results. [11, 12, 57, 73]

$$N_2 + N_2 \xrightarrow{k_f} N + N + N_2 \tag{3.13a}$$

$$N_2 + N \xrightarrow{k_f} N + N + N \tag{3.13b}$$



(a) Dissociation rate coefficient for $N_2 + N_2 \xrightarrow{k_f} N + N + N_2$

(b) Dissociation rate coefficient for $N_2 + N \xrightarrow{k_f} N + N + N$

Figure 3.6: Comparison of existing experimental dissociation rate constant for N_2.

As can be seen from Fig. 3.6, usually there is an order of magnitude difference in different expressions at high temperature, where the dissociation rate constant is extrapolated. In this chapter, we choose the dissociation rate expression by Kewley and

Hornung [57] for stagnation line simulations using both TCE and VFD models. While for the QK model, the dissociation rate constant is self specified based on the DSMC simulation parameters. For the VFD model, we choose a vibrational favoring parameter of $\phi = 3$, as recommended by Haas and Boyd [47].

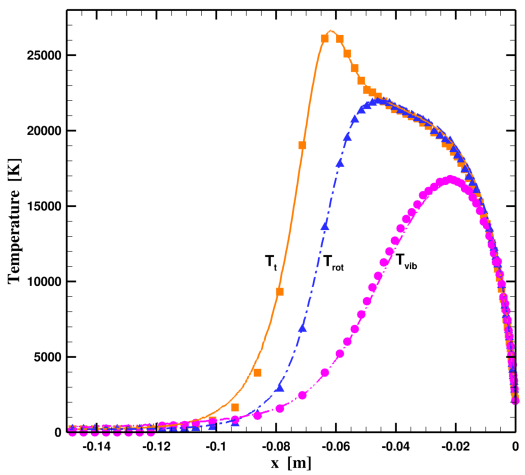
3.5 One Dimensional Stagnation Line Simulations

3.5.1 Stagnation line simulation validation

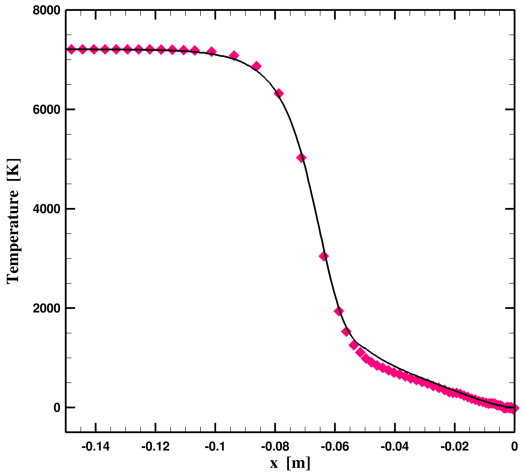
To assess the accuracy of the one dimensional stagnation line simulation technique proposed by Bird [15], results from our implementation of this 1D algorithm are compared with results for 2D flow over a vertical flat plate under the same simulation conditions. The result are shown in Fig. 3.7. The flow has a Mach number of $Ma_\infty = 25$, free stream temperature $T_\infty = 200K$, free stream density $\rho_\infty = 1.60 \times 10^{-5} kg/m^3$, and a wall temperature of $T_w = 1500K$ (this flow conditions are used for all the simulations in this section). The VHS model parameters are given in Table 3.1. We use a constant rotational collision number of 5 (defined according to VHS model mean collision time τ_c) [87]. The results shown for the 2D flow are obtained using cells next to the symmetry plane, and the results are along a straight line. For the 1D stagnation line simulation, particles are removed near the wall regions after the shock wave and boundary layer have been established. Special selection rules are employed for particles selected to be removed near the wall region, to satisfy the conservation of mass, momentum and energy. As can be seen from Fig. 3.7, the 1D stagnation line results are in close agreement with those of the 2D vertical flat plate results. Thus the use of 1D stagnation line simulations for model-model comparison between the different relaxation and dissociation models is appropriate.

Table 3.1: The VHS model parameters

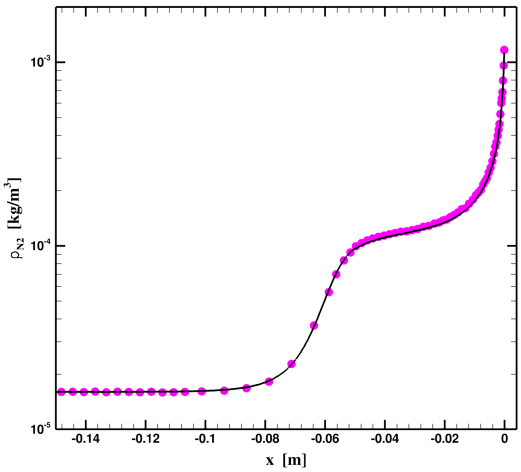
Collision pair	ω	$T_{ref}(K)$	$d_{ref}(m)$
$N_2 - N_2$	0.72	273	4.17×10^{-10}
$N - N$	0.80	273	3.0×10^{-10}
$N_2 - N$	0.76	273	3.585×10^{-10}



(a) Temperature along the stagnation line.



(b) Velocity along the stagnation line.



(c) Density along the stagnation line.

Figure 3.7: Comparison of results between the stagnation stream line for flow over a vertical flat plate, and the one dimensional stagnation line. The symbols are the vertical flat plate result, and the lines are the one dimensional stagnation line results.

3.5.2 Comparison of existing dissociation models in DSMC

Simulations are first conducted to compare only the vibrational relaxation models for a stagnation line flow. Thus, for these simulations, dissociation reactions are not considered. Since model 1 for vibrational relaxation gives a very fast relaxation rate, it is not included in the stagnation line comparison. The results for the remaining three

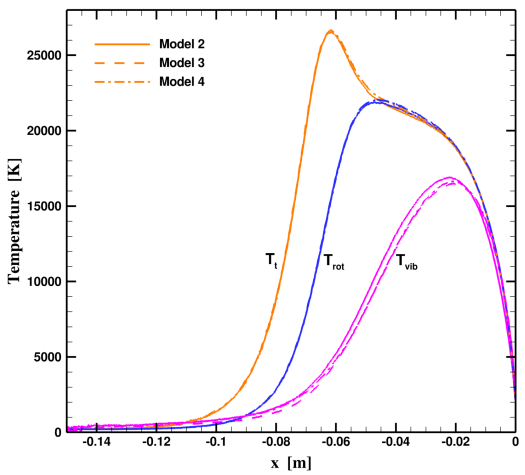
models are shown in Fig. 3.8, Fig. 3.10(a) and Fig. 3.10(c). Since the Millikan-White correlations are used for all three models, it is found that they give similar results for temperature, velocity and density profiles. Model 2 predicts a slightly faster vibrational relaxation rate. This is because that the effective temperature, T_c, is computed for collision pairs, and it is biased toward a higher temperature. This results in a lower vibrational collision number, and hence a faster vibrational relaxation rate. The stagnation point pressure and heat flux is also shown in Table 3.2. The three models give near identical results for both the stagnation point pressure and heat flux.

Simulations are next conducted to compare the three dissociation models. For a consistent comparison, the simulations are all conducted using vibrational relaxation model 4. The results are shown in Fig. 3.9, Fig. 3.10(b) and Fig. 3.10(d). Clearly, the QK model predicts the lowest vibrational temperature. This is consistent with the fact that the dissociation rate constant produced by the QK model is one order of magnitude higher than the experimental results used in this paper in the current temperature range (see Fig. 3.5(b) and Fig. 3.6(a)).

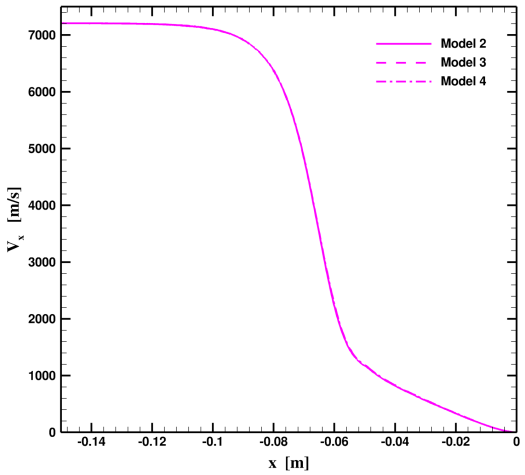
Both TCE and VFD models use the same dissociation rate constant from Kewley and Hornung [57]. In the shock layer, the vibrational temperature is lower than the translational temperature. Due to the vibrational favoring effect, the dissociation cross-section predicted by the VFD model is generally lower than that predicted by the TCE model (for the same collision pair) where the dissociation cross-section is based on the total energy of the colliding pair. As a result, the TCE model gives a slightly higher dissociation rate and a larger N mole fraction as can be seen from Fig. 3.9(c), 3.9(d).

Furthermore, the vibrational energy distribution function (EDF) for the different models are compared in Fig. 3.10, corresponding to the same location. Generally, different vibrational models, and dissociation models result in different vibrational EDF.

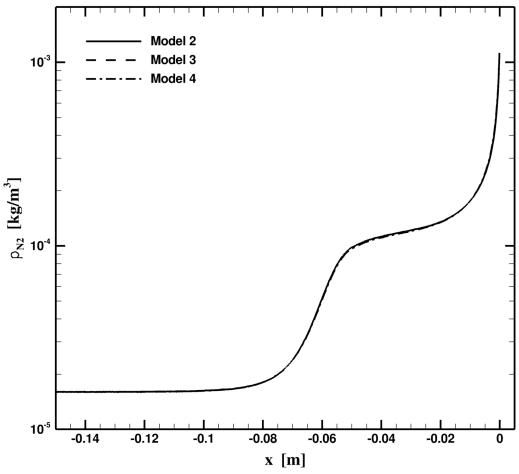
As expected, the stagnation point heat flux is lower for the cases where dissociation reactions are considered compared to the cases where no dissociation reactions are considered. This can be seen from Table. 3.2. The stagnation point pressure is virtually unaffected by the vibration and dissociation models investigated. When dissociation is considered, the stagnation point heat flux is only mildly affected by the models investigated, resulting in a maximum difference of roughly 15%. It is noted that various combinations of rotational, vibrational, and dissociation models may result in larger



(a) Temperature along the stagnation line.



(b) Velocity along the stagnation line.

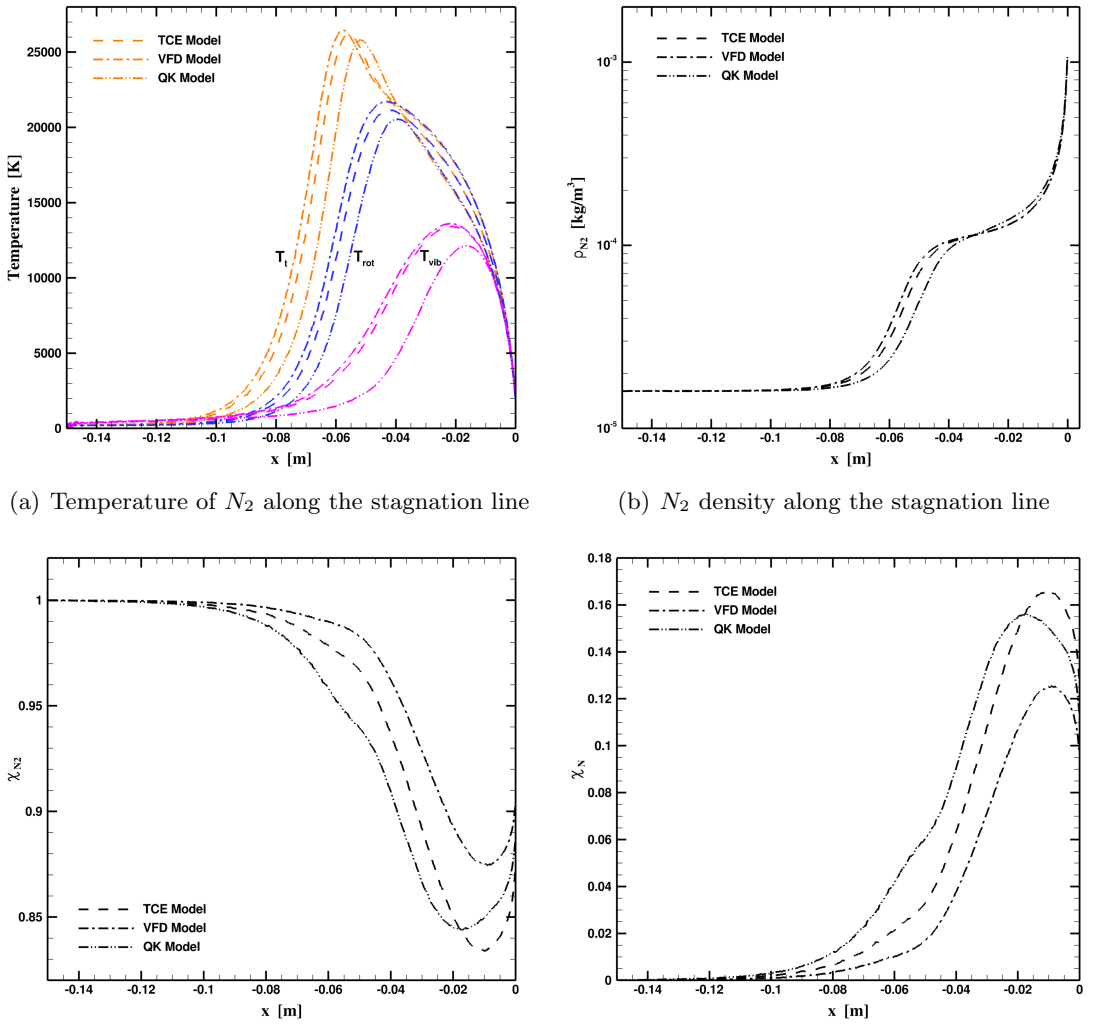


(c) Density along the stagnation line.

Figure 3.8: Comparison of stagnation line flow using different vibrational relaxation model

discrepancies. The models we compared are essentially constructed to reproduce the same experimentally determined rate data and all use the BL model for post-collision energy redistribution, whereas new state-to-state models have been proposed more recently [3, 27, 31, 37, 38, 55, 68, 77, 78]. Furthermore, comparisons for dissociated air mixtures may increase model discrepancies. Finally, if surface chemical reactions driven by atomic species impacting the wall are important, then the total heat flux including

surface chemistry may be more significantly affected by the models examined in this chapter.



(a) Temperature of N_2 along the stagnation line

(b) N_2 density along the stagnation line

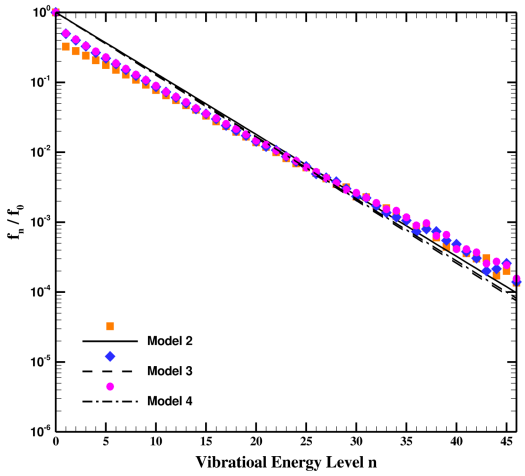
(c) Mole fraction of N_2 along the stagnation line.

(d) Mole fraction of N along the stagnation line.

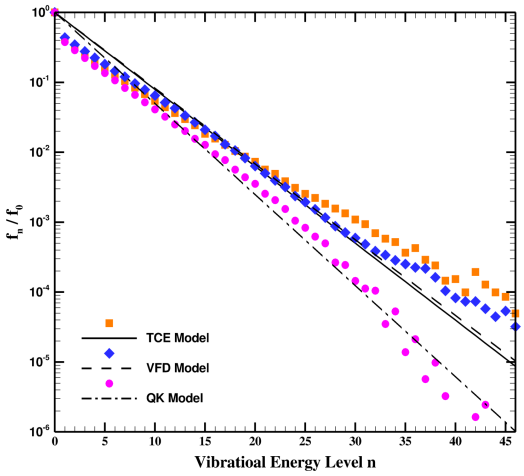
Figure 3.9: Comparison of stagnation line flow using different dissociation models.

3.6 Chapter Conclusions

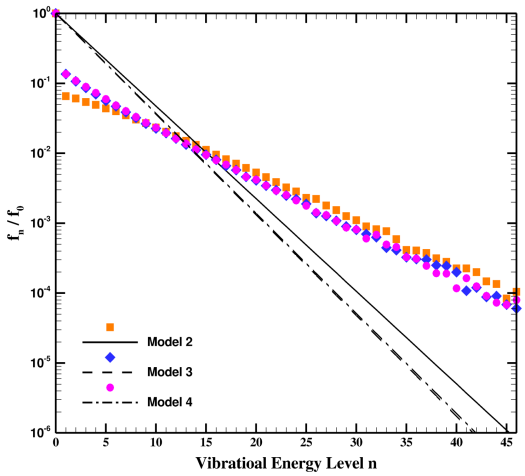
In this chapter, existing phenomenological models for modelling rotational, vibrational relaxation, and chemical reaction processes in the DSMC method are first detailed.



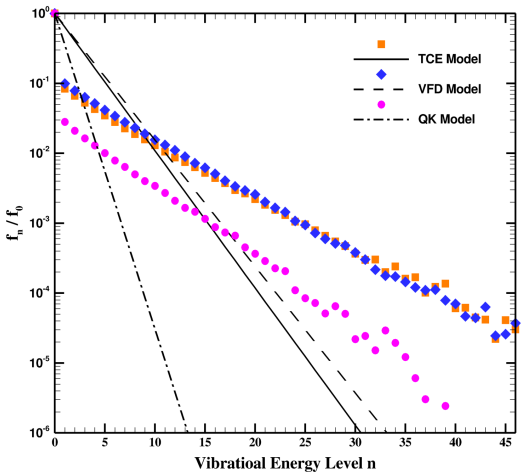
(a) The vibrational energy distribution function at point $x = -0.0225m$ for different vibrational models without considering dissociation reaction.



(b) The vibrational energy distribution function at point $x = -0.0225m$ for different dissociation models.



(c) The vibrational energy distribution function at point $x = -0.045m$ for different vibrational models without considering dissociation reaction.



(d) The vibrational energy distribution function at point $x = -0.045m$ for different dissociation models.

Figure 3.10: Comparison of vibrational energy distribution function for different models. The lines are the Boltzmann distribution function corresponding to the vibrational temperature at the selected location. The symbols are the results from different models.

Table 3.2: DSMC results for stagnation point pressure and heat flux, using different vibrational relaxation models and dissociation models.

Simulation	Vibration relaxation model	Dissociation model	Pressure (Pa)	Heat flux (W/cm^2)
1	Model 2	No dissociation	770.54	42.52
2	Model 3	No dissociation	769.73	42.30
3	Model 4	No dissociation	770.07	42.43
4	Model 4	TCE model	767.20	34.51
5	Model 4	VFD model	767.55	36.66
6	Model 4	QK model	765.53	31.73

These models are then implemented in a modular fashion within a three-dimensional Cartesian grid based DSMC code [43]. During such implementation, the effect of the inelastic collision selection procedure on the relaxation process is also discussed. Several conclusions are draw from this chapter:

1) Using the sequential probability selection procedure as a consistent framework, existing DSMC models for rotational and vibrational relaxation are implemented in a modular fashion, and compared with each other using both isothermal and adiabatic relaxation calculations as well as one dimensional stagnation line simulations for hypersonic flow conditions. The rotational relaxation models implemented in this chapter, have inelastic collision probabilities which correspond to either a constant collision number or a Parker temperature dependent collision number [74]. Regarding the vibrational relaxation models implemented in this chapter, an expression for $Z_{vib}(T)$, parametrized by molecular properties, proposed by Bird [16] (where an effective collision-based temperature is used to calculate $Z_{vib}(T)$) is found to predict a significantly faster vibrational relaxation rate compared to other models constructed to reproduce relaxation data from Millikan and White [69]. For relaxation calculations, when Bird's effective collision-based temperature model is used with $Z_{vib}(T)$ matching Millikan and White data, agreement with the other models considered was obtained. Furthermore, the cell-averaged velocity dependent collision probability model of Boyd [23] was compared with the direct velocity dependent model of Boyd [23, 88] (probability determined for each collision), with Abe's correction to achieve equipartition of energy at equilibrium [1]. It

was found that both models agreed precisely for isothermal vibrational relaxation calculations, where the mean collision time is a constant, however showed a small difference in vibrational temperature profiles for adiabatic relaxations. For the one dimensional stagnation line flow simulation, the two models of Boyd agreed precisely, while the approach of Bird (also fit to Millikan and White data) predicted a slightly faster vibrational relaxation rate. All three vibrational relaxation models predicted virtually the same stagnation point pressure and heat flux.

2) Three existing dissociation models for the DSMC method are implemented in a modular fashion. Using a consistent vibrational relaxation model, the three dissociation models are compared using one dimensional stagnation line simulations. When using the same reaction rate constant (k_f), as expected, the TCE model is shown to give a higher dissociation rate of N_2 compared to the VFD model using a vibrational favoring parameter of $\phi = 3$ [47]. The QK dissociation model predicted the highest N_2 dissociation rate of any model. Thus the largest difference in stagnation point heat flux (roughly 15%) was found between the QK and VFD models, while all models predicted the same stagnation pressure (refer to Table 3.2).

The purpose of this chapter is not to make conclusions about the physical accuracy of any of the models investigated. Rather, the purpose is to present a consistent and rigorous comparison of the most widely used DSMC internal energy transfer and dissociation models. If DSMC predictions are to be compared to continuum CFD predictions for near-continuum flows, or the development of hybrid DSMC-CFD methods are to be extended to chemically reacting flows, then such models for internal energy relaxation and dissociation must be precisely defined and properly implemented. The material presented in this chapter is aimed towards this general goal.

Finally, to reduce the ambiguity in the comparisons, all the models implemented and compared in this chapter are based on data and information generally available from existing literatures and publications. The adjustment of the model parameters may be possible on a case-by-case basis, in order to bring the models into closer agreement with each other, or experimental data.

Chapter 4

Nonequilibrium-Direction-Dependent Rotational Energy Model

4.1 Chapter Introduction

As discussed in the previous chapter, in continuum simulations, rotational energy relaxation is usually modeled by the Jeans equation [48],

$$\frac{d\hat{\varepsilon}_r}{dt} = \frac{\hat{\varepsilon}_r^*(t) - \hat{\varepsilon}_r(t)}{\tau_r} \ , \tag{4.1}$$

where $\hat{\varepsilon}_r(t)$ is the average rotational energy at time t with ζ_r degrees of freedom, and τ_r is the characteristic rotational relaxation time. $\hat{\varepsilon}_r^*(t)$ is the instantaneous equilibrium rotational energy, which is defined according to the instantaneous translational temperature $T_t(t)$,

$$\hat{\varepsilon}_r^*(t) = \frac{\zeta_r}{2} k_B T_t(t) \ , \tag{4.2}$$

where k_B is the Boltzmann constant. τ_r is usually expressed in terms of the mean collision time τ_c, and the rotational inelastic collision number Z_{rot},

$$\tau_r = \tau_c Z_{rot} \ . \tag{4.3}$$

For DSMC [15] simulations, various models for rotational energy relaxation have been proposed [15, 21, 22, 25, 26, 33, 49, 60–63, 95]. Almost all of these models are phenomenological and based on the Borgnakke-Larsen (BL) approach [19], where each selected collision pair is tested for an inelastic collision (involving energy exchange between the translational and rotational energy modes, ε_t and ε_r), with a collision probability p_{rot}. This probability, p_{rot}, is then either directly or indirectly related to Z_{rot} in some fashion, as discussed in the previous chapter.

The rotational inelastic collision number, Z_{rot} in Eq. 4.3, has been shown to depend on the gas temperature in previous theoretical [74], computational [14, 66, 71], and experimental studies [10, 30, 42, 50, 59]. In general, previous continuum and DSMC models successfully established the temperature dependence of Z_{rot}. However, most of the models are constructed based upon some variant of the Parker model [74]. The Parker model was derived analytically by invoking several physical assumptions [66, 87, 95], and the resulting values of Z_{rot} are generally higher than predictions reported using more advanced computational methods [14, 71, 87]. Furthermore, in the Parker model, Z_{rot} is a function of the translational temperature only. However, it has often been questioned if the rotational collision number, Z_{rot}, may depend not only on the translational temperature, but also on the *direction* toward the equilibrium state [21, 66, 87]. For example, compressing flows (such as a shock wave) involve regions where the rotational energy in the gas is much less than the translational energy, and thus rotational energy is being excited. Whereas, expanding flows involve regions where rotational modes can become partially frozen, containing more energy than the translation modes, and thus rotational energy is being de-excited. For the same translational temperature, Z_{rot} may not be the same for both cases. Finally, although the previous rotational relaxation models established a temperature dependence of Z_{rot}, some of the models fail to satisfy the equipartition of energy between the translational and rotational energy modes under thermal equilibrium [1, 22, 33].

All of these aspects of rotational energy modeling in DSMC are thoroughly assessed in a review article by Wysong and Wadsworth [95]. Specifically, the article concludes that difficulties remain in reducing state-to-state rate constants to a rotational collision number expression (i.e., Z_{rot}) due to complications involving collision selection routines (how p_{rot} is precisely related to Z_{rot}), complications achieving equipartition of energy,

and approximations required to obtain an analytical form for the model. More fundamentally, the article concludes that, *"the primary complicating factor, and that which may be of most use in improving phenomenological models of the process, is the potentially large effect of the size and sign of the initial deviation from equilibrium"* [95]. The present chapter addresses each of these issues in-turn, resulting in a computationally efficient and physically realistic model for rotational energy relaxation that can be used in both continuum (Navier-Stokes) and DSMC simulations.

The development of this model was directly enabled through a recent study by Valentini, Zhang, and Schwartzentruber [87], where the rotational relaxation of nitrogen was analyzed using Molecular Dynamics (MD) simulation. The MD calculations provided the detailed quantitative information required to establish the dependence of Z_{rot} on both the temperature of the system and the initial nonequilibrium direction (i.e., compression versus expansion). The interatomic potential used for nitrogen and the MD simulation results were extensively validated against other computational results and experimental data [87]. In theory, these MD results could be used to form a state-resolved model for translational-rotational energy exchange in nitrogen (examples of such models can be found in Refs. 76, 79, 93), or alternatively to determine complete post-collision state probability density functions to be used with a cross-section based model such as the Dynamic Molecular Collision (DMC) model [81]. However, the premise of the current chapter is that such physically complex (and computationally expensive) models may not be necessary to accurately describe translational-rotational energy transfer. As an example, as part of the initial MD study [87], a demonstrative DSMC model was implemented that included dependence of Z_{rot} on both T_t and T_r, resulting in a significant increase in accuracy compared to the Parker model for both compressing and expanding flows. This demonstrative model relied on cell-averaged temperatures. The purpose of the current chapter is to develop a model more applicable to DSMC simulations that is based only on the properties of molecules involved in a specific collision.

This chapter is organized as follows: section 4.2 describes the formation of the proposed Nonequilibrium-Direction-Dependent (NDD) model. Section 4.2.1 details how the inelastic collision selection procedure should be generally handled, section 4.2.2 describes the functional form of the model and consistency with the Jeans equation in

the continuum limit, section 4.2.3 presents a general form for the post-collision energy distribution, outlines its use within the BL model framework, and explains the behavior of prior rotational models in the literature. Section 4.3 describes the NDD model parameterization using data from MD and lists the final NDD model equations for nitrogen. The model is validated with pure MD solutions for normal shock waves and one dimensional (1D) expansions in section 4.4, and the conclusions of this chapter are summarized in section 4.5. Relevant supporting mathematical analysis for the model development is included in Appendix A.

4.2 Nonequilibrium-Direction-Dependent Model Formulation

In this section, we formulate the Nonequilibrium-Direction-Dependent (NDD) rotational collision model for translational-rotational energy exchange. The DSMC collision model uses a probability of inelastic collision of the form,

$$p_{rot} = p_{rot}(\varepsilon_t, \varepsilon_r; \zeta_t, \zeta_r) = p_{rot}(\varepsilon_t, \varepsilon_r) \ . \tag{4.4}$$

Here ε_t and ε_r are the relative translational energy and rotational energy involved in an individual collision with translational and rotational degrees of freedom ζ_t and ζ_r, respectively. This probability of inelastic collision is then incorporated within the general Borgnakke-Larsen (BL) model [19] framework. In the continuum limit, the model reduces to a rotational collision number of the form $Z_{rot} = Z_{rot}(T_t, T_r)$, incorporated into the Jeans equation (Eqs. 4.1-4.3), which is appropriate for multi-temperature computational fluid dynamics (CFD) simulations. The general formulation of such a model is not trivial and involves a number of considerations not clearly described in the literature. Specifically, the NDD model formulation includes dependence on the specific DSMC inelastic collision selection procedure (how molecules or molecule pairs are stochastically selected for inelastic collisions). Furthermore, a general expression for the correct BL post-collision energy distribution is derived that ensures equipartition of energy. Finally, the formulation includes a connection factor accounting for the energy gain (or loss) that is required for consistency with the Jeans equation. All of these considerations

are accounted for while ensuring the physical accuracy and numerical efficiency of the resulting NDD collision model.

4.2.1 Inelastic collision selection procedure

As discussed in depth in Refs. 48 and 67, the exact probability expression used within a DSMC simulation (Eq. 4.4), must depend on the *collision selection procedure* employed. As discussed in the previous chapter, the most popular collision selection procedures include pair selection [48, 67], particle selection permitting double relaxation [15], and particle selection prohibiting double relaxation [44, 48]. Combining the same probability expression with different selection procedures will generally result in different simulated relaxation processes. Thus, although absent in the majority of DSMC articles, it is stressed that both the collision model(s) *and* the collision selection procedure should be identified when presenting DSMC simulation results. In regards to new model development, the inelastic collision selection procedure effectively determines the available degrees of freedom within each collision. As will be described in upcoming sections, the available degrees of freedom must be specified to sample the correct BL post-collision energy distribution and to account for the energy gain connection factor required for consistency with the Jeans equation.

If the collision pair consists of a molecule colliding with an atom, then the only internal degrees of freedom are that of the molecule and all selection procedures produce the same result. However, if the collision is between two molecules, then different selection procedures will result in different energy modes participating in the collision. For example, consider an inelastic collision between two molecules having rotational energy of $\varepsilon_{r,1}$ and $\varepsilon_{r,2}$, associated with $\zeta_{r,1}$ and $\zeta_{r,2}$ degrees of freedom, respectively. For the pair selection procedure [48, 67], both molecules participate in the inelastic collision, and the rotational energy in the collision is $\varepsilon_r = \varepsilon_{r,1}+\varepsilon_{r,2}$, associated with $\zeta_r = \zeta_{r,1}+\zeta_{r,2}$ degrees of freedom. The post collision energy exchange is achieved between ε_t and ε_r first, and then between the two rotational energy modes $\varepsilon_{r,1}$ and $\varepsilon_{r,2}$. Whereas for the particle selection permitting double relaxation [15] and the particle selection prohibiting double relaxation procedures [44, 48], only one of the two molecules participates in the collision with its rotational energy. The other molecule is treated as an atom and its rotational energy does not change during the collision. The energy exchange is between

ε_t and ε_r, where $\varepsilon_r = \varepsilon_{r,i}$ and associated with $\zeta_r = \zeta_{r,i}$ degrees of freedom ($i = 1$ or 2 depending on which molecule is participating in the collision).

The proposed NDD rotational energy exchange model is derived generally in terms of the energies (ε_t, ε_r) and degrees of freedom (ζ_t, ζ_r) that *participate in selected collisions* within the DSMC simulation. Thus, the model equations derived in this chapter are valid for all types of inelastic collision selection procedures as long as the participating energy modes and degrees of freedom are assigned consistently. In terms of the participating translational degrees of freedom, ζ_t should be set consistently with the cross-section model employed, such as the variable hard sphere (VHS) model [15], for which $\zeta_t = 5 - 2\omega$, where ω is the exponent of the desired temperature dependent viscosity, having the form $\mu = \mu_{ref}(T/T_{ref})^{\omega}$, as discussed in the previous chapter. This general formulation for participating energy modes is used in section 4.2.3.1 to derive the correct BL equilibrium distribution to sample post-collision states. In section 4.3 (and also detailed in Appendix A.2), the participating energy modes determine the energy gain connection factor [48, 67] required for consistency with the Jeans equation. However, this connection factor can only be obtained analytically for simple models (such as constant p_{rot} model), and for the present NDD model it is determined through simulations. Thus, the derived model equations are general to all collision selection procedures, except for the value of this connection factor. Specifically, the NDD model is developed using the VHS model and the modified form [97] of the particle collision prohibiting double relaxation procedure [48], which is therefore used to determine the connection factor. As discussed in Ref. 48, this selection procedure has clear advantages for multi-species simulations. A simple and computationally efficient implementation of the selection procedure, which has been proposed in Ref. 97 and detailed in the previous chapter, is used here for the NDD model development.

4.2.2 Functional form for p_{rot} and consistency with Z_{rot}

Since rotational energy relaxation is typically much faster than vibrational and chemical relaxation, strong thermochemical nonequilibrium (where velocity and energy distribution functions may not be Maxwell-Boltzmann) is required for significant rotational-translational nonequilibrium. Thus, the primary purpose of this chapter is to develop

a DSMC model based only on individual collision properties that is therefore physically accurate for flows in strong nonequilibrium. A secondary objective is to formulate this model such that, in the continuum limit, it reduces to a Jeans equation model with a temperature dependent rotational collision number. Whereas prior DSMC rotational models were constructed to reproduce a specific, near-equilibrium, temperature dependent collision number (such as the Parker model) [21, 22], the proposed model is constructed using the recent nonequilibrium MD results detailed in Ref. 87.

To this purpose, it is desirable to find a functional form for the DSMC probability model (p_{rot}) that is able to accurately reproduce the nonequilibrium-direction dependence clearly seen in the MD results, has an analytical form in the continuum limit, and is computationally efficient to employ. We find that a power law dependence, $p_{rot} \sim (\varepsilon_t/\varepsilon_r)^n$ and $Z \sim (T_r/T_t)^n$, has each of these desirable characteristics. Specifically, for the temperature dependent rotational collision number, we use an expression of the form [39, 87],

$$Z_{rot}(T_t, T_r) = \frac{Z_r^\infty}{1 + \frac{T^*}{T_t}} \left(\frac{T_r}{T_t}\right)^n . \tag{4.5}$$

Therefore, in the thermal equilibrium limit ($T_t = T_r = T$), this reduces to $Z_{rot}(T) = Z_r^\infty/(1+\frac{T^*}{T})$. The constants Z_r^∞ and T^* are determined by comparing with the MD data in the thermal equilibrium limit; specifically $T^* = 180$ and $Z_r^\infty = 7.7$ are shown to give the best agreement for N_2-N_2 collisions. Figure 4.1 plots the new model result alongside prior numerical results for the temperature dependence of the rotational collision number. It is noted that compared to the Parker model, the other computational studies determined a much weaker temperature dependence for Z_{rot}. Indeed, the MD results [87] showed that the nonequilibrium-direction and initial degree of nonequilibrium has a more dominant effect on Z_{rot} than the translational temperature.

In order to determine an analytical functional form for the nonequilibrium DSMC probability model (p_{rot}), we start with the following expression that links the collision probability with a collision number [21, 22, 33, 95]:

$$\frac{1}{Z_{rot}(T_t, T_r)} = \int_0^\infty \int_0^\infty \tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t, \varepsilon_r; T_t, T_r) d\varepsilon_t d\varepsilon_r . \tag{4.6}$$

In this equation, $\tilde{p}_{rot}$ is an intermediate probability (which will be explained later in this

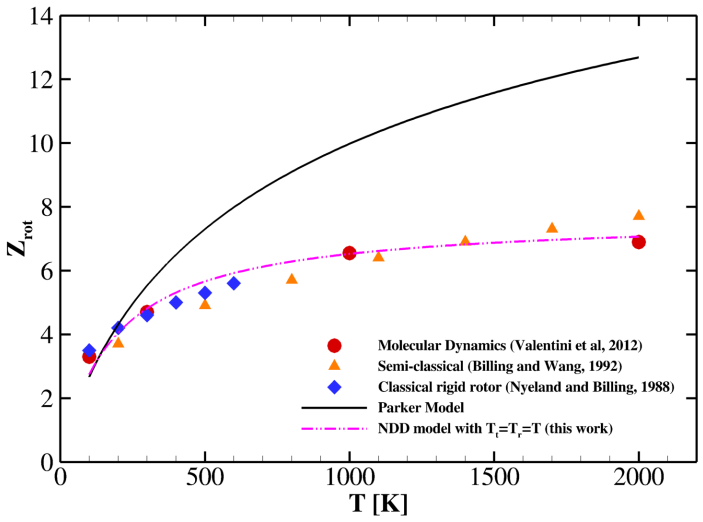


Figure 4.1: Comparison of rotational collision number Z_{rot} as a function of temperature in the limit of thermal equilibrium ($T_t = T_r = T$) for N_2-N_2 collisions.

section), and $f(\varepsilon_t, \varepsilon_r; T_t, T_r)$ is the joint equilibrium energy distribution function of ε_t and ε_r (that participate in *selected* collisions as discussed in section 4.2.1) corresponding to temperature T_t and T_r. This joint distribution can be expressed as

$$f(\varepsilon_t, \varepsilon_r; T_t, T_r) = f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) \ , \tag{4.7}$$

where for a continuous energy mode ε_i with ζ_i degrees of freedom, the equilibrium energy distribution function corresponding to temperature T_i is [15, 52],

$$f(\varepsilon_i; T_i) = \frac{1}{\Gamma(\frac{\zeta_i}{2}) k_B T_i} \left(\frac{\varepsilon_i}{k_B T_i} \right)^{\frac{\zeta_i}{2}-1} e^{-\frac{\varepsilon_i}{k_B T_i}} \ , \tag{4.8}$$

where $\Gamma(\cdot)$ is the Gamma function, and the subscript i could represent either translation (t) or rotation (r).

Equation 4.6 is a generalization of the expressions used in the development of prior rotational models [21, 22, 33] based on collision quantities (such as ε_t, or $\varepsilon_c = \varepsilon_t + \varepsilon_r$). Upon substitution of Eq. 4.5 into Eq. 4.6, we find the following unique solution for $\tilde{p}_{rot}$

(this derivation is detailed in Appendix A.1):

$$\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) = \frac{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})}{\Gamma(\frac{\zeta_t}{2}+n)\Gamma(\frac{\zeta_r}{2}-n)Z_r^{\infty}} \left[1 + \left(\frac{\zeta_t}{2} + n - 1\right)\frac{k_B T^*}{\varepsilon_t}\right]\left(\frac{\varepsilon_t}{\varepsilon_r}\right)^n \ . \tag{4.9}$$

For this solution, $n \neq 1$ and its precise value is determined later through comparison with the MD data in section 4.3. As discussed in section 4.2.1, the energies (ε_t, ε_r) and associated degrees of freedom (ζ_t, ζ_r) should be those that participate in *selected* collisions within the DSMC simulation. In this manner, the solution in Eq. 4.9 is valid for any inelastic collision selection procedure.

It is important to note that $\tilde{p}_{rot}$ (i.e., in Eq. 4.6 and 4.9) is *not* the probability that should be applied within the DSMC simulation. In general, satisfying Eq. 4.6 (and thus the direct use of $\tilde{p}_{rot}$ in DSMC) does not guarantee that the energy relaxation rate simulated by DSMC will be consistent with the Jeans equation (using the corresponding Z_{rot}) [95]. This is due to an inconsistency between the average energy gain in the Jeans equation compared to the average energy gain resulting from inelastic collisions within a DSMC simulation. Essentially, although $\tilde{p}_{rot}$ and Z_{rot} consistently define the average rate of inelastic collisions, the energy gain resulting from these collisions within DSMC is not contained in the expression for Z_{rot}, and thus the simulated energy relaxation process may not be consistent with the Jeans equation. This fact has been extensively studied by previous researchers for the constant Z_{rot} (and hence constant p_{rot}) situation [48, 67, 95]. Specifically, for this constant Z_{rot} situation, $\tilde{p}_{rot} = \frac{1}{Z_{rot}}$ (see Eq. 4.6), and a connection factor C was proposed as

$$p_{rot} = C\tilde{p}_{rot} \ , \tag{4.10}$$

where p_{rot} *is* the probability to be directly applied in the DSMC simulation. The exact form of the required connection factor C depends on the DSMC post-collision energy redistribution (for example the BL method), which depends on the degrees of freedom associated with ε_t and ε_r participating in collisions. Thus, as discussed in section 4.2.1, C also depends on the inelastic collision selection procedure used. For the specific case where p_{rot} is a constant, it has been shown that $C = (\zeta_t + \zeta_r)/\zeta_t$, where the degrees of freedom are those participating in inelastic DSMC collisions [48, 67]. As will be detailed in section 4.3 (and in Appendix A.2), there is no closed-form expression for

C for the proposed NDD rotational energy model, and its value must be determined through simulation [95].

In summary, the final NDD model for the probability of inelastic collision directly applied to selected DSMC collision pairs (p_{rot}) is given by Eqs. 4.9 and 4.10. The inelastic collision number expression (Z_{rot} in Eq. 4.5) is analytically determined in closed-form to be consistent with $\tilde{p}_{rot}$ for any general collision cross-section model and inelastic collision selection procedure. In the limit of thermal equilibrium ($T_t = T_r = T$), the model reproduces the MD data (see Fig. 4.1). The only remaining parameters to be determined include: (1) the power law exponent n, which is required to accurately model the dependence of rotational relaxation on the initial degree of nonequilibrium and direction to the equilibrium state, and (2) the final value of C.

4.2.3 Detailed balance and equipartition of energy

Prior to determining the parameters n and C, which involves comparisons between DSMC and MD simulation results, a description of the post-collision energy redistribution employed by the new NDD model is required. Specifically, the NDD model is developed within the general Borgnakke-Larsen (BL) framework [19]. In this subsection, a general procedure for collision-quantity based relaxation models is presented that ensures detailed balance and the equipartition of energy at equilibrium. This general procedure not only explains the implementation used in the new NDD model, but also mathematically resolves a number of prior reported failures, as well as various corrections, regarding detailed balance and the equipartition of energy found in the literature.

4.2.3.1 Post-collision equilibrium distribution ensuring detailed balance

Consider the probability that an arbitrary collision pair with specific participating energies $(\varepsilon_t, \varepsilon_r)$ will be selected to undergo an inelastic collision within a DSMC simulation of an equilibrium gas ($T_t = T_r = T$). First, the selection of an elastic collision pair is based on the pair's relative translational energy (ε_t) specific to a particular cross-section model (such as the VHS model with associated ζ_t). Thus, ε_t found in selected elastic collision pairs follows the equilibrium distribution, $f(\varepsilon_t; T)$, given in Eq. 4.8. Next, since the rotational energy, ε_r, of each molecule is independent of ε_t in an equilibrium gas, the probability density that a selected elastic collision pair has a rotational energy

equal to ε_r is $f(\varepsilon_r;T)$, again given in Eq. 4.8. Thus, the probability density that a collision pair with energies $(\varepsilon_t, \varepsilon_r)$ will be selected for an elastic collision is $f(\varepsilon_t;T)f(\varepsilon_r;T)$. Finally, inelastic collision pairs are selected from available elastic collision pairs with a probability $p_{rot}(\varepsilon_t, \varepsilon_r)$ (specific to a particular rotational relaxation model) using an acceptance-rejection method. The overall result is that, within a DSMC simulation of an equilibrium gas, an arbitrary collision pair with specific participating energies $(\varepsilon_t, \varepsilon_r)$ and degrees of freedom (ζ_t, ζ_r) will be selected to undergo an inelastic collision with the general probability density: $f(\varepsilon_t;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_t, \varepsilon_r)$. In what follows, we show that in order to maintain the equilibrium state within the general BL model framework, the post-collision properties of inelastic collisions should be sampled from this same distribution: $f(\varepsilon_t;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_t, \varepsilon_r)$.

Specifically, for elastic collisions, detailed balance is achieved through the collision cross-section model (i.e., the VHS model). Since during an elastic collision, ε_t does not change, and the elastic collision cross-section depends only on ε_t, the microscopic reversibility relationship will be satisfied, and the translational energy will stay in the correct equilibrium state. However for an inelastic collision (with participating ε_t and ε_r), the post collision relative translational energy ε_t' and rotational energy ε_r' should satisfy the following equation,

$$\varepsilon_t' + \varepsilon_r' = \varepsilon_t + \varepsilon_r = \varepsilon_c \ , \tag{4.11}$$

where ε_c is the collision energy. Since the collision pair $(\varepsilon_t, \varepsilon_r)$ selected to undergo inelastic collision, follows the distribution $f(\varepsilon_t;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_t, \varepsilon_r)$, as discussed above, then within the BL model framework, the post collision energies $(\varepsilon_t', \varepsilon_r')$ should be sampled from the same distribution, i.e., $f(\varepsilon_t;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_t, \varepsilon_r)$, in order to maintain the equilibrium state of the system. We note that this distribution function is general to any cross-section model, collision selection procedure, and any rotational energy exchange probability model (p_{rot}). The idea here is that, by sampling the same distribution function for both the collision pair selection and post collision energy redistribution, all the selected inelastic collision pairs will stay in the correct equilibrium state. As all particles in the system have a chance to be selected for inelastic collision, this will ensure that the whole system maintains the equilibrium state.

4.2.3.2 Acceptance-rejection sampling

In order to determine (sample) post collision energies $(\varepsilon_t', \varepsilon_r')$ after an inelastic collision, the acceptance-rejection technique is typically employed. To sample a continuous distribution $f(x)$, the standard inequality used is: $u \leq f(y)/M$. Here, the value y is drawn from a uniform distribution and $f(y)$ is evaluated and normalized by a constant that is greater than or equal to the maximum value of the distribution function, $M \geq f(x)|_{max}$. A second value, u, is then drawn from a uniform distribution, and if the inequality holds, then the value y is accepted, and a sample value is determined as $x = y$. If the inequality does not hold, the value y is rejected and the process is repeated. In this manner, the resulting set of x values follow $f(x)$.

Within a nonequilibrium DSMC simulation employing the general BL framework, as discussed above, the post inelastic collision energies $(\varepsilon_t', \varepsilon_r')$ should be sampled from the distribution $f(\varepsilon_t; T) f(\varepsilon_r; T) p_{rot}(\varepsilon_t, \varepsilon_r)$ with the constraint that $\varepsilon_t + \varepsilon_r = \varepsilon_c$ is fixed. We note that, although T is often called the "collision temperature" or "an equilibrium temperature", as shown later in Eq. 4.18, T does not explicitly appear in the acceptance-rejection expression used in DSMC. Due to the constraint, it is equivalent to sample ε_r' from $f(\varepsilon_c - \varepsilon_r; T) f(\varepsilon_r; T) p_{rot}(\varepsilon_c - \varepsilon_r, \varepsilon_r)$ with a constant ε_c, and then let $\varepsilon_t' = \varepsilon_c - \varepsilon_r'$. For a given ε_c, we have the following inequality,

$$\begin{aligned} &[f(\varepsilon_c - \varepsilon_r; T) f(\varepsilon_r; T) p_{rot}(\varepsilon_c - \varepsilon_r, \varepsilon_r)]\,|_{max} \\ \leq &[f(\varepsilon_c - \varepsilon_r; T) f(\varepsilon_r; T)]\,|_{max}\ p_{rot}(\varepsilon_c - \varepsilon_r, \varepsilon_r)|_{max}\ , \end{aligned} \tag{4.12}$$

with $0 \leq \varepsilon_r \leq \varepsilon_c$. Following this inequality, we let,

$$M = [f(\varepsilon_t; T) f(\varepsilon_r; T)]\,|_{max}\ p_{rot}(\varepsilon_t, \varepsilon_r)|_{max}\ . \tag{4.13}$$

We then have,

$$\begin{aligned} I(\varepsilon_r;\varepsilon_c) =& \frac{f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_c-\varepsilon_r,\varepsilon_r)}{M} \\ =& \frac{f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_c-\varepsilon_r,\varepsilon_r)}{[f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)]|_{max}\ p_{rot}(\varepsilon_c-\varepsilon_r,\varepsilon_r)|_{max}} \\ =& \frac{f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)}{[f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)]|_{max}} \times \frac{p_{rot}(\varepsilon_c-\varepsilon_r,\varepsilon_r)}{p_{rot}(\varepsilon_c-\varepsilon_r,\varepsilon_r)|_{max}} \\ =& I_1(\varepsilon_r;\varepsilon_c) \times I_2(\varepsilon_r;\varepsilon_c) \end{aligned} \tag{4.14}$$

Note, in Eq. 4.14, we have dropped the dependence of $I(\varepsilon_r;\varepsilon_c)$ and $I_1(\varepsilon_r;\varepsilon_c)$ on T, since they do not depend on T explicitly, as shown later in Eq. 4.18. Therefore the procedure to sample ε_r' from $f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)p_{rot}(\varepsilon_c-\varepsilon_r,\varepsilon_r)$ with a constant ε_c, using the acceptance-rejection technique is as follows:

(1) Generate a random number R_1 uniformly distributed between $(0,1)$

(2) Calculate the value of $I(\varepsilon_r;\varepsilon_c)$ where $\varepsilon_r = R_1\varepsilon_c$

(3) Generate a different random number, R_2, uniformly distributed between $(0,1)$;

(4) If $R_2 \leq I(\varepsilon_r;\varepsilon_c)$, then the sample is accepted and $\varepsilon_r' = R_1\varepsilon_c$. Else, if $R_2 > I(\varepsilon_r;\varepsilon_c)$, then no sample is accepted and repeat the process (1)-(4) with new R_1 and R_2 values.

4.2.3.3 Implications for prior rotational models

Next, we simplify $I_1(\varepsilon_r;\varepsilon_c)$ and $I_2(\varepsilon_r;\varepsilon_c)$ based on specific rotational relaxation model expressions (specific p_{rot}) from the literature. In doing so, many of the issues and corrections associated with equipartition of energy reported for prior rotational models are clearly explained. This general analysis not only fully describes the implementation of the current NDD rotational model, but also provides a framework for the development

of future inelastic collision models. To start, using Eq. 4.8 we have the following,

$$\begin{aligned} & f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T) \\ =&\frac{1}{\Gamma(\frac{\zeta_t}{2})k_BT}\left(\frac{\varepsilon_c-\varepsilon_r}{k_BT}\right)^{\frac{\zeta_t}{2}-1}e^{-\frac{\varepsilon_c-\varepsilon_r}{k_BT}}\frac{1}{\Gamma(\frac{\zeta_r}{2})k_BT}\left(\frac{\varepsilon_r}{k_BT}\right)^{\frac{\zeta_r}{2}-1}e^{-\frac{\varepsilon_r}{k_BT}} \\ =&\frac{e^{-\frac{\varepsilon_c}{k_BT}}}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})(k_BT_t)^{\frac{\zeta_r}{2}}(k_BT_t)^{\frac{\zeta_t}{2}}}(\varepsilon_c-\varepsilon_r)^{\frac{\zeta_t}{2}-1}\varepsilon_r{}^{\frac{\zeta_r}{2}-1} \\ =&A(\varepsilon_c,T,\zeta_t,\zeta_r)(\varepsilon_c-\varepsilon_r)^{\frac{\zeta_t}{2}-1}\varepsilon_r{}^{\frac{\zeta_r}{2}-1}\ . \end{aligned} \tag{4.15}$$

The maximum value of $f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)$ for a fixed ε_c should appear at $\varepsilon_{r,0}$ where,

$$0=\frac{\partial}{\partial\varepsilon_r}\left[f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)\right]|_{\varepsilon_r=\varepsilon_{r,0}}\ , \tag{4.16}$$

which is,

$$\frac{\varepsilon_{r,0}}{\varepsilon_c}=\frac{\zeta_r-2}{\zeta_t+\zeta_r-4}\ . \tag{4.17}$$

Using Eqs. 4.15 and 4.17, we can write $I_1(\varepsilon_r;\varepsilon_c,T)$ (appearing in Eq. 4.14), as

$$\begin{aligned} I_1(\varepsilon_r;\varepsilon_c)=&\frac{f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)}{[f(\varepsilon_c-\varepsilon_r;T)f(\varepsilon_r;T)]|_{max}} \\ =&\left(\frac{\zeta_t+\zeta_r-4}{\zeta_r-2}\right)^{\frac{\zeta_r}{2}-1}\left(\frac{\zeta_t+\zeta_r-4}{\zeta_t-2}\right)^{\frac{\zeta_t}{2}-1}\left(\frac{\varepsilon_r}{\varepsilon_c}\right)^{\frac{\zeta_r}{2}-1}\left(1-\frac{\varepsilon_r}{\varepsilon_c}\right)^{\frac{\zeta_t}{2}-1}\ . \end{aligned} \tag{4.18}$$

We can clearly see from the above expression, that I_1 does not explicitly depend on temperature, rather it only depends on ε_r, ε_c, ζ_t and ζ_r. For N_2 with $\zeta_r=2$, the above expression for I_1 can be further simplified to

$$I_1(\varepsilon_r;\varepsilon_c)=\left(1-\frac{\varepsilon_r}{\varepsilon_c}\right)^{\frac{\zeta_t}{2}-1}\ . \tag{4.19}$$

Next, the exact form of $I_2(\varepsilon_r;\varepsilon_c)$ (appearing in Eq. 4.14) is analyzed for several rotational relaxation models from the literature that each have different expressions for p_{rot}.

p_{rot} = constant: For this model, since $p_{rot}|_{max}$ is the same as p_{rot}, $I_2=1$. Thus, the inequality appearing in the acceptance-rejection method becomes $R_2\leq I(\varepsilon_r;\varepsilon_c)=$

$I_1(\varepsilon_r;\varepsilon_c)$ (see Eq. 4.18 or 4.19), which is the same as that discussed by Borgnakke and Larsen [19] and has been used for a number of prior rotational relaxation models.

$\underline{\mathbf{p_{rot}} = \mathbf{p_{rot}}(\varepsilon_\mathbf{t})}$: This expression corresponds to the relative translational energy (ε_t) dependent rotational model proposed by Boyd [22], where the model is formulated to reproduce the Parker expression for $Z_{rot}(T)$ in the continuum limit. In this model, $I_2 = p_{rot}(\varepsilon_t)/\{p_{rot}(\varepsilon_t)|_{max}\}$. Without including I_2 in the inequality for acceptance-rejection sampling, there was difficulty in achieving equipartition of energy between ε_t and ε_r in the original model [1, 22, 33, 95]. Initially, a cell-averaged p_{rot} was used to correct this, which is thus constant for all collisions within a cell, resulting in $I_2 = 1$. This approach is similar to using a cell-averaged temperature to determine p_{rot} and has the effect of relaxing all parts of the distribution function (within a given cell) at the same rate. As will be shown later in section 4.4.3 this is generally not accurate for highly nonequilibrium flows. Later, Abe [1] proposed to multiply I_1 in the inequality ($R_2 \leq I_1(\varepsilon_r;\varepsilon_c)$) by a correction term equal to $p_{rot}(\varepsilon_t)/\{p_{rot}(\varepsilon_t)|_{max}\}$. This modification was demonstrated for hard sphere molecules, and the idea came from drawing an analogy between the probability bias introduced by a relative collision velocity dependent elastic collision rate, and that of the collision energies dependent p_{rot} model. This term is actually the same as I_2 (appearing in Eq. 4.14), that should be included in the acceptance-rejection sampling inequality.

$\underline{\mathbf{p_{rot}} = \mathbf{p_{rot}}(\varepsilon_\mathbf{c})}$: This expression corresponds to the total collision energy (ε_c) dependent model [21, 33] which is also based on the Parker expression for Z_{rot}. When using only I_1 in the inequality ($R_2 \leq I_1(\varepsilon_r;\varepsilon_c)$), this model does, in fact, satisfy equipartition of energy. In prior publications, this was explained based on the idea that ε_c is a collision invariant [20, 33]. An equivalent explanation is evident from Eq. 4.14, where since $p_{rot}(\varepsilon_c)|_{max} = p_{rot}(\varepsilon_c)$ for a fixed ε_c, the result is that $I_2 = 1$. This is a benefit of using a probability model based on collision invariants, however, for general physical accuracy it may not be desirable to be restricted to only these collision quantities.

$\underline{\mathbf{p_{rot}} = \mathbf{p_{rot}}(\varepsilon_\mathbf{t}, \varepsilon_\mathbf{r})}$: This expression corresponds to the Nonequilibrium-Direction-Dependent (NDD) model proposed in this chapter. For this model, and any similar model, $I_2(\varepsilon_r;\varepsilon_c) = p_{rot}(\varepsilon_c - \varepsilon_r, \varepsilon_r)/\{p_{rot}(\varepsilon_c - \varepsilon_r, \varepsilon_r)|_{max}\}$. To sample the post collision relative translational energy and rotational energy ($\varepsilon_t', \varepsilon_r'$), the correct formulation for the acceptance-rejection sampling inequality, is $R_2 \leq I(\varepsilon_r;\varepsilon_c) = I_1(\varepsilon_r;\varepsilon_c)I_2(\varepsilon_r;\varepsilon_c)$ as

detailed above.

Therefore, within the general BL framework, post-collision energies should be sampled from an equilibrium distribution that is consistent with the distribution of *selected* inelastic collision pairs. The general expression for this distribution (Eq. 4.14), and its use in acceptance-rejection algorithms, ensures detailed balance and equipartition of energy at equilibrium for any general collision model. This formulation resolves the behavior of prior rotational models, details the implementation of the new NDD model, and also provides a framework for the development of new collision-quantity-based DSMC models; for example, the formulation could be readily applied for vibrational relaxation models.

4.3 NDD Model Parameterization using Molecular Dynamics Data

4.3.1 Further considerations prior to parameterization

In this section, we determine the parameter n in the expression for $\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r)$ (Eq. 4.9), as well as the connection factor C appearing in Eq. 4.10, in order to arrive at the final NDD model expression for p_{rot}. In general, both parameters are determined by comparing DSMC and MD isothermal relaxation simulations. However, in order for the resulting model to remain as general as possible, two further considerations are required.

First, we propose to separate the connection factor (C) into an analytically determined constant (C_a) and another constant that is determined numerically through simulation (C_n). This choice is supported by the mathematical analysis included in Appendix A.2. Specifically, as discussed in section 4.2.2 in reference to Eq. 4.10, for the case of constant p_{rot} (using collision selection prohibiting double relaxation), the required connection factor is analytically determined [48, 67] as $C = (\zeta_t + \zeta_r)/\zeta_t$, by considering the inconsistency between the energy change in DSMC versus Jeans equation. For the general case, $p_{rot} = p_{rot}(\varepsilon_t, \varepsilon_r)$, the connection factor is not necessarily a constant and no simple analytical expression can be derived. However, it is expected that the general connection factor will still contain the dependence due to $C_a = (\zeta_t + \zeta_r)/\zeta_t$, and thus this constant should be maintained separately. We then augment this factor by

an additional constant [95] determined through numerical simulation, thereby adjusting the NDD model probability (from Eq. 4.10) to the form: $p_{rot}^{NDD} = C_n C_a \ \tilde{p}_{rot}$.

Second, in a DSMC simulation, there is a restriction on the maximum probability that can be simulated. Values of p_{rot} greater than 1 clearly can not be simulated, and in general, depending on the specific collision selection procedure used (see section 4.2.1), the maximum value of p_{rot} that can be simulated may be different [48, 67]. Since most DSMC models are phenomenological (including the proposed NDD model), such models may satisfy Eq. 4.6 and produce the desired Z_{rot} behavior (found from experimental or computational studies), yet encounter values of p_{rot} that can not be simulated within a DSMC simulation. Ideally, a general model should ensure that values of p_{rot} encountered lie within the range applicable to DSMC simulation. Thus, we use a cut-off probability of $\frac{1}{2}$, which is consistent with the inelastic collision selection prohibiting double relaxation procedure of Haas *et al.* [48, 97], and is also valid for the other selection procedures discussed in section 4.2.1. Thus, the NDD model probability is further adjusted as $p_{rot}^{NDD} = min\{C_n C_c \ \tilde{p}_{rot}, \frac{1}{2}\}$. This modification also has the benefit of defining the maximum value of p_{rot} that is required for the acceptance-rejection algorithm, particularly for the general case where $p_{rot} = p_{rot}(\varepsilon_t, \varepsilon_r)$. Ultimately the specific choice of this cutoff value may alter the numerically determined constant, C_n, but otherwise enables the NDD model to be generally used within any DSMC implementation.

4.3.2 Parameterization through comparison with Molecular Dynamics

Parameterization of the NDD model for nitrogen is enabled by recent data obtained through MD simulation [87]. These MD calculations provide detailed quantitative information for the dependence of Z_{rot} on both the temperature of the system and the initial nonequilibrium direction (i.e., compression versus expansion). It is noted that the interatomic potential (for nitrogen) used in the MD simulations was extensively validated against other computational results and experimental data in Ref. 87.

MD data computed from a large number of translational-rotational relaxation simulations is taken directly from Ref. 87 and re-plotted in Fig. 4.2. Here, Z_{rot} values in the figure correspond to different nonequilibrium initial states $(T_t(0), T_r(0))$ used for isothermal relaxation simulations. T_{inf} in the figure represents the temperature of the

system after reaching the equilibrium state. Specifically, in these MD simulations, nitrogen molecules were initialized within a periodic box according to Maxwell-Boltzmann velocity and Boltzmann rotational energy distributions corresponding to various initial states ($T_t \neq T_r$). The system was integrated with the MD method allowing isothermal (constant T_t) relaxation to the equilibrium state. It was found that the resulting relaxation profiles for $T_r(t)$ were well fit with by the isothermal solution to the Jeans equation (essentially an exponential fit):

$$\frac{T_r(\infty) - T_r(t)}{T_r(\infty) - T_r(0)} = \exp\left(-\frac{t}{Z_{rot}\tau_c}\right) . \tag{4.20}$$

Thus, each MD relaxation simulation was adequately represented by a single exponential fitting parameter, which we will refer to as Z_{rot-MD}. The values of Z_{rot-MD} resulting from a number of different initial temperature states (T_t, T_r) taken from Ref. 87, are plotted in Fig. 4.2 as solid symbols. The NDD model parameters n and C_n are determined through a two step fitting process to the MD data plotted in Fig. 4.2. First, we set $C_n = 1$ and determine the power law dependence n. Then, we fix this value of n and proceed to determine the required connection factor C_n.

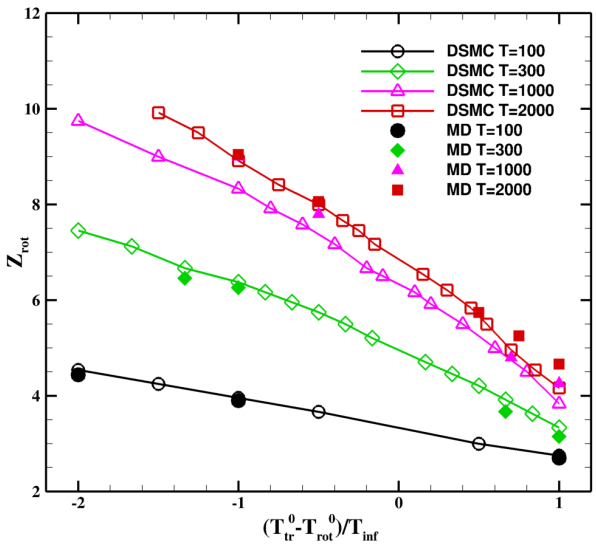


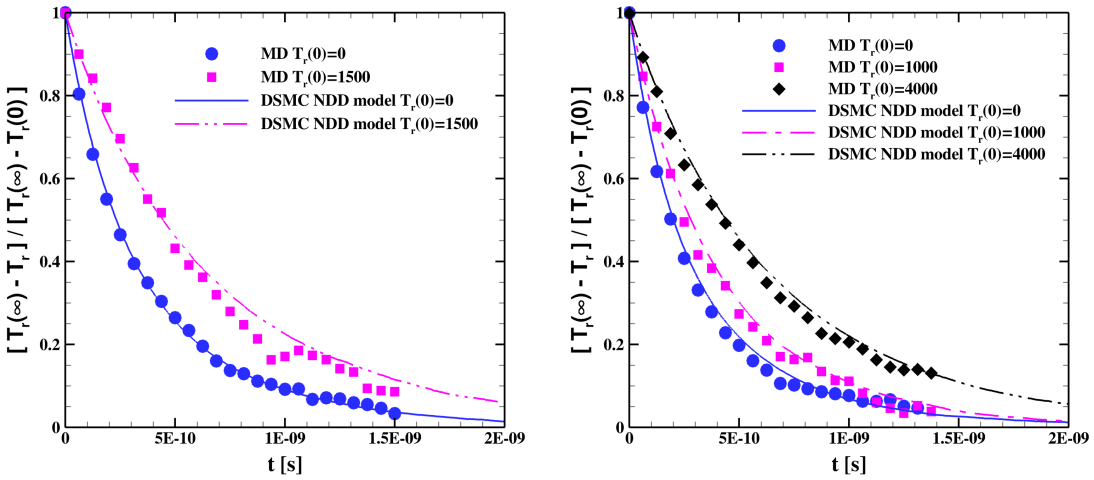
Figure 4.2: Comparison of the rotational collision number Z_{rot} computed by both DSMC (with the proposed NDD rotation model) and MD.

In step one, for each initial nonequilibrium state (T_t, T_r) with available MD data (see Fig. 4.2), a number of DSMC simulations were performed using various values for the power law dependence, n, but using a fixed value of $C_n = 1$. In the isothermal relaxation DSMC simulations, simulated particles are confined within a box with specular reflection walls. The translational temperature T_t (or the average relative translational energy, $\bar{\varepsilon_t}$) of the system is kept constant, while the rotational energy is allowed to relax. To maintain a constant T_t, the velocity of each particle is re-sampled from a Maxwell-Boltzmann distribution (corresponding to T_t) after the collision phase of each time step.

Similar to the MD relaxation curves discussed above, the DSMC curves are also fit by an exponential curve (specifically Eq. 4.20). Thus for each relaxation condition, a value of $\tilde{Z}_{rot-DSMC}$ is determined, where the $(\tilde{\cdot})$ denotes that these DSMC results are not yet final, since we have fixed $C_n = 1$. It is important to note that $Z_{rot-MD} = \tau_r/\tau_{c,KT}$, where $\tau_{c,KT} = \frac{4\mu}{\pi p}$ (the kinetic theory result), μ is the viscosity, and p is the pressure. Whereas, in DSMC simulation, the collision number is calculated according to $Z_{rot-DSMC} = \tau_r/\tau_{c,VHS}$, where $\tau_{c,VHS}$ is the mean collision time according to the VHS collision model. To have a consistent comparison, all DSMC collision numbers are reformulated to the kinetic theory defined collision number based on the relation between $\tau_{c,VHS}$ and $\tau_{c,KT}$ [48]. By comparing the resulting $\tilde{Z}_{rot-DSMC}$ values with the Z_{rot-MD} values, we find that the NDD model best reproduces the MD trends for $Z_{rot}(T_t, T_r)$ (refer to Fig. 4.2) when $n = \frac{1}{2}$.

In step two, we first directly evaluate values for $\tilde{Z}_{rot-DSMC}/Z_{rot-MD}$ for all relaxation simulations with $n = \frac{1}{2}$ and $C_n = 1$. We find that the resulting values have only a weak dependence on T_t and T_r, which suggests that a constant value for C_n may indeed be appropriate. In analyzing a previous rotational relaxation model [21] with $p_{rot} = p_{rot}(\varepsilon_c)$, Wysong et al. [95] also suggested the use of a constant C to account for the difference between Z_{rot} and p_{rot}. The same set of isothermal relaxation DSMC simulations are again performed, now for various C_n values and a fixed value of $n = \frac{1}{2}$. We determine that a value of $C_n = 1.92$ results in the best quantitative agreement between DSMC with the NDD model and MD simulation for all $Z_{rot}(T_t, T_r)$ values, and thus for all isothermal relaxation curves. Isothermal relaxation results simulated by DSMC using the NDD model are plotted in Figs. 4.3(a) and 4.3(b) for representative

translational temperatures $T_t = 1000$ and $T_t = 2000$ K, at different initial rotational temperatures $T_r(0)$. The MD simulation results are also plotted at the same initial temperature pairs for comparisons. As can be seen from Fig. 4.2 and 4.3, overall, the agreement between DSMC using NDD model and MD results is very satisfactory. We further note that since the translational temperature is constant, the Parker model can not be parametrized to reproduce the varying relaxation rates shown in Fig. 4.2 and 4.3.



(a) $T = 1000$ K. (b) $T = 2000$ K.

Figure 4.3: Comparison of rotational temperatures T_r relaxation curves, for isothermal relaxation simulation at two representative translational temperatures, with different initial T_r. The symbols are the MD results, the lines are the DSMC simulation results using the NDD model.

4.3.3 Final NDD rotational energy exchange model

Therefore, the final Nonequilibrium-Direction-Dependent (NDD) DSMC model uses the following probability of inelastic collision:

$$p_{rot}^{NDD}(\varepsilon_t, \varepsilon_r) = min\{C_n C_a \; \tilde{p}_{rot}(\varepsilon_t, \varepsilon_r), \frac{1}{2}\} \; , \tag{4.21}$$

where $\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r)$ is given in Eq. 4.9 with $n = \frac{1}{2}$. Since our DSMC implementation employs the collision selection prohibiting double relaxation procedure, we set $C_a = (\zeta_t + \zeta_r)/\zeta_t$. However, C_a could be set consistently with other inelastic collision selection

procedures. In all equations the energies $(\varepsilon_t, \varepsilon_r)$ and degrees of freedom (ζ_t, ζ_r) are those that participate within *selected* collisions in the DSMC simulation. Specific for nitrogen, we find $T^* = 180$ K and $Z_r^\infty = 7.7$. In general, the numerically determined connection factor, $C_n = 1.92$, is specific to our collision selection procedure, however, it is possible that the value of C_n is rather insensitive to the selection procedure due to the separation of C_a. The expression in Eq. 4.21 is directly used as the probability of an inelastic collision between a selected collision pair, and is also directly used for the BL post-collision energy redistribution within the acceptance-rejection algorithm detailed in section 4.2.3.2. In the continuum limit of Maxwell-Boltzmann energy distributions (at T_t and T_r), the model reduces to the rotational collision number $Z_{rot}(T_t, T_r)$ in Eq. 4.5, which is appropriate for use in multi-temperature CFD solvers. While the NDD model parameters are specific to N_2-N_2 collisions, the model framework and parameter fitting procedures could be repeated with further MD data relevant to other species.

The DSMC simulation results using this final model are referred to as $Z_{rot-DSMC}$ and are plotted against the Z_{rot-MD} values in Fig. 4.2. The DSMC simulation results with the new NDD model have good agreement with MD for all values of $Z_{rot}(T_t, T_r)$ and thus for all isothermal relaxation curves. Furthermore, the collision-based probability itself exhibits the expected collision physics. This is evident in Fig. 4.4 where for a fixed ε_t, the probability of rotational energy exchange decreases as ε_r increases. Essentially, non-rotating molecules have a higher probability of gaining rotational energy than highly-rotating molecules have of losing rotational energy.

4.4 NDD Model Validation for Compressing and Expanding Flows

Enabled by modern high performance computing, pure MD simulations of one dimensional flow features are now possible [84–87], which can be used to validate phenomenological models. Since the NDD model is the first to account for the dependence of rotational relaxation on the direction to equilibrium, we validate the model with pure MD simulations of both one-dimensional compressing flows (shocks) as well as expansions. As a reference, we also include DSMC results using the Parker model in the comparisons.

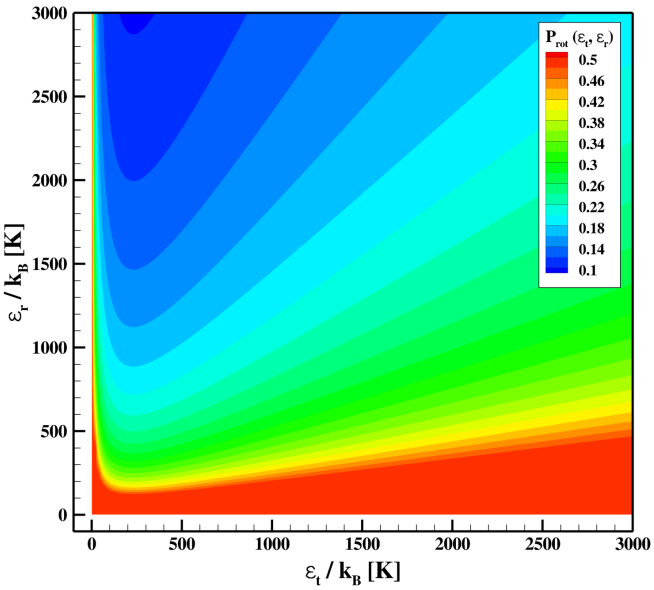


Figure 4.4: The rotational inelastic collision probability $p_{rot}^{NDD}(\varepsilon_t, \varepsilon_r)$ as a function of both ε_t and ε_r. ε_t and ε_r are normalized by the Boltzmann constant k_B.

4.4.1 Normal shock wave simulations

The MD simulation results were taken from Ref. 87, where the MD simulation technique was also fully described. The DSMC results were obtained using the Molecular Gas Dynamic Simulator (MGDS) code developed at the University of Minnesota [43, 98]. The VHS collision model was used along with the recently modified form [97] of the collision selection prohibiting double relaxation procedure [48]. Uniform grids and simulation time steps were used. The mean free path and mean collision time (denoted as λ_1 and τ_1) based on the upstream flow conditions were calculated using the VHS model. Two shock wave calculations were performed, for which the flow conditions and numerical parameters are listed in Table 4.1. It is important to note that the VHS model parameters for nitrogen, listed in Table 4.1, have been validated with experimental viscosity data in the appropriate temperature range and are also fully consistent with the interatomic potential used for the MD simulations (details are contained in Ref. 87). In all the shock wave simulations conducted here, simulation cells with upstream conditions

contained around 1000 simulated particles, while cells downstream of the shock contained approximately 5500 particles. The shock waves were verified to be stationary with no numerical stabilization techniques [15] necessary. The simulation domain was set between $(50-60)\lambda_1$ for the two shock wave conditions simulated. At the beginning of the simulation, particles were generated in the upstream and downstream half of the domain, with pre- and post-shock flow properties, respectively, and eight rows of buffer cells were employed at each end of the domain. Particles in the buffer cells were regenerated each time step to maintain the correct boundary conditions. Particle information was sampled for 10,000 time steps after steady-state was reached. Such a large sampling size, together with the large number of particles per cell, provided results with low statistical scatter.

Table 4.1: Shock wave conditions and DSMC simulation parameters

Mach number	T_1 (K)	ρ_1 kg/m^3	$\Delta t/\tau_1$	$\Delta x/\lambda_1$	ω	d_{ref} (m)	T_{ref} (K)
7	28.3	0.1	0.03	0.23	0.88	4.50×10^{-10}	273
7	300	0.1	0.02	0.16	0.72	4.17×10^{-10}	273

DSMC simulations were performed using both the proposed NDD rotational energy model as well as the Parker model. Specifically, the Parker model was implemented using a cell averaged temperature to compute the collision number Z_{rot} in each cell, using the relation:

$$Z_{rot}(T) = \frac{Z_r^{\infty}}{1 + a(T^*/T) + b(T^*/T)^{\frac{1}{2}}} \ , \tag{4.22}$$

where $Z_r^{\infty} = 23.5, a = \pi(1+\pi/4), b = \pi^{3/2}/2, T^* = 91.5$ K are model constants [21, 66], and T is the translational temperature. The continuum collision number Z_{rot} was then converted to DSMC collision probability using the connection factor C_a for the constant collision number model [48, 97]; $p_{rot}(T) = C_a/Z_{rot}(T)$. Since in the shock wave simulation, each cell contained a large number of particles (> 1000) , the cell averaged temperature is accurate within several percent for sampling over a single time step. Moreover, after reaching steady state, the temperature was computed based on accumulated samples, resulting in essentially no statistical scatter in the computed value

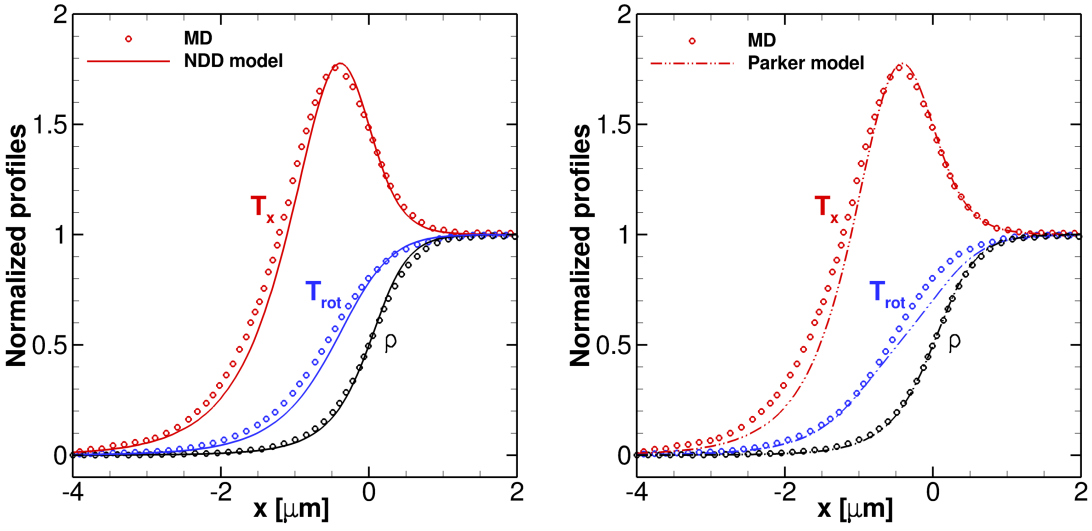
of $Z_{rot}(T)$. Results from all simulations were normalized using,

$$x_{norm} = \frac{x - x_1}{x_2 - x_1} \ , \tag{4.23}$$

where x is some flow variable, x_{norm} is the normalized variable, x_1 and x_2 are the upstream and downstream values of the corresponding variable, respectively.

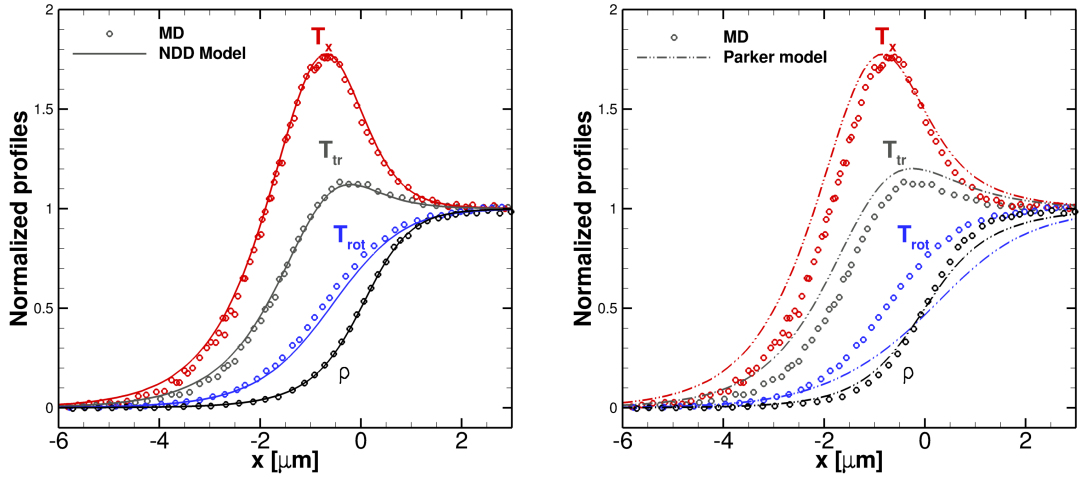
Simulation results for the low temperature shock wave ($T_1 = 28.3$ K) are shown in Fig. 4.5. Solutions using both NDD and Parker models are in close agreement with MD results for this case, with the NDD model in slightly better agreement than the Parker model for the rotational temperature in the later portion of the shock wave. These results are very consistent with Fig. 4.1 where the Parker model is seen to predict a slightly lower Z_{rot} value compared with MD for the low free stream temperature (< 50 K), however predicts a higher Z_{rot} value than MD (slower relaxation) for the post-shock temperature (~ 300 K). Since the flow is in strong compression, the NDD model actually simulates a $Z_{rot}(T_t, T_r)$ that is lower than the near-thermal equilibrium curve $Z_{rot}(T_t = T_r = T)$ shown in Fig. 4.1. In fact, additional DSMC simulations employing a constant Z_{rot} model determined that a value of $Z_{rot} \approx 1.8$ led to best agreement with the MD solution [87].

Simulation results for the high temperature shock wave ($T_1 = 300$ K) are shown in Fig. 4.6. Here, the NDD model is in excellent agreement with MD results, while the Parker model shows significant discrepancy, in particular a much slower rotational relaxation. This is again consistent with Fig. 4.1 in that the Parker model predicts a significantly higher Z_{rot} (slower relaxation) than MD and the other computational chemistry calculations for the temperature range of this shock wave ($300 \text{ K} < T < 3000$ K). Again, since the flow is in strong compression, the NDD model actually simulates a $Z_{rot}(T_t, T_r)$ value that is lower than the near-thermal equilibrium curve $Z_{rot}(T_t = T_r = T)$ shown in Fig. 4.1. For these shock conditions, additional DSMC simulations employing a constant Z_{rot} model determined that a value of $Z_{rot} \approx 4$ led to best agreement with the MD solution [87]. Certainly, the Parker model (Eq. 4.22) could be re-parametrized with such Z_{rot} values in order to match the MD data for these shock calculations. However, as just discussed, the resulting curve fit for $Z_{rot}(T)$ would lie below the MD curve (for near thermal equilibrium) seen in Fig. 4.1. Since the Parker



(a) NDD model (b) Parker model

Figure 4.5: Comparison of the shock profile at $T_1 = 28.3$ K. The symbols are for the MD results, the lines are for the DSMC simulation results using the NDD model, or the Parker model.



(a) NDD model (b) Parker model

Figure 4.6: Comparison of shock wave profiles at $T_1 = 300$ K, between two DSMC models and MD data.

model can not account for the direction to equilibrium, such a parameterization would lead to inaccurate results for expanding flows over a similar temperature range.

4.4.2 1D expansion simulations

In contrast to compressing flows, in expanding flows the rotational temperature is typically higher than the translational temperature (the direction toward equilibrium is reversed). We construct a one dimensional steady state flow field where particles are continually refreshed within an equilibrium reservoir (a buffer region) on the left side of the domain. Reservoir particle properties are sampled from Maxwell-Boltzmann distributions for a gas in thermal equilibrium ($T_t = T_r = T$) and a bulk velocity of zero. Thus, the particle flux from the reservoir into the computational domain is equal to the ratio of unidirectional normal momentum flux to mass flux through a stationary plane in an equilibrium reservoir [15]. The right side boundary (outlet) is set as a vacuum boundary condition. Due to the pressure difference, particles inside the flow field accelerate toward the outlet, and the bulk velocity inside the flow field is finite.The flow through the domain is in thermal nonequilibrium with a higher rotational temperature and a lower translational temperature, due to the rapid expansion. This simple 1D flow field is computationally feasible for pure MD simulation. MD results were previously obtained in Ref. 87 and are used here to compare with DSMC simulations of the same flow.

Three conditions with inlet temperatures of 1000 K, 2000 K, and 3000 K were simulated, all with an inlet density of 0.1 kg/m^3. A uniform grid was used with $\Delta x = (0.35-0.46)\lambda_1$ (depending on inlet temperature) and $\Delta t = 0.2\tau_1$, where the VHS model was used to calculate the inlet conditions (λ_1 and τ_1). Each cell contained between 200-600 simulation particles and, after reaching steady state, each cell was sampled for 20,000 time steps.

The simulation results are shown in Fig. 4.7 where the difference between the translational and rotational temperature ($T_t - T_r$) is shown for the NDD model and the Parker model compared to the MD solution. In Fig. 4.7, all simulations are seen to predict strong thermal nonequilibrium near the inlet, approach to thermal equilibrium ($T_t - T_r \approx 0$) in the middle portion of the simulation domain, and a departure from equilibrium near the vacuum exit (which is a boundary effect). While the NDD model result agrees very closely with the MD solution, the Parker model exhibits a noticeably slower rotational relaxation rate. Again, this is because the Parker model predicts higher values for Z_{rot} compared to the current MD calculations and therefore compared

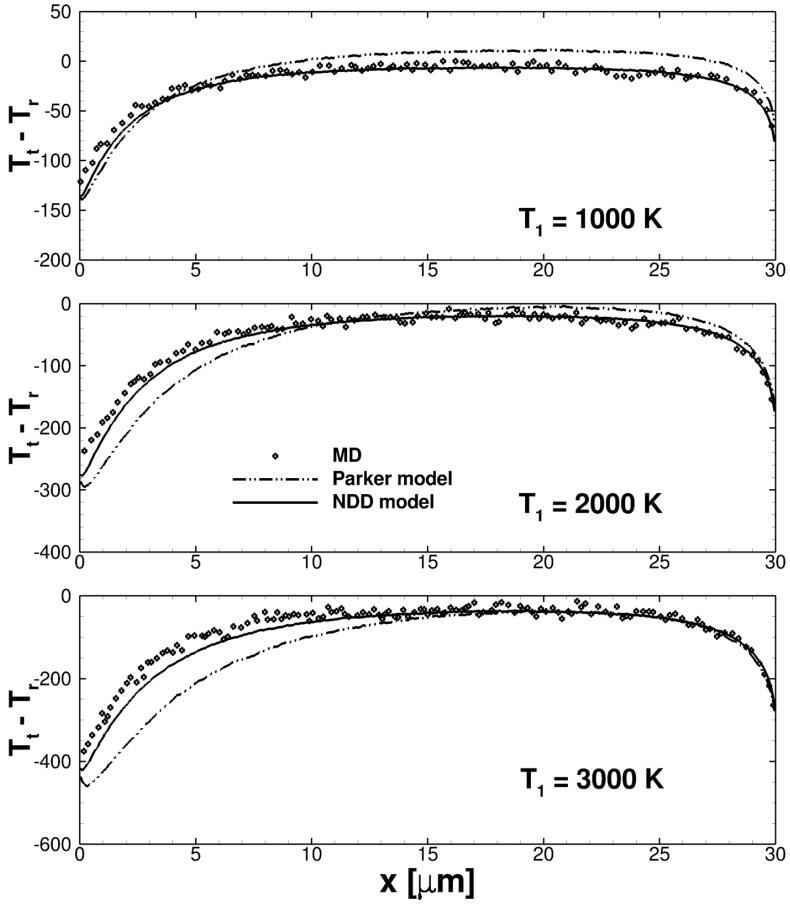


Figure 4.7: Results of the one dimensional expansion at different inlet temperatures. The MD results are compared with DSMC simulation results using the proposed NDD model and the Parker temperature dependent rotational collision number model.

to the NDD model. It is important to note that separate DSMC simulations employing a constant Z_{rot} model found closest agreement with MD result for $Z_{rot} \approx 6.9$ in the $T_1 = 2000$ K expansion simulation [87]. Thus, if the Parker model was re-parametrized to match this MD solution, it would no longer be able to accurately simulate the high temperature shock wave flow, which as discussed above, requires a $Z_{rot} \approx 4.0$ within the same equilibrium temperature range. Thus the proposed NDD model accurately reproduces pure MD solutions for all isothermal relaxation simulations, shock wave simulations, and expansion simulations. The current parameterization of the Parker model

predicts significantly higher Z_{rot} values compared to MD, and even if re-parameterized, the Parker model is not able to reproduce all MD results since it does not account for the direction towards the equilibrium state.

4.4.3 Rotational energy distribution functions

It is very interesting to investigate the agreement for rotational energy distribution functions (rdfs) within a shock wave between the proposed NDD model and pure MD simulation. Precise agreement is not expected since, in DSMC, only a fraction of collisions are considered as inelastic and exchange translational-rotational energy, which is redistributed using the phenomenological Borgnakke-Larsen model. While in MD simulations, all "collisions" are inherently inelastic, exchanging translational-rotational energy, and their post collision states are computed deterministically. In addition to comparing the level of agreement between the NDD model and pure MD simulation, it is also interesting to investigate the rdfs predicted by a collision-quantity based DSMC model versus a cell-averaged DSMC model. We thus implemented a collision probability model using the cell-averaged translational and rotational temperature $(\bar{T}_t, \bar{T}_r)$. In this model, the inelastic collision probability for each collision cell is a constant, and is directly related to a collision number using Eq. A.24, where the collision number for each cell is obtained from the cell averaged temperatures $(\bar{T}_t, \bar{T}_r)$ using Eq. 4.5 with the same parameters as NDD model. The average temperatures $(\bar{T}_t, \bar{T}_r)$ are maintained and updated through sampling in each cell and therefore at steady state are free of statistical scatter.

The temperature profiles (for the $T_1 = 28.3$ K shock wave, see Table 4.1) predicted by the NDD model, the cell-averaged model, and MD are compared in Fig. 4.8. All solutions are in close agreement. The rdfs predicted by each simulation are shown in Fig. 4.9, where the rdf points in the figure were chosen at locations corresponding to the same normalized T_r values. The rdfs obtained from DSMC were a function of ε_r and were plotted versus rotational quantum states j in the figure. Here, discrepancies between the simulations are now evident. In particular, the NDD model solution is in much better agreement with the pure MD solution compared to the cell-averaged model. Although the cell-averaged model accounts for the varying rate of rotational relaxation throughout the shock wave, due to the dependence on T_t and T_r local to

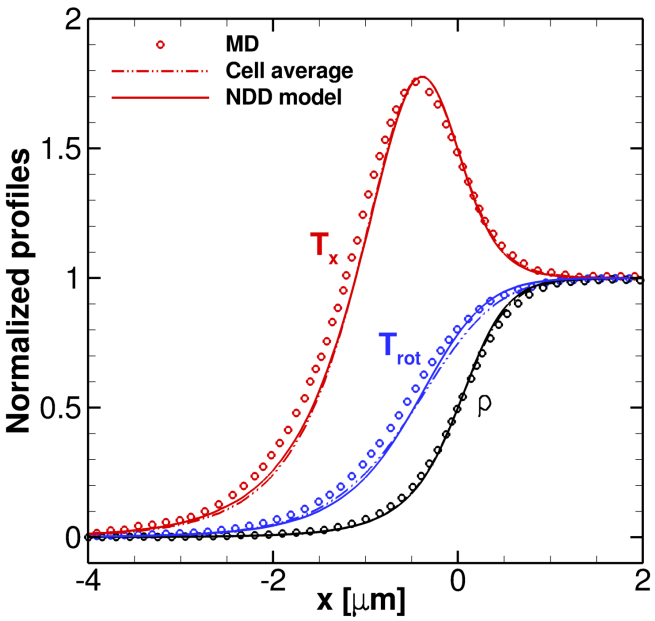


Figure 4.8: Comparison of shock wave profiles at $T = 28.3$ K between MD data, the NDD model, and a cell averaged temperature version of the NDD model.

each cell, it relaxes all parts of the local distribution function at the same rate (i.e., at the cell-average rate). In contrast, the collision-quantity based NDD model relaxes different parts of the distribution function at different rates (i.e., on a collision specific basis). This comparison clearly shows how a collision-quantity based DSMC model can be more physically accurate than a cell-averaged model. We note that this is not always clear in the predicted density and temperature profiles, as seen in Fig. 4.8. We also note that without the capability to perform pure MD simulation of the shock wave, such a conclusion would not be possible.

4.5 Chapter Conclusions

A new nonequilibrium-direction-dependent (NDD) rotational energy exchange model for DSMC and multi-temperature Navier-Stokes methods, is presented in this chapter. Several conclusions are draw from the study:

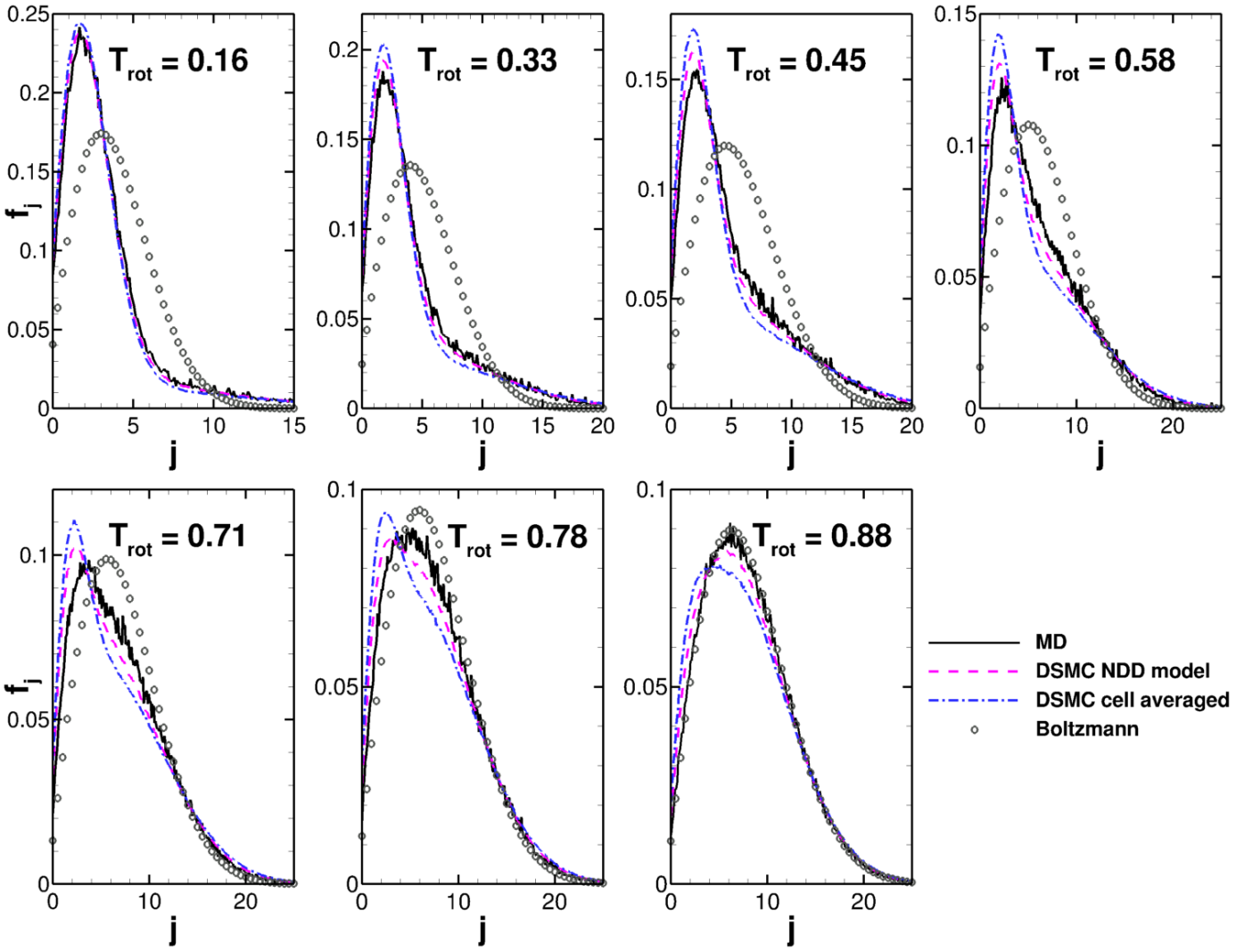


Figure 4.9: Comparison of the rotational energy distribution function at selected points within the $T_1 = 28.3$ K shock wave between DSMC, MD results, and the Boltzmann distribution.

1) The DSMC NDD model developed in this chapter, is based only on collision-quantities (translational and rotational energy of the collision pair ε_t, ε_r) and reduces to a rotational collision number model in the continuum limit (Eq. 4.5), which is applicable for use with the Jeans relaxation equation. The model is formulated based on recent MD simulations of rotational relaxation in nitrogen (Valentini *et al.*, Phys. Fluids 24, 106101 (2012) [87]) and accounts for the dependence of the relaxation rate on the direction to the equilibrium state. This enables a single parameterization of the model to accurately simulate rotational relaxation in nitrogen for both compressing and expanding flows, unlike the widely used Parker model.

2) The DSMC NDD model is simple to implement compared to previously developed phenomenological models, and has good numerical efficiency. More importantly, it accurately reproduces a range of pure MD solutions including zero dimensional isothermal relaxations, one dimensional normal shock wave simulations, and one dimensional expansion simulations [87].

3) In developing the new DSMC NDD model, a general form for the energy distribution function that should be sampled for post-collision states (using the Borgnakke-Larsen approach) is presented. This general formulation ensures detailed balance and equipartition of energy at equilibrium condition for any collision-quantity based DSMC model and also explains the behavior of prior rotational models in the literature [1, 21, 22, 33, 95].

4) During the development of the NDD model, the energy modes of the collision pairs (ε_t, ε_r) and their corresponding degrees of freedom (ε_t, ε_r) are set as general expressions, and are independent of the collision selection procedures used. Thus the NDD model formulation is general to the inelastic collision selection procedure used, which is shown to be a crucial aspect in implementing a DSMC collision model [44, 48, 67].

5) Finally, the increased accuracy of a collision-quantity based model compared to a cell-averaged model is demonstrated by comparing rotational energy distribution functions (rdfs) within a shock wave against a pure MD solution. As shown in the present chapter, the rdfs simulated by the collision-quantity dependent NDD model, is in better agreement with the MD solution, compared to the cell-averaged version of the NDD model.

The NDD model and parameter fitting procedure presented in this chapter is very

general, and provides a framework by which new models can be developed for other species of interest as additional MD data becomes available.

Chapter 5

Consistent Framework for State-to-State Collision Models

5.1 Chapter Introduction

Phenomenological type models have been extensively used in DSMC simulations of highly nonequilibrium flows. However, the collision probability (or cross-section) of the phenomenological models are generally set to reproduce the results of the continuum relaxation equations or rate equations [15, 47, 48], rather than the true collision probability (or cross-section) of each molecular collision process. This is the case even for models using a collision energy dependent probability (or cross-section), such as the widely used total collision energy (TCE) chemical reaction model [15]. For problems where the macroscopic flow quantities are important and have dominant effect on simulation results, the phenomenological model may provide accurate results. However, in situations where the detailed internal energy distribution functions are important, such as for hypersonic dissociating flows, more detailed collision models are desirable. For example, the vibrational energy distribution function may impact the macroscopic chemical reaction rate, which will affect the flow field species concentration, and in turn affect the surface heat transfer rate. Developing advanced DSMC models seems to be the natural choice to more accurately model highly nonequilibrium chemically reacting flows. A state-to-state collision model, where the collision probabilities (or

cross-sections) for collisions between different molecule states are resolved, can naturally be incorporated into the DSMC method, and therefore is the direction to work on [27, 58, 72, 76, 79, 93].

The development and implementation of a state-to-state (state-resolved) collision model in the DSMC method, however, is far from trivial. First, in the DSMC implementation, we need to satisfy the microscopic reversibility relationship at the collision level between each type of collision event and its reverse process; hence satisfying detailed balance at the macroscopic level under equilibrium conditions. With these relationships satisfied, we then could expect that the equipartition of energy between different energy modes in the DSMC simulations are maintained at equilibrium conditions. Second, for a state-to-state collision model, the transport properties are determined by the collision probabilities (or cross-sections), and are not related to any phenomenological viscosity model, such as the VHS model. Thus, careful consideration is required when determining the transport properties of a state-resolved model or when combining a desired transport property model (i.e. viscosity) with state-resolved transition probabilities (or cross-sections) for internal energy states.

In this chapter, we first examine a simple state-to-state model with analytical transition probabilities (or cross-sections), where microscopic reversibility and detailed balance relationships are inherently satisfied [36]. Then, we propose and discuss the detailed implementation of a general rovibrational state-to-state collision model for the DSMC method. The connection between the cross-sections of the full state-to-state model, and the transport properties implied by this model are also discussed. The proposed model is verified using constructed state-resolved collision probabilities for vibrational transitions. The DSMC simulation results are compared with corresponding master equation simulation results to verify the model and its implementation. We then develop a vibrational state-to-state DSMC collision model using the transition probability data provided by the forced harmonic oscillator (FHO) model [3, 68, 77, 78, 82, 96]. Comparison with master equation results using the macroscopic transition rate data obtained from the FHO model is also conducted.

5.2 Examination of a Simple State-to-state DSMC Model with Analytical Transition Probabilities

We start with a simple state-to-state DSMC collision model proposed by Anderson *et al.* [36], where an analytical expression for the transition probability (or cross-section) exists. We refer to this simple model as the Anderson model hereafter. By examining this model, it will show us the important steps we need to consider in formulating and developing a full state-to-state rovibrational collision model, where the state-to-state transition probability, or collision cross-section could be computed through computational chemistry trajectory calculations.

5.2.1 Model formulation for hard sphere molecules

The original Anderson model is for hard sphere molecules. Consider the following type of collision,

$$X_i + X_j \rightarrow X_k + X_l \ , \tag{5.1}$$

where X_i etc., is a molecule with a specific internal energy state, and $i, j, k, l \in 0, ..., v_{max}$ are the molecular internal energy states, with corresponding internal energies $\varepsilon_i, \varepsilon_j, \varepsilon_k, \varepsilon_l$. Here, depending on the description of the internal energy modes (classical, or quantum mechanical description), i, j, k, l could correspond to the rotational or vibrational energy states, or the rovibrational energy states. Thus, the notation is very general.

Denoting the cross-section for a collision specified in Eq. 5.1 as σ_{ij}^{kl}, in the Anderson model [36], we have,

$$\sigma_{ij}^{kl} = \sigma_0 P_{ij}^{kl} \ . \tag{5.2}$$

Here, $\sigma_0 =$ constant, is the hard sphere collision cross-section, while,

$$P_{ij}^{kl} = \frac{g_k g_l (\varepsilon_c - \varepsilon_k - \varepsilon_l)}{\sum_{m,n} g_m g_n (\varepsilon_c - \varepsilon_m - \varepsilon_n)^+} \ , \tag{5.3}$$

and

$$(\varepsilon_c - \varepsilon_m - \varepsilon_n)^+ = \begin{cases} 0 & \text{if} \quad \varepsilon_c - \varepsilon_m - \varepsilon_n < 0 \\ \varepsilon_c - \varepsilon_m - \varepsilon_n & \text{if} \quad \varepsilon_c - \varepsilon_m - \varepsilon_n \geq 0 \end{cases} \tag{5.4}$$

In the above equations, g_k, g_l, g_m, g_n are the degeneracies associated with the internal

energy state k, l, m, n. ε_c is the total collision energy, which is,

$$\varepsilon_c = \varepsilon_t + \varepsilon_i + \varepsilon_j = \varepsilon_t' + \varepsilon_k + \varepsilon_l, \tag{5.5}$$

where ε_t and ε_t' are the relative translational energy of the collision pair before and after the collision.

Next, we conduct a detailed examination of the Anderson model from both the microscopic and macroscopic levels. Using the Anderson model as an example, we introduce several important terminologies, including the microscopic reversibility relationship describing each collision and its reverse collision process, the definition of the transition rate, and detailed balance regarding the transition rate at the equilibrium condition. Finally, the implementation of the Anderson model in the DSMC method will also be discussed.

5.2.1.1 Microscopic reversibility

The microscopic reversibility relation for collisional transition between initial molecular energy states (i, j) and final energy states (k, l) is stated as [80],

$$g_i g_j (\mu g)^2 \sigma_{ij}^{kl} = g_k g_l (\mu' g')^2 \sigma_{kl}^{ij}, \tag{5.6}$$

where $\mu = \frac{m_i m_j}{m_i + m_j}$, and $\mu' = \frac{m_k m_l}{m_k + m_l}$ are the reduced mass of the collision pairs (i, j) and (k, l), respectively, and m_i, m_j, m_k, m_l are the mass of molecules X_i, X_j, X_k, X_l. The quantity $(\mu g)^2$ effectively represents the degeneracy of the translational energy state.

For the above specified excitation/de-excitation collision given in Eq. 5.1, $\mu = \mu'$, and $\varepsilon_t = \mu g^2$, $\varepsilon_t' = \mu' g'^2$. As a result, Eq. 5.6 can be equivalently written as,

$$g_i g_j \varepsilon_t \sigma_{ij}^{kl} = g_k g_l \varepsilon_t' \sigma_{kl}^{ij} \tag{5.7}$$

With the Anderson model defined collision cross-section for state-to-state transitions between state (i, j) and (k, l) (Eq. 5.2), microscopic reversibility will be automatically satisfied. This is verified by substituting the corresponding terms in Eqs. 5.2 – 5.5 into Eq. 5.7.

5.2.1.2 Transition rate determined from state-resolved cross-sections

Denote the total number of collisions per volume (number density of such collisions) for molecules with initial internal energy states (i, j) transiting to final states (k, l) as n_{ij}^{kl}. Following the definition of the reaction rate for chemical reactions, the transition rate is defined according to,

$$\frac{dn_{ij}^{kl}}{dt} = k_{ij}^{kl} n_i n_j \ , \tag{5.8}$$

where n_i, n_j are the number density of molecules in the i, j internal energy states.

Alternatively, according to the kinetic theory definition, the total number of collisions per volume per time for molecules with initial internal energy states (i, j) transiting to final states (k, l) is termed the collision rate, ν_{ij}^{kl}, which can be expressed as,

$$\nu_{ij}^{kl} = n_i n_j \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(\mathbf{C_i}; T) f(\mathbf{C_j}; T) \sigma_{ij}^{kl}(g) |\mathbf{C_i} - \mathbf{C_j}| d\mathbf{C_i} d\mathbf{C_j} = n_i n_j \overline{\sigma_{ij}^{kl} g} \ , \tag{5.9}$$

where we have used $\overline{\sigma_{ij}^{kl} g}$ to denote the integral contained in the above equation. In Eq. 5.9, $\mathbf{C_i} = (C_{i,1}, C_{i,2}, C_{i,3})$ and $\mathbf{C_j} = (C_{j,1}, C_{j,2}, C_{j,3})$ are the molecule velocities of X_i and X_j, respectively; $d\mathbf{C_i} = dC_{i,1} dC_{i,2} dC_{i,3}$, $d\mathbf{C_j} = dC_{j,1} dC_{j,2} dC_{j,3}$; $g = |\mathbf{C_i} - \mathbf{C_j}|$ is the relative collision velocity of collision pair (i, j); $\sigma_{ij}^{kl}(g)$ is the collision cross-section at relative velocity g for the collision given in Eq. 5.1; $f(\mathbf{C_i}; T)$ and $f(\mathbf{C_j}; T)$ are the equilibrium velocity distribution functions at temperature T for molecules X_i and X_j, respectively. Specifically,

$$\begin{aligned} f(\mathbf{C_i}; T) &= \left(\frac{m_i}{2\pi k_B T} \right)^{\frac{3}{2}} \exp\left(-\frac{m_i (C_{i,1}^2 + C_{i,2}^2 + C_{i,3}^2)}{2 k_B T} \right) \\ &= \left(\frac{m_i}{2\pi k_B T} \right)^{\frac{3}{2}} \exp\left(-\frac{m_i c_i^2}{2 k_B T} \right) \ , \end{aligned} \tag{5.10}$$

where $c_i = |\mathbf{C_i}| = (C_{i,1}^2 + C_{i,2}^2 + C_{i,3}^2)^{\frac{1}{2}}$ and m_i is the mass of molecule X_i. The integral in Eq. 5.9 has been written in a simplified form, with $\int_{-\infty}^{+\infty} d\mathbf{C_i}$ representing $\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} dC_{i,1} dC_{i,2} dC_{i,3}$.

By equating Eqs. 5.8 and 5.9, we obtain $\frac{dn_{ij}^{kl}}{dt} = \nu_{ij}^{kl}$. The state-to-state collision

transition rate, k_{ij}^{kl}, is therefore,

$$\begin{aligned} k_{ij}^{kl} &= \overline{\sigma_{ij}^{kl} g} \\ &= \frac{(m_i m_j)^{\frac{3}{2}}}{(2\pi k_B T)^3} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sigma_{ij}^{kl}(g) g \, \exp\left(-\frac{m_i c_i^2 + m_j c_j^2}{2k_B T}\right) d\mathbf{C_i} d\mathbf{C_j} \ . \end{aligned} \tag{5.11}$$

To simplify the above equation, we introduce the center of mass framework. In this framework, the independent variables are $\mathbf{C_m}$ and $\mathbf{C_r}$, which are the center of mass velocity vector and the relative collision velocity vector. The magnitude of these two vectors are, $c_m = |\mathbf{C_m}|$ and $c_r = |\mathbf{C_r}| = g$. By conservation of energy, we have,

$$m_i c_i^2 + m_j c_j^2 = (m_i + m_j) c_m^2 + \mu g^2 \ , \tag{5.12}$$

Through change of variables, we also have,

$$d\mathbf{C_i} d\mathbf{C_j} = d\mathbf{C_m} d\mathbf{C_r} \tag{5.13}$$

$$d\mathbf{C_m} = 4\pi c_m^2 dc_m \tag{5.14}$$

$$d\mathbf{C_r} = 4\pi c_r^2 dc_r = 4\pi g^2 dg \tag{5.15}$$

In order to derive Eq. 5.14, $d\mathbf{C_m} = 4\pi c_m^2 dc_m$, a change of variable from a Cartesian coordinate system $(C_{m,1}, C_{m_2}, C_{m,3})$ to a spherical coordinate system (c_m, θ, ϕ), and the following two expressions have been used,

$$\frac{\partial(C_{m,1}, C_{m,2}, C_{m,3})}{\partial(c_m, \theta, \phi)} = c_m^2 \sin\phi \ , \tag{5.16}$$

$$\begin{aligned} \int_{-\infty}^{\infty} f(\mathbf{C_m}) d\mathbf{C_m} &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(C_{m,1}, C_{m,2}, C_{m,3}) dC_{m,1} dC_{m,2} dC_{m,3} \\ &= \int_0^{\infty} \int_0^{2\pi} \int_0^{\pi} f(c_m, \theta, \phi) c_m^2 \sin\phi dc_m d\theta d\phi \\ &= \int_0^{\infty} 4\pi c_m^2 f(c_m) dc_m \ . \end{aligned} \tag{5.17}$$

Eq. 5.15 is obtained with a similar derivation.

Substituting Eqs. 5.13, 5.14 and 5.15 into Eq. 5.11, we then have,

$$\begin{aligned}
k_{ij}^{kl} =& \frac{(m_i m_j)^{\frac{3}{2}}}{(2\pi k_B T)^3} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sigma_{ij}^{kl}(g) g \exp\left(-\frac{m_i c_i^2 + m_j c_j^2}{2k_B T}\right) d\mathbf{C_i} d\mathbf{C_j} \\
=& \frac{(m_i m_j)^{\frac{3}{2}}}{(2\pi k_B T)^3} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \sigma_{ij}^{kl}(g) g \exp\left(-\frac{(m_i + m_j) c_m^2 + \mu g^2}{2k_B T}\right) d\mathbf{C_m} d\mathbf{C_r} \\
=& \frac{(m_i m_j)^{\frac{3}{2}}}{(2\pi k_B T)^3} \int_{0}^{+\infty} \int_{0}^{+\infty} \sigma_{ij}^{kl}(g) g 4\pi c_m^2 4\pi g^2 \exp\left(-\frac{(m_i + m_j) c_m^2 + \mu g^2}{2k_B T}\right) dc_m dg \\
=& 4\pi \left(\frac{\mu}{2\pi k_B T}\right)^{\frac{3}{2}} \int_{0}^{+\infty} \sigma_{ij}^{kl}(g) g^3 \exp\left(-\frac{\mu g^2}{2k_B T}\right) dg \ .
\end{aligned} \tag{5.18}$$

In the above equation, k_{ij}^{kl} can also be simply expressed as $k_{ij}^{kl} = \int_0^{+\infty} \sigma_{ij}^{kl}(g) g \zeta(g) dg$, with $\zeta(g) = 4\pi \left(\frac{\mu}{2\pi k_B T}\right)^{\frac{3}{2}} g^2 \exp\left(-\frac{\mu g^2}{2k_B T}\right)$, the Maxwell-Boltzmann velocity distribution function for the relative velocity g.

In Eq. 5.18, the transition rate is expressed in terms of the relative collision velocity g. Alternatively, we can express the transition rate using the relative translational energy, $\varepsilon_t = \frac{1}{2}\mu g^2$, as,

$$\begin{aligned}
k_{ij}^{kl} =& 4\pi \left(\frac{\mu}{2\pi k_B T}\right)^{\frac{3}{2}} \int_{0}^{+\infty} \sigma_{ij}^{kl}(g) g^3 \exp\left(-\frac{\mu g^2}{2k_B T}\right) dg \\
=& \frac{1}{\sqrt{\pi \mu}} \left(\frac{2}{k_B T}\right)^{\frac{3}{2}} \int_{0}^{+\infty} \sigma_{ij}^{kl}(\varepsilon_t) \varepsilon_t \exp\left(-\frac{\varepsilon_t}{k_B T}\right) d\varepsilon_t \\
=& A \int_{0}^{+\infty} \sigma_{ij}^{kl}(\varepsilon_t) \varepsilon_t \exp\left(-\frac{\varepsilon_t}{k_B T}\right) d\varepsilon_t \ ,
\end{aligned} \tag{5.19}$$

with constant $A = \frac{1}{\sqrt{\pi \mu}} \left(\frac{2}{k_B T}\right)^{\frac{3}{2}}$.

Thus if the state-resolved cross-sections, σ_{ij}^{kl}, are known (as a function of g or ε_t), then the transition rates k_{ij}^{kl} are determined by Eq. 5.18 or Eq. 5.19. It is again noted that states i, j, k, l are general energy states that could correspond to rotation states, or vibrational states, or combined rovibrational states.

5.2.1.3 Detailed balance

For molecular collision process, detailed balance specifies the constraint that should be satisfied between each collision and its reverse collision. When the microscopic reversibility is satisfied between each collision and its reverse collision process, the transition rate also satisfies the detailed balance relation under thermal equilibrium condition macroscopically.

In collisions between (X_i, X_j) and (X_k, X_l), since the total collision energy, $\varepsilon_c = \varepsilon_t + \varepsilon_i + \varepsilon_j = \varepsilon_t' + \varepsilon_k + \varepsilon_l$, does not change during the collision, we can define,

$$Q(\varepsilon_c) = \sum_{m,n} g_m g_n (\varepsilon_c - \varepsilon_m - \varepsilon_n)^+ \ . \tag{5.20}$$

Then the state-to-state collision cross-section $\sigma_{ij}^{kl}(\varepsilon_t)$ in the Anderson model, for transition $(X_i, X_j) \to (X_k, X_l)$ with initial translational energy ε_t, can be expressed as,

$$\sigma_{ij}^{kl}(\varepsilon_t) = \sigma_0 P_{ij}^{kl} = \sigma_0 \frac{g_k g_l (\varepsilon_c - \varepsilon_k - \varepsilon_l)}{Q(\varepsilon_c)} \tag{5.21}$$

Similarly, the state-to-state collision cross-section $\sigma_{kl}^{ij}(\varepsilon_t')$ in the Anderson model, for transition $(X_k, X_l) \to (X_i, X_j)$ with initial translational energy ε_t', can be expressed as,

$$\sigma_{kl}^{ij}(\varepsilon_t') = \sigma_0 P_{kl}^{ij} = \sigma_0 \frac{g_i g_j (\varepsilon_c - \varepsilon_i - \varepsilon_j)}{Q(\varepsilon_c)} \tag{5.22}$$

With the above definition, the forward transition rate, k_{ij}^{kl}, for transition $(X_i, X_j) \to (X_k, X_l)$ can be written as,

$$\begin{aligned} k_{ij}^{kl}(T) =& A\sigma_0 g_k g_l \int_0^{+\infty} \frac{(\varepsilon_c - \varepsilon_k - \varepsilon_l)\varepsilon_t}{Q(\varepsilon_c)} \exp\left(-\frac{\varepsilon_t}{k_B T}\right) d\varepsilon_t \\ =& A\sigma_0 g_k g_l \int_0^{+\infty} \frac{(\varepsilon_c - \varepsilon_k - \varepsilon_l)(\varepsilon_c - \varepsilon_i - \varepsilon_j)}{Q(\varepsilon_c)} \exp\left(-\frac{\varepsilon_c - \varepsilon_i - \varepsilon_j}{k_B T}\right) d\varepsilon_c \ . \end{aligned} \tag{5.23}$$

Similarly, we have the backward transition rate, k_{kl}^{ij}, for transition $(X_k, X_l) \to (X_i, X_j)$

as,

$$
\begin{aligned}
k_{kl}^{ij}(T) =& A\sigma_0 g_i g_j \int_0^{+\infty} \frac{(\varepsilon_c - \varepsilon_i - \varepsilon_j)\varepsilon_t'}{Q(\varepsilon_c)} \exp\left(-\frac{\varepsilon_t'}{k_B T}\right) d\varepsilon_t' \\
=& A\sigma_0 g_i g_j \int_0^{+\infty} \frac{(\varepsilon_c - \varepsilon_i - \varepsilon_j)(\varepsilon_c - \varepsilon_k - \varepsilon_l)}{Q(\varepsilon_c)} \exp\left(-\frac{\varepsilon_c - \varepsilon_k - \varepsilon_l}{k_B T}\right) d\varepsilon_c \\
=& A\sigma_0 g_i g_j \exp\left(\frac{\varepsilon_k + \varepsilon_l - \varepsilon_i - \varepsilon_j}{k_B T}\right) \\
& \int_0^{+\infty} \frac{(\varepsilon_c - \varepsilon_i - \varepsilon_j)(\varepsilon_c - \varepsilon_k - \varepsilon_l)}{Q(\varepsilon_c)} \exp\left(-\frac{\varepsilon_c - \varepsilon_i - \varepsilon_j}{k_B T}\right) d\varepsilon_c \\
=& \frac{g_i g_j}{g_k g_l} k_{ij}^{kl} \exp\left(\frac{\varepsilon_k + \varepsilon_l - \varepsilon_i - \varepsilon_j}{k_B T}\right) .
\end{aligned}
\tag{5.24}
$$

Eq. 5.24 can be rearranged as,

$$
\frac{k_{ij}^{kl}}{k_{kl}^{ij}} = \frac{g_k g_l}{g_i g_j} \exp\left(-\frac{\varepsilon_k + \varepsilon_l - \varepsilon_i - \varepsilon_j}{k_B T}\right) . \tag{5.25}
$$

By introducing the equilibrium constant $K_{eq}((i,j) \rightarrow (k,l))$ for the collision in Eq. 5.1,

$$
K_{eq}((i,j) \rightarrow (k,l)) = \frac{k_{ij}^{kl}}{k_{kl}^{ij}} , \tag{5.26}
$$

we then have,

$$
K_{eq}((i,j) \rightarrow (k,l)) = \frac{g_k g_l}{g_i g_j} \exp\left(-\frac{\varepsilon_k + \varepsilon_l - \varepsilon_i - \varepsilon_j}{k_B T}\right) . \tag{5.27}
$$

The above equation (Eq. 5.27) is consistent with the definition of the equilibrium constant, and detailed balance is inherently satisfied for the Anderson model, with the specified state-to-state collision cross-sections.

Under equilibrium conditions, the fraction of molecules in internal energy state i is,

$$
f_i = \frac{n_i}{n} = \frac{g_i \exp\left(-\frac{\varepsilon_i}{k_B T}\right)}{\sum_m g_m \exp\left(-\frac{\varepsilon_m}{k_B T}\right)} , \tag{5.28}
$$

where n is the gas number density (number of molecules in the system per volume).

Combining Eqs. 5.26 – 5.28, we then have,

$$\frac{k_{kl}^{ij}}{k_{ij}^{kl}} = \frac{n_i n_j}{n_k n_l} \ , \tag{5.29}$$

or equivalently,

$$k_{ij}^{kl} n_i n_j = k_{kl}^{ij} n_k n_l \ . \tag{5.30}$$

This means that the total number of collisions due to the transition from internal energy states (i, j) to (k, l), is equal to the total number of collisions due to the reverse transition from internal energy states (k, l) to (i, j). This is an alternative explanation for the detailed balance relationship.

5.2.1.4 DSMC implementation of the Anderson model

For molecule-molecule collisions where both molecules' internal energy participate in the energy exchange, as shown above, the transition probability expression in Eq. 5.3 could be used. While for molecule-atom collisions, or molecule-molecule collisions where only one molecule's internal energy participate in the collision process, i.e., for the following collision,

$$X_i + M \rightarrow X_k + M \ , \tag{5.31}$$

with M an atom or molecule (which does not change internal energy states), we can similarly specify the state-to-state collision cross-section for a hard sphere molecule in the Anderson model; specifically,

$$\sigma_i^k = \sigma_0 P_i^k, \tag{5.32}$$

with,

$$P_i^k = \frac{g_k(\varepsilon_c - \varepsilon_k)}{\sum_m g_m(\varepsilon_c - \varepsilon_m)^+} \ , \tag{5.33}$$

and

$$(\varepsilon_c - \varepsilon_m)^+ = \begin{cases} 0 & \text{if} \quad \varepsilon_c - \varepsilon_m < 0 \\ \varepsilon_c - \varepsilon_m & \text{if} \quad \varepsilon_c - \varepsilon_m \geq 0 \end{cases} \ . \tag{5.34}$$

With the above defined state-to-state collision cross-section and transition probability, microscopic reversibility is inherently satisfied for collisions specified by Eq. 5.31. Detailed balance is also satisfied under equilibrium conditions. This can be similarly

proved as shown in section 5.2.1.3.

Since the total collision cross-section with respect to different initial energy states ((i,j) or i) is the same constant σ_0 in the Anderson model, the no-time-counter (NTC) method [15] by Bird can be used without any special modification. The implementation of the Anderson model in DSMC is therefore very straight forward for hard sphere molecules. For a given initial internal energy states, the desired finial energy states will be determined using the acceptance-rejection technique, according to the state-to-state transition probability P_i^k or P_{ij}^{kl} (Eq. 5.3 or Eq. 5.33), depending on the type of molecule collision that performed (Eq. 5.1 or 5.31) in the DSMC simulation.

We take the vibrational state-to-state model as an example. Suppose the bounded simple harmonic oscillator (SHO) model is used to describe the vibrational energy state, with the largest bounded vibrational energy level as v_{max}. For collisions corresponding to Eq. 5.1, the complete DSMC implementation is:

1. The NTC algorithms is used to select collision pairs within a given DSMC collision cell. This includes first calculating the expected number of potential collisions in each collision cell, and then using the acceptance-rejection algorithm to determine which collision pairs actually collide, to be consistent with the overall collision rate. For the Anderson model, this will correspond to a hard sphere collision rate with constant σ_0

2. Determine the actual collision for each collision pair. For a collision pair with initial vibrational energy states (i,j), two random numbers R_1, R_2 uniformly distributed between 0 and 1 are first generated. The potential final vibrational energy states (k,l) are then set as $k = R_1 v_{max}$, $l = R_2 v_{max}$. Next, the probability of transition P_{ij}^{kl} for $(i,j) \rightarrow (k,l)$ is calculated according to the collision energies of the potential collision using Eq. 5.3. P_{ij}^{kl} is then compared with a newly generated random number, R_3, uniformly distributed between 0 and 1.

3. If $P_{ij}^{kl} > R_3$, then the $(i,j) \rightarrow (k,l)$ transition is performed; otherwise, return to step 2, and repeat the acceptance-rejection technique is repeated with two newly generated random numbers.

4. The post collision relative translational energy, ε_t', is calculated from the conservation of energy. The velocities of the two collision particles, are then set following

the hard sphere scattering rule.

The procedures for collisions corresponding to Eq. 5.31 are similar. With the above implementation of the Anderson model in the DSMC method, the state-to-state collision cross-sections and transition rates will be correctly simulated.

One drawback of the Anderson model, however, is the unrealistic state-to-state collision cross-section, σ_{ij}^{kl}, for collisional transitions between different molecular internal energy states. σ_{ij}^{kl} is set analytically, purely under the consideration of satisfying the microscopic reversibility relation, rather than from theoretical or computational calculations of the true molecular collision processes. Since the transition rates are completely determined by σ_{ij}^{kl}, the state-to-state collisional transition rates obtained from this model, and therefore the macroscopic internal energy relaxation rates, may be unrealistic.

To control the macroscopic relaxation rates of the Anderson model and the corresponding rates simulated in the DSMC method, a collision probability P_{rot} or P_{vib} may be used to specify the percentage of collisions that will undergo state-to-state type collisions. Similar to the inelastic collision selection procedure and phenomenological models proposed previously (Chapter 2), each collision pair could first be tested for the possibility of a state-to-state collision using the acceptance-rejection technique with $R_n < P_{rot}$ or $R_n < P_{vib}$. R_n appearing in the inequalities is a random number uniformly distributed between 0 and 1. Only the fractions of collisions that pass the test will undergo state-to-state type collisions of the form $(i, j) \rightarrow (k, l)$; all the other collisions will be elastic, and only involve the change of molecular velocities of the collision pair. The elastic collision could be performed with the widely used VHS or VSS type models.

The collision probability P_{rot} and P_{vib} could be correlated with realistic relaxation rate data, such as the MD data for N_2–N_2 rotational relaxation presented in Ref. 87 (Chapter 4) or the Millikan-White correlation for vibrational relaxation, for example. To do so, a series of relaxation simulations could be performed, similar to those conducted in Chapter 4. The DSMC simulated relaxation results could then be compared with the corresponding results using the Jeans or Landau-Teller equation, the master equation simulations, or the MD results. Through such a detailed comparison, the connection between P_{rot} or P_{vib} and the relaxation rate could then be established specific to the state-resolved model, resulting in a more realistic state-to-state DSMC collision

model. In general, the probability P_{rot} or P_{vib} can depend on temperature, or some combination of collision energy modes, as discussed in Chapters 2 and 4. As already outlined in the Anderson model implementation, the post collision energy in this model is determined through the collision pair selection process directly, instead of using the phenomenological Borgnakke-Larsen model [19]. Detailed balance is inherently satisfied for any temperature or collision energy dependent P_{rot}, P_{vib} model.

To verify the proposed Anderson model implementation in the DSMC method, a series of isothermal vibrational relaxation simulations are performed, where the translational temperature is set as a constant, and the vibrational temperature is relaxing from a different value toward equilibrium. The relaxation simulation results are shown in Fig. 5.1 for the vibration-vibration-translation (VVT) collision process of Eqs. 5.1 and 5.31. Results corresponding to different P_{vib} values to control the relaxation rate, are compared in the figures. The effect of P_{vib} in adjusting the overall relaxation rate is evident from such comparisons. More importantly, the vibrational energy reaches the equilibrium value. This indicates that the DSMC implementation of the Anderson model satisfies the detailed balance relation, and the equipartition of energy between different energy modes is maintained. Furthermore, the vibrational energy distribution function after reaching the equilibrium state is plotted in Fig. 5.2 for the VVT collision process. As can be seen from the figure, the EDF agrees well with the Boltzmann distribution at the corresponding temperature.

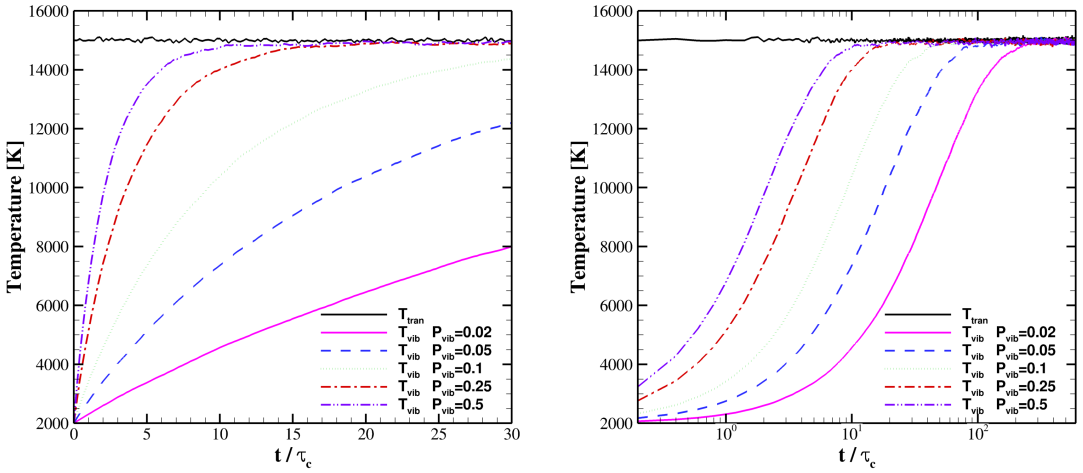
5.2.2 Model extension for variable hard sphere molecules

To better simulate the temperature dependence of viscosity, it is desirable to extend the original Anderson model, and use more realistic molecule interaction model to replace the hard sphere model. A straight forward extension is to change the hard sphere collision cross-section σ_0 in Eq. 5.2 or 5.32 to the variable hard sphere (VHS) collision cross-section $\sigma_{T,VHS}$, and rewrite the state-to-state collision cross-section as,

$$\sigma_{ij}^{kl} = \sigma_{T,VHS}(\varepsilon_t) P_{ij}^{kl} \ , \tag{5.35}$$

and,

$$\sigma_{i}^{k} = \sigma_{T,VHS}(\varepsilon_t) P_{i}^{k} \ . \tag{5.36}$$



(a) Relaxation for short time.

(b) Relaxation for long time.

Figure 5.1: Isothermal vibrational relaxation of the Anderson model for different P_{vib} values.

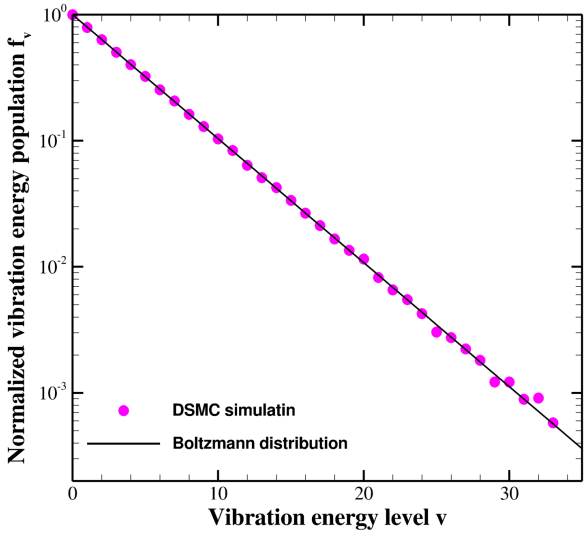


Figure 5.2: The vibrational energy distribution function (EDF) for the isothermal relaxation, after reaching equilibrium temperature of $T = 15000$K.

$\sigma_{T,VHS}$ is a function of the relative collision velocity g or the relative translational energy ε_t. When expressed in terms of ε_t, $\sigma_{T,VHS}$ has the following form,

$$\sigma_{T,VHS}(\varepsilon_t) = \pi d_{ref}^2 \frac{k_B T_{ref}}{\Gamma(\frac{5}{2} - \omega)} \varepsilon_t^{\frac{1}{2}-\omega} \tag{5.37}$$

where d_{ref} is a reference diameter defined at a reference temperature T_{ref}, $\Gamma(\cdot)$ is the Gamma function, and ω is the exponent of the temperature dependent viscosity formulation,

$$\mu(T) = \frac{15\sqrt{\pi m k T_{ref}}}{2\pi d_{ref}^2 (5-2\omega)(7-2\omega)} \left(\frac{T}{T_{ref}}\right)^{\omega} , \tag{5.38}$$

where m is the molecule mass.

This straight forward extension of the Anderson model, however, will lead to DSMC simulation results not satisfying detailed balance and equipartition of energy under equilibrium conditions. To show this, we conduct a series of vibrational relaxation simulations using the Anderson model with different values of P_{vib}. The VHS model parameters used in the simulation are given in Table. 5.1. The results are plotted in Fig. 5.3. As can be clearly seen from the figure, the vibrational and translational temperatures do not come into equilibrium with each other.

Table 5.1: DSMC simulation parameters for the VHS model.

ω	d_{ref} (m)	T_{ref} (K)
0.72	4.17×10^{-10}	273

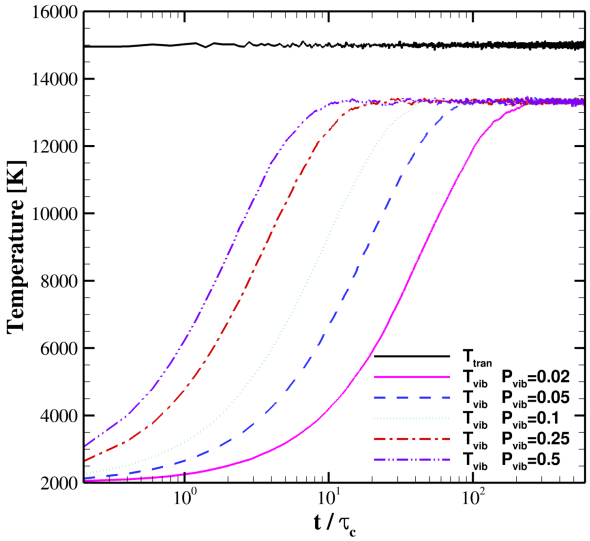


Figure 5.3: Isothermal relaxation of the Anderson model with VHS collision cross-section.

The explanation for the difference in temperature at steady state, is due to the violation of microscopic reversibility, and hence detailed balance at equilibrium conditions in the DSMC simulation. Because of this, some vibrational levels will be over-populated compared to their equilibrium populations, while some vibrational levels will be under-populated. This can be clearly seen by examining the steady state vibrational energy distribution (EDF) function resulting from the simulation. The vibrational EDF after reaching steady state is plotted in Fig. 5.4 for the VVT collision process. Overall, the result shown in the figure is consistent with the analysis.

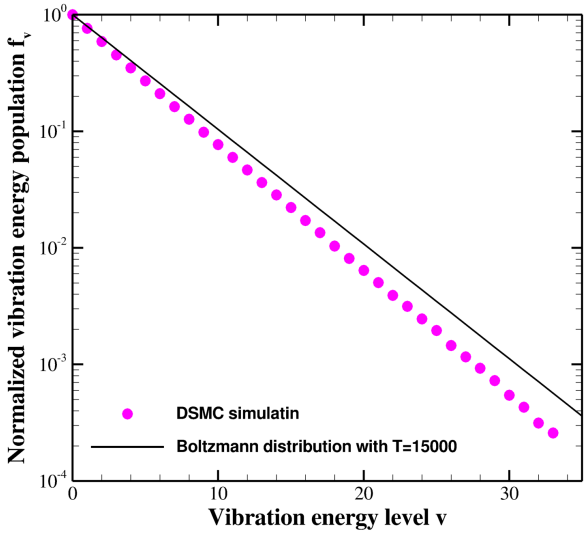


Figure 5.4: The steady state vibrational energy distribution function (EDF) for isothermal relaxation, using the Anderson model with VHS cross-section.

To arrive at a more rigorous explanation, we perform the following mathematical analysis taking the molecule-molecule collision as an example. According to the specified state-to-state collision cross-section and transition probability (Eqs. 5.2 and 5.3), we have,

$$\sigma_{ij}^{kl}(\varepsilon_t) = \sigma_{T,VHS}(\varepsilon_t) P_{ij}^{kl} = \sigma_{T,VHS}(\varepsilon_t) \frac{g_k g_l (\varepsilon_c - \varepsilon_k - \varepsilon_l)}{Q(\varepsilon_c)} = \sigma_{T,VHS}(\varepsilon_t) \frac{g_k g_l \varepsilon_t'}{Q(\varepsilon_c)} \ , \quad (5.39)$$

and,

$$\sigma_{kl}^{ij}(\varepsilon_t') = \sigma_{T,VHS}(\varepsilon_t')P_{kl}^{ij} = \sigma_{T,VHS}(\varepsilon_t')\frac{g_i g_j(\varepsilon_c - \varepsilon_i - \varepsilon_j)}{Q(\varepsilon_c)} = \sigma_{T,VHS}(\varepsilon_t')\frac{g_i g_j \varepsilon_t}{Q(\varepsilon_c)} \ , \quad (5.40)$$

for the $(i,j) \rightarrow (k,l)$ and $(k,l) \rightarrow (i,j)$ transitions with relative translational energies ε_t and ε_t', respectively. We then define,

$$I_1 = g_i g_j \varepsilon_t \sigma_{ij}^{kl} = g_i g_j g_k g_l \frac{\varepsilon_t \varepsilon_t'}{Q(\varepsilon_c)} \sigma_{T,VHS}(\varepsilon_t) \ , \quad (5.41)$$

and,

$$I_2 = g_k g_l \varepsilon_t' \sigma_{kl}^{ij} = g_i g_j g_k g_l \frac{\varepsilon_t \varepsilon_t'}{Q(\varepsilon_c)} \sigma_{T,VHS}(\varepsilon_t') \ . \quad (5.42)$$

This leads to,

$$\frac{I_1}{I_2} = \frac{\sigma_{T,VHS}(\varepsilon_t)}{\sigma_{T,VHS}(\varepsilon_t')} \quad (5.43)$$

By comparing Eqs. 5.41, 5.42 and 5.43 with Eq. 5.7 (the microscopic reversibility relationship), we can clearly see that this cannot be satisfied in general, since the VHS cross-section term $\sigma_{T,VHS}$ in I_1 and I_2 corresponds to different translational energies (ε_t and ε_t'). Consequently, if Eq. 5.35 or 5.36 is used in the DSMC method, the state-to-state transition rate obtained from such cross-sections will not satisfy detailed balance relation. The DSMC simulation results will not achieve the equipartition of energy between different energy modes, and hence a difference in steady state translation and vibrational temperatures will result.

To fix the problem, the probabilities P_{ij}^{kl} and P_i^k specified by Eqs. 5.3 and 5.33 must be modified. We propose the following modification to the original expressions in Eqs. 5.3 and 5.33. For molecule-molecule collisions, P_{ij}^{kl} is modified to,

$$\hat{P}_{ij}^{kl} = \frac{g_k g_l (\varepsilon_c - \varepsilon_k - \varepsilon_l)^{\frac{3}{2}-\omega}}{\sum_{m,n} g_m g_n (\varepsilon_c - \varepsilon_m - \varepsilon_n)^*} \ , \quad (5.44)$$

with,

$$(\varepsilon_c - \varepsilon_m - \varepsilon_n)^* = \begin{cases} 0 & \text{if} \quad \varepsilon_c - \varepsilon_m - \varepsilon_n < 0 \\ (\varepsilon_c - \varepsilon_m - \varepsilon_n)^{\frac{3}{2}-\omega} & \text{if} \quad \varepsilon_c - \varepsilon_m - \varepsilon_n \geq 0 \end{cases} \ . \quad (5.45)$$

For molecule-atom collisions, P_i^k is modified to,

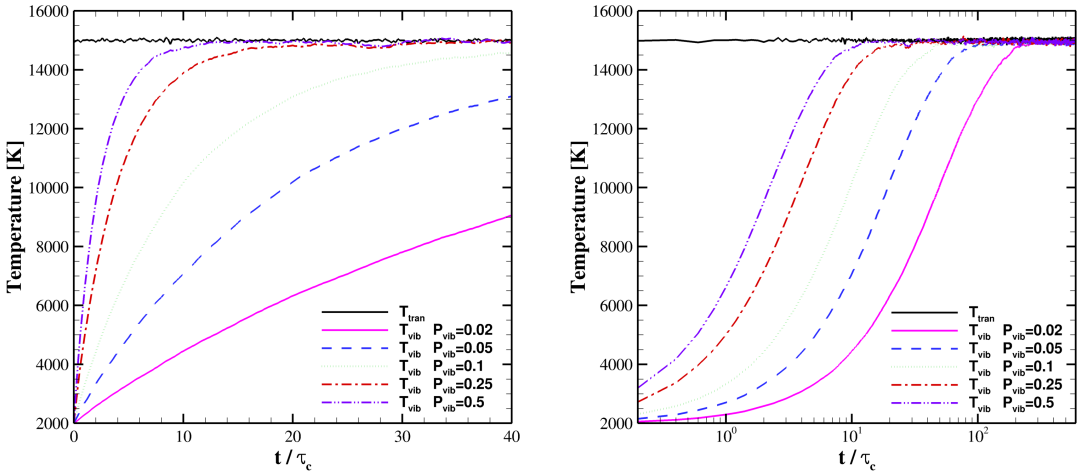
$$\hat{P}_i^k = \frac{g_k(\varepsilon_c - \varepsilon_k)^{\frac{3}{2}-\omega}}{\sum_m g_m(\varepsilon_c - \varepsilon_m)^*} \ , \tag{5.46}$$

with,

$$(\varepsilon_c - \varepsilon_m)^* = \begin{cases} 0 & \text{if} \quad \varepsilon_c - \varepsilon_m < 0 \\ (\varepsilon_c - \varepsilon_m)^{\frac{3}{2}-\omega} & \text{if} \quad \varepsilon_c - \varepsilon_m \geq 0 \end{cases} \ . \tag{5.47}$$

With this modification, the microscopic reversibility will be automatically satisfied when using the VHS collision cross-section, resulting in the the state-to-state transition rate achieving the detailed balance relation. Using the DSMC implementation procedures outlined in section 5.2.1.4, the simulated transition rate will also satisfy detailed balance and the simulation results will achieve equipartition of energy between different energy modes at steady state. This is confirmed by actual DSMC tests. A series of vibrational relaxation simulation are performed for both the molecule-molecule and molecule-atom collisions, and the results are plotted in Fig. 5.5. From the figure, it can be easily seen that the vibrational and translational temperatures are in equilibrium with each other. To further verify detailed balance, the vibrational energy distribution function (EDF) after reaching steady state is plotted in Fig. 5.6 for the VVT collision process. Again, the EDF agrees well with the Boltzmann distribution at the corresponding temperature.

We can again use the probability (or fraction) of state-to-state collisions (P_{rot} and P_{vib}) to control the simulated relaxation rate as discussed in section 5.2.1.4. P_{rot} and P_{vib} could be correlated with existing experimental, theoretical, or computational results regarding the relaxation rate (or collision number) for each type of relaxation process, through extensive comparison of the DSMC simulation results with those of the Jeans or Landau-Teller equation, master equation simulation results, or Molecular Dynamics results, for example. However, our objectives were to use the Anderson model to demonstrate the correct implementation of a state-resolved model, demonstrate an extension of the model, and most importantly identify the key aspects crucial for formulating a full rovibrational state-to-state DSMC collision model. As a result, we will not proceed further along the direction of the Anderson model. The correlation of the rotational or vibrational relaxation rate predicted by the Anderson model together with a probability



(a) Relaxation for short time.

(b) Relaxation for long time.

Figure 5.5: Isothermal vibrational relaxation of the Anderson model for different P_{vib} values with VHS model.

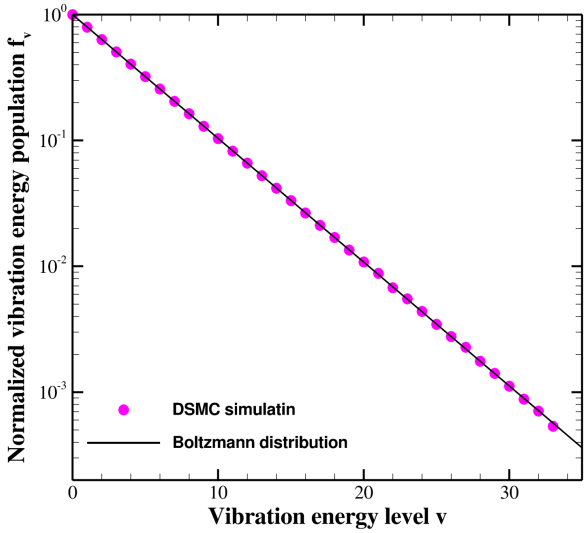


Figure 5.6: The vibrational energy distribution function (EDF) for the isothermal relaxation, after reaching equilibrium temperature $T = 15000$K.

P_{rot} or P_{vib}, may indeed result in rotational or vibrational relaxation models superior to the previously developed phenomenological models. Such a model may also result in more realistic energy distribution functions, however, this requires further investigation,

and is not the focus of the current chapter.

5.2.3 Implications for general state-to-state DSMC collision models

Through detailed analysis of the Anderson model and the proposed VHS-modification, we clearly observe the importance of satisfying the microscopic reversibility relation for the state-to-state transition probabilities or collision cross-sections. The simulations in sections 5.2.1 and 5.2.2 focus on the vibrational energy mode, however, the analysis is general for any internal energy exchange process. Whether the state-to-state data considers only vibrational state-to-state transition, or full rovibrational state-to-state transition, the microscopic reversibility should be accordingly satisfied. If this is accomplished, then the detailed balance relation will also be satisfied. Combined with the correct procedures and implementation of the state-to-state collision data within the DSMC method, we can expect to obtain a fully consistent state-to-sate DSMC collision model. The development a full state-to-state DSMC collision model, however, requires careful considerations to maintain all of these requirements. Specifically, we need to consider the followings:

1. To start with, the state-to-state collision cross-section database should (ideally) satisfy microscopic reversibility, however, in practice this may not be the case. When such a relationship is not satisfied, modification to the database may be required. When the database is such that microscopic reversibility is satisfied between each collision and its reverse collision process, the transition rate will also satisfy the detailed balance relation under thermal equilibrium condition macroscopically.

2. The state-to-state cross-sections must be implemented within the DSMC method such that the DSMC simulated transition rates reproduce the desired rates (input from the data base). This includes consistent implementation of collision rates (cross-sections), pair selection procedures, and acceptance-rejection algorithms.

3. Finally, the calculation of desired collision rates in the DSMC method, the selection of potential collision pairs, and the procedure to perform actual collision processes should be computationally efficient and able to capture rare transition events.

5.3 Development of a General State-to-state DSMC Collision Model

By carefully considering the microscopic reversibility and detailed balance relationships for general rovibrational state-to-state transition probability data, and satisfying them exactly, we propose a consistent framework to develop state-to-state collision models in the DSMC method. We start from the fundamental collision process, build up the framework step by step, and then propose the full procedures to develop such final collision model. For the proposed model implementation, we will first examine it by utilizing arbitrarily constructed transition probabilities (cross-section) data. In section 5.4, we will use the proposed framework to implement a vibration state-to-state DSMC collision model utilizing the FHO model transition probability data [3].

The detailed collision process is modeled by the macroscopic transition rate in the continuum description, while it is described in terms of transition probability or cross-section from the molecular point of view. There is a correspondence between the transition rate and cross-section in continuum and molecular descriptions. Specifically, this correspondence should be maintained between the continuum simulation, such as the master equation simulation, and the DSMC simulation using the state-to-state transition cross-section, to ensure the consistency between the two methods. To verify this, the DSMC simulation results from the state-to-state model will be compared with master equation simulation results using the corresponding temperature dependent transition rate data.

Due to the complexity of the detailed collision process, there is no simple analytical expression to link the state-to-state model cross-section to the transport coefficients. This is different compared to phenomenological collision models, such as the VHS or VSS model [15], where the transport coefficients (such as for viscosity) can be analytically expressed. Thus, the connection between the state-to-state collision model cross-section and the corresponding transport properties predicted by such model will also be discussed.

5.3.1 State-to-state collision data from computational chemistry

Ideally, the collision process could be fully simulated in a deterministic manner. The starting point is the molecule interaction potential, or the potential energy surface (PES), which could be obtained from high level quantum chemistry calculations, such as the density functional theory. With the PES in hand, we could perform a detailed trajectory calculation or Molecular Dynamics simulation to simulate the molecule collision event.

For trajectory calculation, there are quasi-classical trajectory (QCT) calculation and classical trajectory calculation (CTC) depending on the description of the molecule, either quantum description or classical description, and the ways molecule internal energy states are sampled [31, 37, 38, 55]. Regardless of the slight difference in the two trajectory calculation methods, the input parameters in the trajectory simulation is a specific molecule collision configuration, corresponding to certain translational and internal energy. The simulated molecules are then allowed to interact with each other according to the classical dynamics based on the PES. By simulating all the possible molecule collision events, we could then obtain the detailed informations of the molecule collisions.

5.3.1.1 Molecules without internal structure/energy

We first consider the two body collision problem of molecules without internal structure/energy. In other words, the molecule can be treated as an atom. Denote the two collision pairs as M_1 and M_2, with mass m_1 and m_2, respectively. The reduced mass of the collision pair is $\mu = \frac{m_1 m_2}{m_1+m_2}$, and the relative collision velocity is g. Consider the collision process in the center of mass framework, that is, describe the collision as a molecule with a reduced mass of μ interacting with a central potential. The parameters that describe the collision process are:

1. impact factor, b, $0 \leq b \leq \infty$.
2. relative velocity of the collision pair, g, $0 \leq g \leq \infty$.
3. angle ε, the reference angle of the plane of trajectory with respect to a reference plane, $0 \leq \varepsilon \leq 2\pi$.

Equivalently, the collision process can be described by alternate parameters:

1. relative velocity of the collision pair, g, $0 \leq g \leq \infty$.

2. deflection angle of the collision trajectory after collision, χ, $0 \leq \chi \leq \pi$.

3. angle ε, the reference angle of the plane of trajectory with respect to a reference plane, $0 \leq \varepsilon \leq 2\pi$.

Thus, we have two sets of parameters to describe the equivalent collision process:

1. (b, g, ε);

2. (g, χ, ε).

The deflection angle χ in the second description, can be expressed as a function of b and g: $\chi = \chi(b, g)$. This explicitly accounts for the influence of impact factor and relative collision velocity on the deflection angle (alternatively, the impact factor can be expressed as a function of relative collision velocity and deflection angel, i.e., $b = b(g, \chi)$). These relations link the two sets of collision parameters, and the two approaches are equivalent. The first description focuses on describing the molecule states before the collision, while the second description focuses on describing the collision outcomes after the collision.

We can therefore have two definitions of the differential cross-section according to the two descriptions: (1) differential cross-section $= \sigma(g, \chi)d\Omega$, or (2) differential cross-section $= bdbd\varepsilon$. The two descriptions are equivalent to each other,

$$\sigma(g, \chi)d\Omega = bdbd\varepsilon \tag{5.48}$$

with $\sigma(g, \chi)$, the collision cross-section and $d\Omega$, the differential solid angle $d\Omega = sin\chi d\chi d\varepsilon$. The collision cross-section can be expressed in terms of b and χ as,

$$\sigma(g, \chi) = \frac{b}{sin\chi}\left|\frac{\partial b}{\partial \chi}\right| \tag{5.49}$$

Then the total collision cross-section corresponding to the relative collision velocity g is,

$$\sigma_T(g) = \int_0^{\pi}\int_0^{2\pi} \sigma(g, \chi)d\Omega = \int_0^{\pi}\int_0^{2\pi} \sigma(g, \chi)sin\chi d\chi d\varepsilon \tag{5.50}$$

Equivalently, the total collision cross-section at the relative collision velocity g can be calculated as,

$$\sigma_T(g) = \int_0^{\infty} \int_0^{2\pi} bdbd\varepsilon \tag{5.51}$$

In the above description, the molecule's state after the collision is not explicitly given, since according to the classical collision dynamics, the post collision state is determined by the molecule's initial states, and is deterministic. A more complete description of the total collision cross-section should include the post collision relative velocity g' as a parameter, such that $\sigma_T(g) = \sigma_T(g \to g')$. However, due to the deterministic nature of the collision, it is simply represented as $\sigma_T(g)$.

5.3.1.2 Molecules with internal structure/energy

Next, we consider the two body collision problem of molecules with internal structure/energy. In this case, we first consider the molecule-atom collision of the following type,

$$N_2(j, v) + M \to N_2(j', v') + M \ , \tag{5.52}$$

and take a N_2 molecule as an example for the analysis. The analysis here is general, and can be extended to molecule-molecule collisions of the form,

$$A_2(j_1, v_1) + B_2(j_2, v_2) \to A_2(j_1', v_1') + B_2(j_2', v_2') \ , \tag{5.53}$$

where A_2, B_2 are two molecules with initial internal energy states $(j_1, v_1), (j_2, v_2)$ and final energy states $(j_1', v_1'), (j_2', v_2')$, respectively. Here j_1, j_2, j_1', j_2' represent the rotational quantum numbers, while v_1, v_2, v_1', v_2' represent the vibrational quantum numbers.

In Eq. 5.52, M is either an atom or a molecule that does not change internal energy during the collision, and (j, v), (j', v') are the rotational and vibrational quantum states of the N_2 molecule before and after the collision event, respectively. Denote the internal energy change process as $\Gamma = (j, v \to j', v')$. For the collision in Eq. 5.52, suppose the transition (or reaction) probability from the initial states (j, v) to the final states (j', v') is $P_R(b, \varepsilon; g, \Gamma)$. Here the subscript "R" denotes "reaction", as we could consider all such transitions as "reactions". Here, we enforce the following identity for the transition

probability $P_R(b, \varepsilon; g, \Gamma)$,

$$\sum P_R(b, \varepsilon; g, \Gamma) = \sum_{j', v'} P_R(b, \varepsilon; g, j, v \rightarrow j', v') = 1 \ , \text{ for any } g, j, v \ . \tag{5.54}$$

Consider a specific collision with initial impact factor value b and relative collision velocity g. After the collision process, the differential collision cross-section ($bdbd\varepsilon$), will be mapped to a differential cross-section $I_R(\chi, \varepsilon; g, \Gamma)d\Omega$, with probability $P_R(b, \varepsilon; g, \Gamma)$. Here, $I_R(\chi, \varepsilon; g, \Gamma)$ is a similarly defined state-specific collision cross-section for the transition $(g, \Gamma) = (g, j, v \rightarrow j', v')$ corresponding to the scattering angle χ and reference angle ε. This means that we have,

$$I_R(\chi, \varepsilon; g, \Gamma)d\Omega = P_R(b, \varepsilon; g, \Gamma)bdbd\varepsilon \tag{5.55}$$

The above equation simply denotes the mapping between the differential area before and after the collision.

The reactive collision cross-section for the transition $(g, j, v \rightarrow j', v')$ is therefore,

$$\sigma_R(g, \Gamma) = \int \int I_R(\chi, \varepsilon; g, \Gamma)d\Omega = \int_0^{2\pi} \int_0^{2\pi} I_R(\chi, \varepsilon; g, \Gamma)sin\chi d\chi d\varepsilon \ , \tag{5.56}$$

or alternatively,

$$\sigma_R(g, \Gamma) = \int \int P_R(b, \varepsilon; g, \Gamma)bdbd\varepsilon = \int_0^{\infty} \int_0^{2\pi} P_R(b, \varepsilon; g, \Gamma)bdbd\varepsilon \ . \tag{5.57}$$

For the above integration, we must choose an upper limit for the integration (b_{max}), otherwise, the integration does not converge. Although $P_R(b, \varepsilon; g, \Gamma)$ depends on b and ε, in practice, we can define a conservative cut-off distance b_{max} and a corresponding maximum collision cross-section πb_{max}^2. In this case, the velocity and internal energy states dependent collision cross-section is,

$$\sigma_R(g, \Gamma) = \int_0^{\infty} \int_0^{2\pi} P_R(b, \varepsilon; g, \Gamma)bdbd\varepsilon \approx \pi b_{max}^2 \int_0^{b_{max}} \int_0^{2\pi} \frac{P_R(b, \varepsilon; g, \Gamma)}{\pi b_{max}^2} bdbd\varepsilon \ . \tag{5.58}$$

This allows us to write the state specific collision cross-section for transition $(g, \Gamma) =$

$(g, j, v \rightarrow j', v')$ as,

$$\sigma_R(g, \Gamma) = \sigma_R(g, j, v \rightarrow j', v') = \pi b_{max}^2 P_R^0(g, \Gamma) \ , \tag{5.59}$$

where $P_R^0(g, \Gamma)$ is,

$$P_R^0(g, \Gamma) = \int_0^{2\pi} \frac{P_R(b, \varepsilon; g, \Gamma)}{\pi b_{max}^2} b db d\varepsilon \ , \tag{5.60}$$

and the dependence on b and ε has been intergrated out of Eq. 5.58, for example by a large number of trajectory calculations sampling $0 < b < b_{max}$ and $0 < \varepsilon < 2\pi$. The integrated result is represented by $P_R^0(g, \Gamma)$ in Eq. 5.59. Eq. 5.59 is the most fundamental expression for all state-to-state transition cross-sections from which a full state-resolved model can be developed.

The collision cross-section for the transition from a specific initial quantum states (j, v) to all final quantum states is then,

$$\sigma_R(g, j, v) = \sum_{j', v'} \sigma_R(g, j, v \rightarrow j', v') = \sum_{j', v'} \pi b_{max}^2 P_R^0(g, j, v \rightarrow j', v') \ . \tag{5.61}$$

We can therefore define the state-to-state transition probability as,

$$P_R(g, j, v \rightarrow j', v') = \frac{\sigma_R(g, j, v \rightarrow j', v')}{\sigma_R(g, j, v)} \ . \tag{5.62}$$

5.3.2 General determination of state-to-state collision cross-sections

In some situations, the full database for state-to-state collision cross-sections (Eq. 5.59) may be available (through computational chemisty methods for example). However, in some situations, the full state-to-state collision cross-section data may not be available, is incomplete, or is inconsistent in some manner. For example, in the forced harmonic oscillator (FHO) model, only the transition probabilities are given and information regarding the collision cross-section is not specified. It is also possible, that only excitation collision cross-sections are given, in which case we need to define the de-excitation collision cross-sections based on microscopic reversibility. It is also possible that statistical scatter (resulting from a finite number of trajectory calculations) in the cross-section database and/or the binning of classical energies into quantum states may result in

subtle inconsistencies in a set of cross-sections with respect to microscopic reversibility. In this section, we generally determine a consistent set of state-to-state collision cross-sections for such situations. In general, we propose the following procedure to arrive at a consistent set of state-to-state collision cross-sections:

1. We start from an energy dependent collision cross-section, $\sigma_R^{(0)}(g, j, v)$. This could result from a computational chemistry database ($\sigma_R(g, j, v)$ as discussed in the previous section), or be imposed by a phenomenological model such as the VHS model, where $\sigma_R^{(0)}(g, j, v) = \sigma_R(g)$ is assumed independent of the energy states (j, v) and is consistent with some viscosity law (Eqs. 5.37 and 5.38). We also have a set of transition probabilities $P_R^{(0)}(g, j, v \to j', v')$. This could result from computational chemistry as discussed in the previous section (Eq. 5.62), or be determined by some theoretical analysis (such as in the FHO model).

2. In the k-th step, we define the excitation transition cross-sections as,

$$\sigma_R^{(k)}(g, j, v \to j', v') = \sigma_R^{(k-1)}(g, j, v) P_R^{(k-1)}(g, j, v \to j', v'), \quad \text{for } \varepsilon(j, v) \leq \varepsilon(j', v') \tag{5.63}$$

   while calculating the de-excitation transition cross-sections as,

$$\sigma_R^{(k)}(g, j, v \to j', v') = \frac{\sigma_R^{(k)}(g', j', v' \to j, v) g_n(j', v') \varepsilon_t'}{g_n(j, v) \varepsilon_t}, \quad \text{for } \varepsilon(j, v) < \varepsilon(j', v') \tag{5.64}$$

   from microscopic reversibility. Here, $g_n(j, v)$ and $g_n(j', v')$ are the degeneracies of the (j, v) and (j', v') quantum states, respectively; $\varepsilon_t = \frac{1}{2}\mu g^2$ and $\varepsilon_t' = \frac{1}{2}\mu g'^2$ are the relative translational energies corresponding to (j, v) and (j', v') energy states, respectively, which satisfy the conservation of energy equation,

$$\varepsilon_t + \varepsilon(j, v) = \varepsilon_t' + \varepsilon(j', v') \tag{5.65}$$

3. From Eqs. 5.63 and 5.64, all the state-to-state collision cross-sections $\sigma_R^{(k)}(g, j, v \to j', v')$ $(\forall g, j, v, j', v')$ are defined for the k-th step. We can then re-calculate the state-specific collision cross-section, $\sigma_R^{(k)}(g, j, v)$, for all transitions with an initial

states (j, v) as,

$$\sigma_R^{(k)}(g, j, v) = \sum_{j', v'} \sigma_R^{(k)}(g, j, v \to j', v'). \tag{5.66}$$

This result can be compared to the corresponding value in the (k-1)-th step, $\sigma_R^{(k-1)}(g, j, v)$. If the original state-to-state collision cross-sections and probabilities were initially fully consistent, then the result of Eq. 5.66 will be identical to the previous value $\sigma_R^{(k-1)}(g, j, v)$ (specifically for k=1). However, in most cases the results will not be exactly equal, and may differer substantially. This indicates that microscopic reversibility is not satisfied by the initial set of cross-sections.

4. The transition probabilities from the previous iteration can then be updated according to $\sigma_R^{(k)}(g, j, v \to j', v')$ and $\sigma_R^{(k)}(g, j, v)$ as,

$$P_R^{(k)}(g, j, v \to j', v') = \frac{\sigma_R^{(k)}(g, j, v \to j', v')}{\sigma_R^{(k)}(g, j, v)} \ . \tag{5.67}$$

After 2 steps, we stop at k = 2, and obtained a fully consistent set of state-to-state cross-sections.

5. At the end of the process, we have $\sigma_R(g, j, v \to j', v') = \sigma_R^{(k_0)}(g, j, v \to j', v')$ and $\sigma_R(g, j, v) = \sigma_R^{(k_0)}(g, j, v)$, with $k_0 = 2$. For the final consistent set of state-to-state collision cross-sections, microscopic reversibility is automatically satisfied between initial and final states: (j, v) (j', v'),

$$\sigma_R(g, j, v \to j', v') g_n(j, v) \varepsilon_t = \sigma_R(g', j', v' \to j, v) g_n(j', v') \varepsilon_t' \ . \tag{5.68}$$

This can be easily checked according to Eqs. 5.63 and 5.64.

As a remark, if the state-to-state transition probability database is complete and fully consistent in itself, then from the state-to-state transition probabilities, $P_R(\varepsilon, g, b, \Gamma)$, or the differential collision cross-sections, $I_R(\chi, \varepsilon, g, \Gamma)$, we could get the full consistent

state-to-state collision cross-sections as,

$$\begin{aligned} \sigma_R(g,\Gamma) &= \int_0^{\pi}\int_0^{2\pi} I_R(\chi,\varepsilon,g,\Gamma) sin\chi d\chi d\varepsilon \\ \text{or} \quad &= \int_0^{\infty}\int_0^{2\pi} P_R(\varepsilon,g,b,\Gamma) b db d\varepsilon. \end{aligned} \tag{5.69}$$

These cross-sections naturally satisfy microscopic reversibility. As a result, the above iteration process may be not necessary in this situation.

Finally, when the full set of state-to-state collision cross-sections are determined (through the iteration process or directly from computational chemistry) the cross-sections can be further integrated to arrive at transition rates (according to Eq. 5.18 or Eq. 5.19), which will satisfy detailed balance under equilibrium condition, i.e.,

$$\frac{k_f(\Gamma,T)}{k_f(\Gamma',T)} = \frac{g_n(j',v')}{g_n(j,v)} \exp\left(-\frac{\varepsilon(j',v')-\varepsilon(j,v)}{k_B T}\right) . \tag{5.70}$$

Here, $k_f(\Gamma,T)$ and $k_f(\Gamma',T)$ are the state-to-state transition rates at temperature T, with $\Gamma = (j, v \rightarrow j', v')$ and $\Gamma' = (j', v' \rightarrow j, v)$.

5.3.3 Detailed balance at equilibrium

We have in hand the following two equations,

1. The definition of the differential collision cross-section:

$$I_R(\chi,\varepsilon,g,\Gamma) d\Omega = P_R(\varepsilon,g,b,\Gamma) b db d\varepsilon \tag{5.71}$$

2. The microscopic reversibility relationship:

$$g_n(j,v)\varepsilon_t I_R(\chi,\varepsilon,g,\Gamma) = g_n(j',v')\varepsilon_t' I_R(\chi,\varepsilon,g',\Gamma') \tag{5.72}$$

From the definition of the state-to-state collision cross-section and the above two equations, we have,

$$\begin{aligned}
\sigma_R(g,\Gamma) &= \int_0^\infty \int_0^{2\pi} P_R(\varepsilon,g,b,\Gamma) b db d\varepsilon \\
&= \int_0^\infty \int_0^{2\pi} P_R(\varepsilon,g',b,\Gamma') \frac{g_n(j',v')\varepsilon_t'}{g_n(j,v)\varepsilon_t} b db d\varepsilon \\
&= \frac{g_n(j',v')\varepsilon_t'}{g_n(j,v)\varepsilon_t} \int_0^\infty \int_0^{2\pi} P_R(\varepsilon,g',b,\Gamma') b db d\varepsilon \\
&= \frac{g_n(j',v')\varepsilon_t'}{g_n(j,v)\varepsilon_t} \sigma_R(g',\Gamma') \ ,
\end{aligned} \tag{5.73}$$

where we have used the definition of the differential collision cross-section, and the microscopic reversibility relation,

$$\begin{aligned}
P_R(\varepsilon,g,b,\Gamma) &= \frac{I_R(\chi,\varepsilon,g,\Gamma) d\Omega}{b db d\varepsilon} \\
&= \frac{I_R(\chi,\varepsilon,g',\Gamma') g_n(j',v')\varepsilon_t' d\Omega}{g_n(j,v)\varepsilon_t b db d\varepsilon} \\
&= I_R(\chi,\varepsilon,g',\Gamma') d\Omega \frac{g_n(j',v')\varepsilon_t'}{g_n(j,v)\varepsilon_t b db d\varepsilon} \quad . \\
&= P_R(\varepsilon',g',b,\Gamma') b db d\varepsilon \frac{g_n(j',v')\varepsilon_t'}{g_n(j,v)\varepsilon_t b db d\varepsilon} \\
&= P_R(\varepsilon',g',b,\Gamma') \frac{g_n(j',v')\varepsilon_t'}{g_n(j,v)\varepsilon_t}
\end{aligned} \tag{5.74}$$

Hence, we have,

$$g_n(j,v)\varepsilon_t \sigma_R(g,\Gamma) = g_n(j',v')\varepsilon_t' \sigma_R(g',\Gamma') \ . \tag{5.75}$$

Or writing Γ, Γ' explicitly, we have,

$$g_n(j,v)\varepsilon_t \sigma_R(g,j,v \to j',v') = g_n(j',v')\varepsilon_t' \sigma_R(g',j',v' \to j,v). \tag{5.76}$$

Therefore, the collision cross-section defined in Eqs. 5.63 and 5.64 satisfies the microscopic reversibility. As a result, the detailed balance relationship is also satisfied, and the proof is given below.

Denote $k_f(j, v \rightarrow j', v'; T)$ and $k_f(j', v' \rightarrow j, v; T)$ as the state-to-state transition rate at temperature T of the forward and backward transition between (j, v) and (j', v'). From the definition of the transition rate, $k_f(j, v \rightarrow j', v'; T)$,

$$k_f(j, v \rightarrow j', v'; T) = A \int_0^\infty \sigma_R(g, j, v \rightarrow j', v') \varepsilon_t \exp\left(-\varepsilon_t / k_B T\right) d\varepsilon_t \ , \tag{5.77}$$

we then have,

$$\begin{aligned}
& k_f(j, v \rightarrow j', v'; T) \\
=& A \int_0^\infty \sigma_R(g, j, v \rightarrow j', v') \varepsilon_t \exp\left(-\varepsilon_t / k_B T\right) d\varepsilon_t \\
=& A \int_0^\infty \sigma_R(g, j', v' \rightarrow j, v) \varepsilon_t' \frac{g_n(j', v')}{g_n(j, v)} \exp\left(-(\varepsilon_t' + \varepsilon(j', v') - \varepsilon(j, v))/k_B T\right) d\varepsilon_t' \\
=& \frac{g_n(j', v')}{g_n(j, v)} \exp\left(-(\varepsilon(j', v') - \varepsilon(j, v))/k_B T\right) \\
& A \int_0^\infty \sigma_R(g, j', v' \rightarrow j, v) \varepsilon_t' \exp\left(-\varepsilon_t'/k_B T\right) d\varepsilon_t' \\
=& \frac{g_n(j', v')}{g_n(j, v)} \exp\left(-(\varepsilon(j', v') - \varepsilon(j, v))/k_B T\right) k_f(j', v' \rightarrow j, v; T)
\end{aligned} \tag{5.78}$$

This is equivalent to,

$$\frac{k_f(j, v \rightarrow j', v'; T)}{k_f(j', v' \rightarrow j, v; T)} = \frac{g_n(j', v')}{g_n(j, v)} \exp\left(-\Delta\varepsilon / k_B T\right) \ , \tag{5.79}$$

where $\Delta\varepsilon = \varepsilon(j', v') - \varepsilon(j, v)$, which can be rewritten as,

$$\begin{aligned}
& g_n(j, v) \exp\left(-\varepsilon(j, v)/k_B T\right) k_f(j, v \rightarrow j', v'; T) \\
=& g_n(j', v') \exp\left(-\varepsilon(j', v')/k_B T\right) k_f(j', v' \rightarrow j, v; T) \ .
\end{aligned} \tag{5.80}$$

Thus if the state-to-state collision cross-sections satisfy microscopic reversibility, then the transition rates will satisfy the detailed balance in equilibrium. Then if the state-to-state model is implemented correctly within the DSMC method, DSMC simulations will satisfy these relationships as well, including the equipartition of energy.

5.3.4 Implementation of the state-to-state model in DSMC

Now, the question is how to correctly implement the state-to-state collision data in the DSMC method, in order to accurately simulate the desired transition behavior. The state-to-state DSMC collision model implementation should consider the following aspects: select the correct number of collision pairs (collision rate); ensure that the collisions result in the correct collision outcomes; $\sigma_R(g, j, v \rightarrow j', v')$ or $P_R(g, j, v \rightarrow j', v')$ is correctly simulated. To achieve all these aspects, we divide the problems into several parts:

1. Collision pair selections / collision rate specification. Since in general, no analytical expression for $\sigma(g, j, v \rightarrow j', v')$ exists, $\sigma(g, j, v \rightarrow j', v')$ is expressed as a discrete function in the DSMC simulation.

2. Final transition state selections. For an initial energy level (j, v) and relative collision velocity g, select the final energy level (j', v') according to $\sigma_R(g, j, v \rightarrow j', v')$.

3. Redistribution of the post-collision translational energy between the two collision partners.

4. For an incomplete state-to-state transition rate model (such as the FHO model), some internal energy transfer processes may not be specified by the state-to-state transition rate data, and therefore need to be modeled. This could be achieved by using the phenomenological collision number model (Z_{rot}, Z_{vib}, etc.). For the FHO model, we can consider the state-to-state transition of vibrational energy mode first, and then for each collision pair, apply the usual phenomenological rotational collision number model. However, it is noted that when combining models in such a manner, the rotatioin model would likely need to be re-parametrized using the new combined-model framework.

5.3.4.1 Collision rate determination

We first consider the collision rate and number of collisions for an equilibrium gas with a number density n, temperature T, and a volume V. We also have the following,

- $\sigma_R(g, j, v \rightarrow j', v')$, cross-section for the transition from (j, v) to (j', v') energy level;

- $\sigma_R(g, j, v)$, cross-section for the transition from (j, v) energy level, with $\sigma_R(g, j, v) = \sum_{j',v'} \sigma_R(g, j, v \rightarrow j', v')$;

According to the definition of the collision rate (number of collisions per volume per time for one molecule), we have the collision rate for the $(j, v \rightarrow j', v')$ collision (for one molecule at the initial (j, v) state) corresponding to temperature T as, $\nu(j, v \rightarrow j', v'; T)$,

$$\begin{aligned} \nu(j, v \rightarrow j', v'; T) =& n \int_0^{+\infty} \sigma_R(g, j, v \rightarrow j', v') g f(g) dg \\ =& n \overline{\sigma_R(g, j, v \rightarrow j', v') g} \ . \end{aligned} \tag{5.81}$$

The total collision rate at temperature T is then,

$$\begin{aligned} \nu(T) =& \sum_{j,v} f(j, v; T) \sum_{j',v'} \nu(j, v \rightarrow j', v'; T) \\ =& n \sum_{j,v} f(j, v; T) \sum_{j',v'} \overline{\sigma_R(g, j, v \rightarrow j', v') g} \\ =& n \sum_{j,v} f(j, v; T) \overline{\sigma_R(g, j, v) g} \end{aligned} \tag{5.82}$$

Here, we have $\overline{\sigma_R(g, j, v \rightarrow j', v') g} = \int_0^{+\infty} \sigma_R(g, j, v \rightarrow j', v') f(g) g dg$, and $\overline{\sigma_R(g, j, v) g} = \int_0^{+\infty} \sigma_R(g, j, v) f(g) g dg$. $f(j, v; T)$ is the population of (j, v) quantum energy states in the system corresponding to temperature T (the probability that the molecule will be in the (j, v) energy states). $f(j, v; T)$ is included in the total collision rate calculation to account for the population of (j, v) state at equilibrium. At equilibrium condition, we have,

$$f(j, v) = g_n(j, v) \exp\left(-\frac{\varepsilon(j, v)}{k_B T}\right) / Q \ , \tag{5.83}$$

where $g_n(j, v)$ is the degeneracies of the (j, v) quantum state, $\varepsilon(j, v)$ is the energy of the (j, v) quantum energy state, k_B is the Boltzmann constant, and Q is the partition function,

$$Q = \sum_{j,v} g_n(j, v) \exp\left(-\frac{\varepsilon(j, v)}{k_B T}\right) \ . \tag{5.84}$$

Total number of collisions, $M(j, v \to j', v')$, in a cell with volume V, during the time step Δt, for the $(j, v \to j', v')$ transition at temperature T is then,

$$\begin{aligned} M(j, v \to j', v') =& \frac{1}{2}\nu(j, v \to j', v')\Delta t V n f(j, v) \\ =& \frac{1}{2}n^2 V \Delta t f(j, v)\overline{\sigma_R(g, j, v \to j', v')g} \end{aligned} \tag{5.85}$$

Note, here we have dropped the dependence of $N(j, v \to j', v')$ on temperature T for convenience. Also note, the symmetric factor $\frac{1}{2}$ in the above equation applies to single species gas collision (for example N_2-N_2 collision with only one N_2 molecule change internal energy). For collision of different gas species, the symmetric factor $\frac{1}{2}$ will drop. This holds for the following derivations. The total number of collisions, $N(j, v)$, in a cell with volume V, during the time step Δt, for the (j, v) energy states at temperature T is then,

$$M(j, v) = \sum_{j', v'} M(j, v \to j', v') \tag{5.86}$$

The total number of collisions, M, in a cell with volume V, during the time step Δt at temperature T is then,

$$M = \sum_{j, v} M(j, v) = \sum_{j, v} \sum_{j', v'} M(j, v \to j', v') \tag{5.87}$$

An alternate (but equivalent) definition of M is as follows. In a cell with volume V, the number of collisions in time Δt for particles in the velocity range $(g, g + dg)$ and at energy state (j, v) are,

$$dM = \frac{1}{2}\underbrace{nVf(j, v)f(g)dg}_{\text{I}} \cdot \underbrace{\underbrace{\sigma_R(g, j, v)g\Delta t}_{\text{II}} \cdot n}_{\text{III}}, \tag{5.88}$$

where term I is the # of particles in [(g, g+dg), (j, v)] states, term II is the volume swept by one molecule, and term III is the # of particles in the swept volume. Then,

the total number of collisions is,

$$M = \sum_{j,v} \int_0^\infty dM = \sum_{j,v} \int_0^\infty \frac{1}{2} n^2 V \Delta t f(j,v) \sigma_R(g,j,v) f(g) g dg \ , \tag{5.89}$$

which is equivalent to Eq. 5.87.

Next, consider the situation in a DSMC collision cell of volume V. Supposed the particle weight of each DSMC simulation particle is W_p, we then have the number of simulation particles, N, in a collision cell as,

$$N = nV/W_p \ . \tag{5.90}$$

We then incorporate the no time counter (NTC) algorithm in the state-to-state collision model, to simulate the desired transition rate (collision rate, or collision numbers) in the DSMC method. Specifically, we propose three NTC type algorithms to simulate the state-to-state collision rate, and collision numbers within DSMC. For simplicity, we only consider a vibrational state-to-state collision model here, and for the molecule-atom collision process. However, the algorithms can be easily extended to a full rovibrational state-to-state model for either molecule-molecule or molecule-atom collision processes.

We call the three NTC algorithms state-to-state NTC (STS-NTC) algorithms. The major difference between the NTC algorithm for the phenomenological VHS or VSS type model, and the STS-NTC algorithms for the state-to-state DSMC collision model, lies in the calculation of the collision rate and the number of collisions. In the phenomenological model, only one collision rate (that is the total collision rate) and number of collisions (in the DSMC simulation) is required. The collision rate and the number of collisions are linked to the total collision cross-section $\sigma_T(g)$, either VHS or VSS cross-section. However, for the state-to-state DSMC collision model, we need to calculate the state-to-state collision rate and the number of collisions. Depending on the number of transitions allowable, there may be thousands or millions of such quantities. Each collision rate is linked to a state-to-state cross-section $\sigma_R(g,j,v \rightarrow j',v')$. More importantly, we need to correctly simulate the number of collision events for each type of state-to-state transition, to ensure the correct state-to-state transition or collision rate are simulated.

Each of the three STS-NTC algorithms proposed here, can simulate all the state-to-state transition simultaneously. Specifically, during each collision phase of the DSMC method, algorithm 1 explicitly calculate the number of potential collisions for each state-to-state transition, and pick the desirable amount of pairs lie in the specific energy level to perform the actual collision. Each transition process is explicitly accounted for in algorithm 1. Through this, the correct amount of collisions for each transition process is simulated. Due to the finite amount of particles in a DSMC simulation cell, it is often impossible to simulate each state-to-state transition *each time step*. This may cause problem in the actual DSMC simulations. Since the number of available particles in the high energy states is very small, the collision event involving high energy states may not be simulated in every DSMC time step, resulting in not satisfying the detailed balance and hence equipartition of energy. Algorithm 2 and 3 improve on this. Specifically for algorithm 2, since the final collision states are selected randomly (uniformly) from all the allowable energy states, each transition has an equal chance to be selected in every DSMC time step; transitions that are not simulated in one time step, have the chance to be simulated in a later time step. Similar to algorithm 2, algorithm 3 further improves on this.

In the following part, i, j represent the initial and final vibrational energy states, respectively; $\tilde{N}_{ij}$, $\tilde{N}_i$, $\tilde{N}$ represent the maximum expected number of collisions in a DSMC collision cell with volume V during a single time step Δt. The three algorithms calculate these quantities as,

- Algorithm 1:

$$\tilde{N}_{ij} = \frac{1}{2} N_i \bar{N} (\sigma_R(i \to j) g)_{\max} \Delta t W_p / V \tag{5.91}$$

- Algorithm 2:

$$\tilde{N}_i = \frac{1}{2} N_i \bar{N} (\sigma_R^{(0)}(i) g)_{\max} (v_{\max} + 1) \Delta t W_p / V \tag{5.92}$$

- Algorithm 3:

$$\tilde{N} = \frac{1}{2} N \bar{N} (\sigma_R^{(0)} g)_{\max} (v_{\max} + 1) \Delta t W_p / V \tag{5.93}$$

In the above equation, N_i is the number of simulation particles in the i-th vibrational energy level. $\bar{N}$ is the average number of particles in the cell. With

$$(\sigma_R(i \to j)g)_{\max} = \max_g(\sigma_R(g, i \to j)g) \tag{5.94}$$

$$(\sigma_R^{(0)}(i)g)_{\max} = \max_j(\sigma_R(i \to j)g)_{\max} \ , \tag{5.95}$$

and,

$$(\sigma_R^{(0)}g)_{\max} = \max_i(\sigma_R^{(0)}(i)g)_{\max} \tag{5.96}$$

The demonstrative model used in section 5.3.5 includes 48 vibrational levels (bounded anharmonic oscillator model for vibrational energy) and ignores rotational energy. This means there are $(48)^2$ stored values of $\sigma_R(i \to j)g)_{\max}$ for algorithm 1, 48 stored values of $(\sigma_R^{(0)}(i)g)_{\max}$ for algorithm 2, and one stored value of $(\sigma_R^{(0)}g)_{\max}$ for algorithm 3. Algorithm 1 is similar as proposed by Willauer and Varghese [92] for a rotational state-to-state DSMC model.

5.3.4.2 Collision pair selection and final transition state selection

We then select desired number of collision pairs for each transition process, in accordance with the three algorithms (Eqs. 5.91 – 5.91) for $\tilde{N}_{ij}$, $\tilde{N}_i$, $\tilde{N}$. For a collision pair selected for collision, the final state (energy level) is determined by the acceptance-rejection technique, with $R_1, R_2, ..., ...$ random numbers, uniformly distributed between 0 and 1. Corresponding to the three algorithms, we first determine the transition probability. The probabilities for the three algorithms are,

- Algorithms 1, $P_{i\to j} = \frac{\sigma_R(g,i\to j)g}{(\sigma_R(i\to j)g)_{\max}}$;
- Algorithms 2, $P_{i\to j} = \frac{\sigma_R(g,i\to j)g}{(\sigma_R^{(0)}(i)g)_{\max}}$, with $j = \text{Int}((v_{\max} + 1)R_1)$;
- Algorithms 3, $P_{i\to j} = \frac{\sigma_R(g,i\to j)g}{(\sigma_R^{(0)}g)_{\max}}$, with $j = \text{Int}((v_{\max} + 1)R_1)$.

For a selected collision pair, the transition $(i \to j)$ is then tested with $P_{i\to j} > R_2$. If the inequality holds, then the transition is allowed, and we perform the actual collision process; Otherwise, the collision pair does not undergo a collision.

5.3.4.3 Post collision velocity determination

The post collision velocities of the two collision particles is determined according to the hard sphere scattering. This is the same as determined by the VHS or VSS type model.

5.3.5 Verification of the DSMC implementations

We use zero dimensional relaxation simulations to verify the three algorithms for the state-to-state DSMC collision model implementations. We use constructed collision probability to verify the implementation. Only vibrational state-to-state transition is considered here. The transition probability are set for excitation collisions, with,

$$P_R(g, i \to j) = a_{ij} \quad \text{for} \quad i \leq j \ , \tag{5.97}$$

while all the de-excitation transition probabilities are set according to microscopic reversibility according to section 5.3.2. a_{ij} in the above equation is set as constant, specifically,

$$a_{ij} = \begin{cases} a & \text{if} \quad i = j \\ b & \text{if} \quad i < j \end{cases} \ , \tag{5.98}$$

with $a = 0.7, b = 0.01$. Such a choice of state-to-state transition probabilities is qualitatively accurate, with elastic transition probability greater than the inelastic transition probability. The collision cross-section are determined as according to section 5.3.2, with $\sigma_R(g) = \sigma_{T,VHS}(g)$, the VHS collision cross-section.

Bounded anharmonic oscillator (AHO) model is used to describe the quantized vibrational energy states, where only vibrational states with energy less than the dissociation energy of the molecule are considered. Specifically for a vibrational states i, its energy is calculated as,

$$\varepsilon_i = i k_B \theta_v (1 - \chi_e (i + 1)), \tag{5.99}$$

where θ_v is the vibrational temperature constant, and χ_e is the anharmonicity of the molecule. The nitrogen molecule properties are used in the simulations, with $\theta_v = 3397.3$K, $\chi_e = 0.006125$, and the dissociation energy $\theta_D = 113269.0$K (expressed in terms of temperature) [51]. There are 48 bounded vibration levels with energy less than dissociation energy, θ_D, and the largest bounded vibrational level is $v_{\max} = 47$. The

degeneracies of nitrogen molecule is $g_n(i) = 1$. For the AHO model, the population of each vibration level, $f(i)$, is,

$$f(i) = \exp\left(-\frac{\varepsilon(i)}{k_B T}\right)/Q \ , \tag{5.100}$$

where Q is the partition function,

$$Q = \sum_{i=0}^{v_{\max}} \exp\left(-\frac{\varepsilon(i)}{k_B T}\right) \ . \tag{5.101}$$

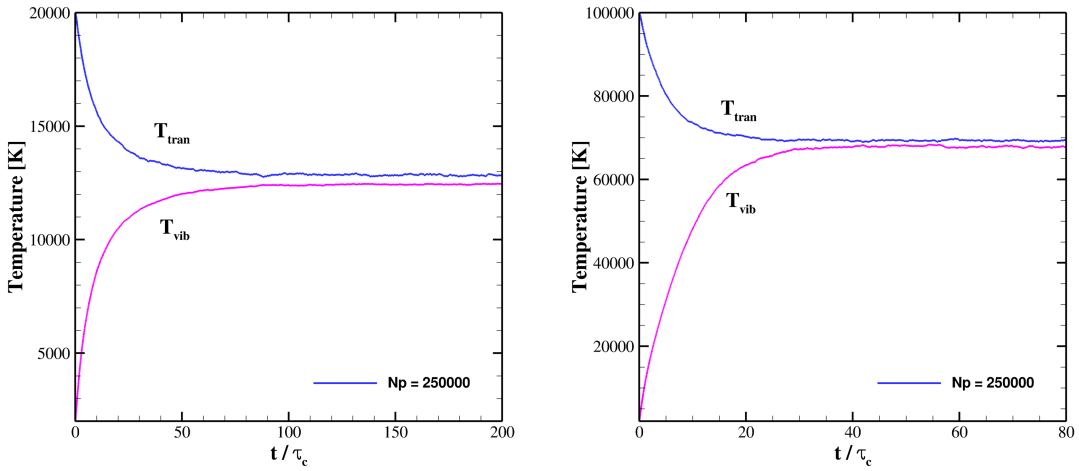
The average vibrational energy, $< \varepsilon(T) >$, corresponding to temperature T for the AHO model is,

$$< \varepsilon(T) >= \sum_{i=0}^{v_{\max}} f(i)\varepsilon(i) \ . \tag{5.102}$$

A loop-up table is then generated to store the correspondence between the vibrational temperature and the average vibrational energy. To obtain the vibrational temperature for AHO model in the DSMC simulation, the sampled average vibrational energy is compared with the vibration energy values in the look-up table, to determine the appropriate vibrational temperature.

Zero dimensional adiabatic simulation of nitrogen are conducted with different initial translational and vibrational temperature. Specifically, $T_{\text{tran}}(0) = 20000\text{K}$, while $T_{\text{vib}}(0) = 2000\text{K}$. The density of the flow is set at $0.1kg/m^3$. For algorithms 1, the results are shown in Fig. 5.7 and Fig. 5.8 with different initial $(\sigma_R(i \to j)g)_{\max}$ values. Specifically, in Fig. 5.7, the initial $(\sigma_R(i \to j)g)_{\max}$ value is $(\sigma_R(i \to j)g)_{\max} = 1.0\times10^{-18}m^3/s$, while in Fig. 5.8, the initial $(\sigma_R(i \to j)g)_{\max}$ value is $(\sigma_R(i \to j)g)_{\max} = 1.0\times10^{-18}m^3/s$. The $(\sigma_R(i \to j)g)_{\max}$ value is updated during the simulation. There are two observations from the results. First, with small initial $(\sigma_R(i \to j)g)_{\max}$, there is a small discrepancy between translational and vibrational temperature at low temperature range (Fig. 5.7(a)), while reduced at high temperature range (Fig. 5.7(b)). Second, by using large initial $(\sigma_R(i \to j)g)_{\max}$ value, the discrepancy between translational and vibrational temperature is negligible even at low temperature range. This is due to the fact that in algorithm 1, during each time step, every transition must be simulated. However, due to the low population of high vibrational energy levels, some transition may not be

simulated, even using a very large number of particles per cell. This is evident from the figure, where a very large amount of particles are used ($N_p = 250000$). With a large $(\sigma_R(i \to j)g)_{\max}$ value, collision probabilities for collisions with small cross-sections are artificially augmented, leading to higher chance of collision. This is useful in simulating collisions with small probabilities, and to ensure detailed balance and equipartition of energy in the DSMC simulation. This can be be clearly seen from the comparison between corresponding results in Fig. 5.7 and 5.8. When temperature increases, high vibrational energy levels are more populated, and have higher chance of collision. algorithm 1 is then able to achieve detailed balance even at small $(\sigma_R(i \to j)g)_{\max}$. This can be seen from the comparison between Fig. 5.7(a) and 5.7(b).

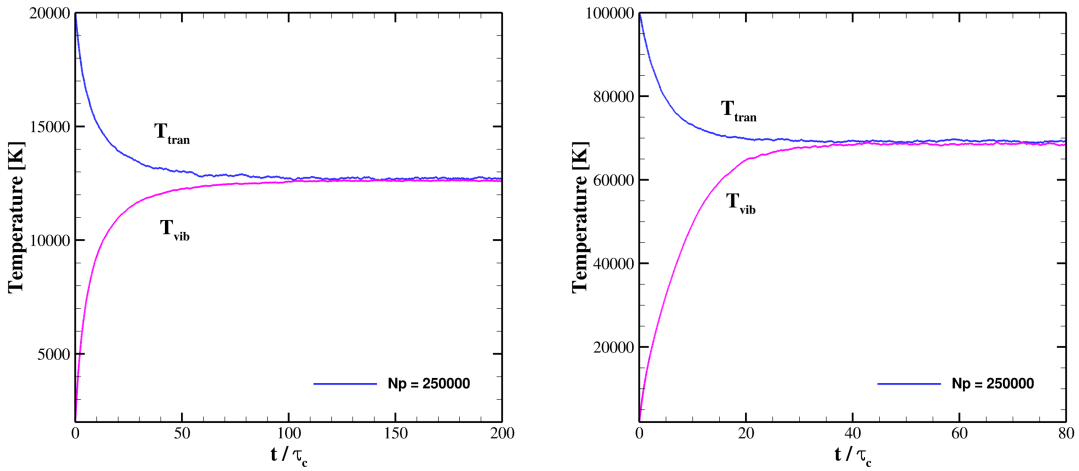


(a) Low equilibrium temperature

(b) High equilibrium temperature

Figure 5.7: Adiabatic vibrational relaxation of the state-to-state DSMC collision model with constructed transition probability. The results are shown for different equilibrium temperature values using algorithm 1. The number of simulation particles in the collision cell is $N_p = 250000$, with initial $(\sigma_R(i \to j)g)_{\max} = 1.0 \times 10^{-18} m^3/s$.

Algorithms 2 and 3 overcome such difficulties, and can successfully simulate the relaxation process even with very small number of particles per cell. Where from Figs. 5.9 and 5.10, we can clearly see this. Algorithm 2 and 3 are accurate even with a particle of 100, although there will be large statistical scattering. Such ability is very important for simulating more complex flow fields, where in each collision cell, there may be around 20 particles per cell. Overall, algorithm 3 is slightly better than algorithm 2, due to



(a) Low equilibrium temperature

(b) High equilibrium temperature

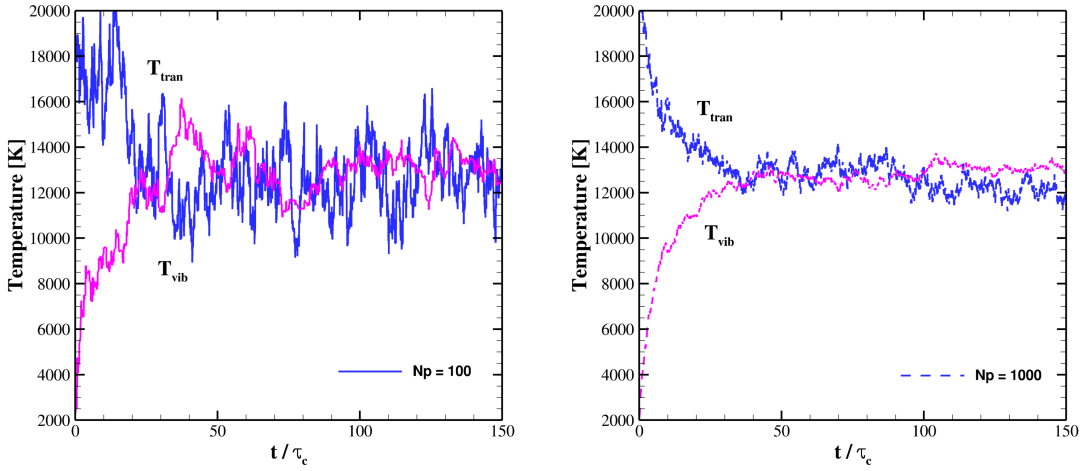
Figure 5.8: Adiabatic vibrational relaxation of the state-to-state DSMC collision model with constructed transition probability. The results are shown for different equilibrium temperature values using algorithm 1. The number of simulation particles in the collision cell is $N_p = 250000$, with initial $(\sigma_R(i \to j)g)_{\max} = 1.0 \times 10^{-16} m^3/s$.

the lesser constraint on the number of particles per simulation cell. The vibrational energy distribution function (EDF) is also plotted in Fig. 5.11 for an adiabatic relaxation simulation. The temperature in the figure corresponds to $T = 20000$K. The agreement with the Boltzmann distribution is very good.

Finally, as a note, algorithm 1, 2 and 3 generally work well, but it may be possible to improve them (especially algorithm 2 and 3) further.

To end the general state-to-state model framework, we conclude that by simulating all the possible molecule collision events, we could obtain the detailed informations of the molecule collisions. Ideally this could be represented as the state-to-state collision cross-section. Specifically for the molecule-atom collision, this could be the full database of state-to-state collision cross-section $\sigma_R(g, j, v \to j', v')$ for all the allowable transitions between the initial and final energy states (j, v) and (j', v') states. The cross-section could then be integrated under equilibrium conditions to obtain the state-to-state transition rate data, that is $k_R(j, v, \to j', v'; T)$ for all the allowable transitions $(j, v) \to (j', v')$, with transition rate for each transition expressed as a function of temperature T.

In essence, such a database should satisfy the microscopic reversibility and detailed

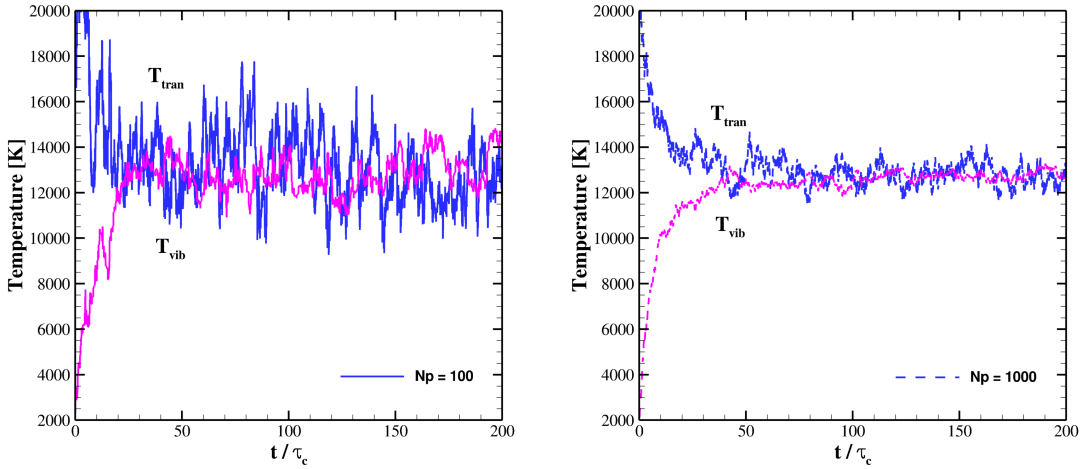


(a) $N_p = 100$ (b) $N_p = 1000$

(c) $N_p = 10000$ (d) $N_p = 250000$

Figure 5.9: Adiabatic vibrational relaxation of the state-to-state DSMC collision model with constructed transition probability. The results are shown for different number of simulation particles in the collision cell, using algorithm 2.

balance relations. The transition rate could then be used in continuum simulation, such as the computational fluid dynamics (CFD) simulation, while the collision cross-section data could be used in particle simulation, such as the DSMC method. Then, there is a correspondence between the continuum and molecule simulations. We could expect that the simulation results for problems suitable for both the CFD and DSMC methods



(a) $N_p = 100$ (b) $N_p = 1000$

(c) $N_p = 10000$ (d) $N_p = 250000$

Figure 5.10: Adiabatic vibrational relaxation of the state-to-state DSMC collision model with constructed transition probability. The results are shown for different number of simulation particles in the collision cell, using algorithm 3.

be consistent with each other. By keeping the correspondence in mind, we could then develop reduced order models for both the continuum and molecule simulation methods, where the number of total internal energy states considered may be reduced. This can be done through grouping internal energy states based on their energies. Regardless of whether each energy state corresponding to one rotational and vibrational quantum

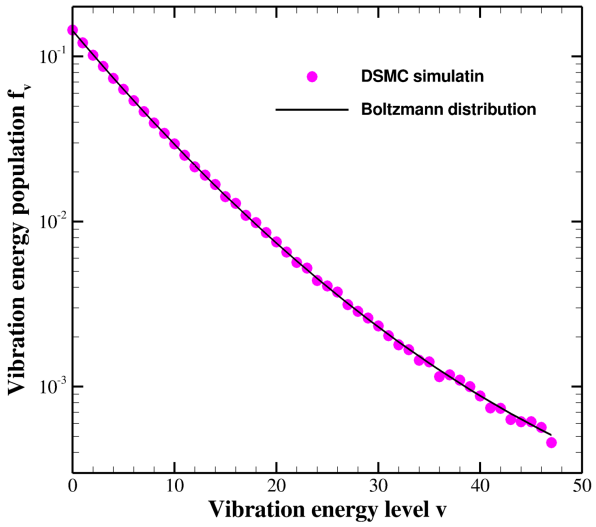


Figure 5.11: The steady state vibrational energy distribution function (EDF) for isothermal relaxation, using the state-to-state DSMC collision model with constructed transition probability. The equilibrium temperature is $T = 20000$K.

number, or represent the average energy state of the group, the analysis in the following sections still holds. We could then arrive at consistent models for both the continuum and molecule simulations.

5.4 Implementation of the Forced Harmonic Oscillator Model in DSMC

The energy transfer of molecules through collisions are generally controlled by the following two processes [3], the vibration-translation (V-T) processes,

$$AB(i) + M \rightleftharpoons AB(f) + M \tag{5.103}$$

and the vibration-vibration-translation (V-V-T) processes

$$AB(i_1) + CD(i_2) \rightleftharpoons AB(f_1) + CD(f_2) \tag{5.104}$$

where, AB, CD represents diatomic molecules and M an atom, i, f, i_1, i_2, f_1, f_2 are vibrational quantum numbers.

The FHO model is a state-to-state based model where the transition between different quantum vibrational states are derived based on a harmonic oscillator with an exponential force acting on it, modeling the interaction between the two colliding particles. The state-to-state transition probabilities in FHO model are expressed as follows for the V-T and V-V-T processes [3, 78, 82, 96].

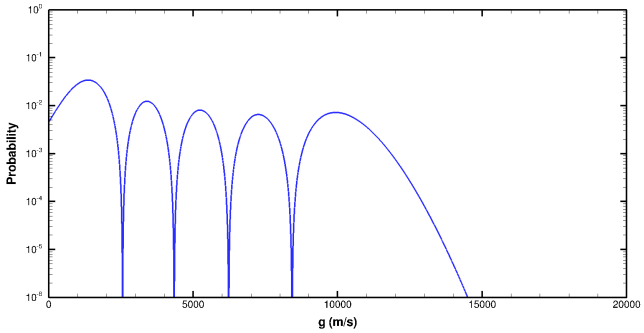
$$P_{VT}(i \rightarrow f, \epsilon) = i!f!\epsilon^{i+f} \exp(-\epsilon) \mid \sum_{r=0}^{n} \frac{(-1)^r}{r!(i-r)!(f-r)!\epsilon^r} \mid^2 \quad , \tag{5.105}$$

and,

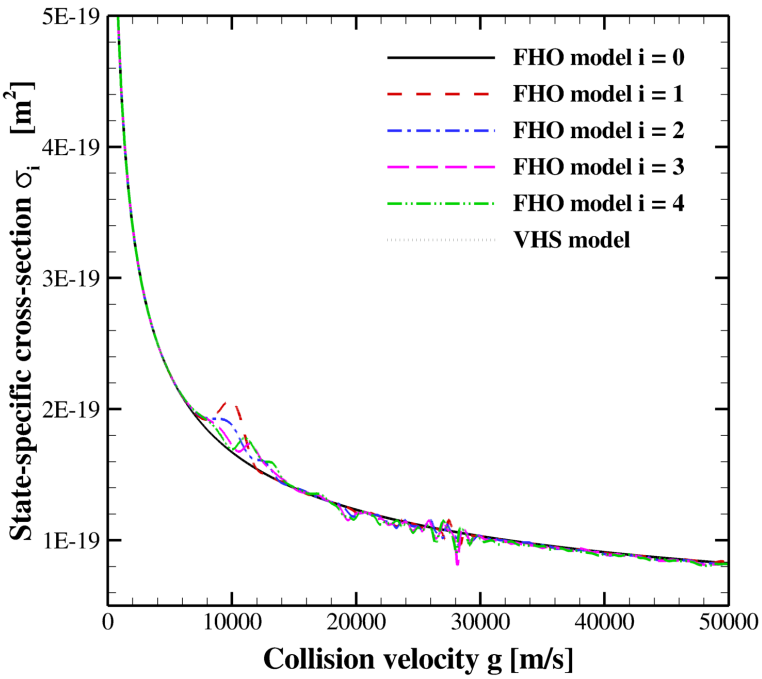
$$\begin{aligned} & P_{VVT}(i_1, i_2 \rightarrow f_1, f_2, \epsilon, \rho) \\ &= \mid \sum_{r=0}^{n} C_{r+1,i_2+1}^{i_1+i_2} C_{r+1,f_2+1}^{f_1+f_2} \exp\left[-i(f_1+f_2-r)\rho\right] P_{VT}^{1/2}(i_1+i_2-r \rightarrow f_1+f_2-r, 2\epsilon) \mid^2 \end{aligned} \tag{5.106}$$

In Eq. 5.105, i and f are the initial and final vibrational state of the molecule, and $n = min(i, f)$. i_1, i_2, f_1, f_2 in Eq. 5.106 are the initial and final vibrational states of the colliding molecules and $n = min(i_1 + i_2, f_1 + f_2)$. ϵ and ρ in the above equations are related to the two-state transition probabilities of first order perturbation theory (FOPT) [3], and are functions of the molecular properties and the symmetrized relative collision velocity (arithmetic average of the initial and final collision velocity, to enforce detailed balance and energy conversation).

Example transition probabilities according to the above expression are shown in Fig. 5.12(a) for the V-T process. The transition probability is a function of the relative velocities g of the colliding particles. To use the FHO model transition probabilities in the DSMC implementation, we determined a consistent set of state-to-state collision cross-sections for the V-T process according to section 5.3.2, with $\sigma_R(g) = \sigma_{T,VHS}(g)$, the VHS collision cross-section. Example state-specific total collision cross-sections, $\sigma_i(g)$, are plotted in Fig. 5.12(b), and compared with the VHS model total cross-section



(a) Pure V-T transition probability for a $5 \rightarrow 4$ N_2 – N_2 collision (with only one N_2 molecule participating in vibrational energy exchange).



(b) Example FHO model state-to-state collision cross-sections as a function of collision velocity g.

Figure 5.12: Example state-to-state transition probabilities and collision cross-sections in the FHO model for V-T transition process.

$\sigma_{T,VHS}(g)$. The behavior of $\sigma_i(g)$ (the oscillation behavior as a function of g) is consistent with the FHO model transition probability. Since in our implementation, the objective is to have a set of state-specific total cross-sections corresponding to a viscosity

law, the agreement with the VHS model total cross-section is expected.

By integrating the state-to-state cross-section using Eq. 5.18, the corresponding temperature dependent transition rate can be obtained, and example results are shown in Fig. 5.13. The cross-sections data could be used in the DSMC simulation as discussed above, while the transition rates data could be used in the master equation simulation.

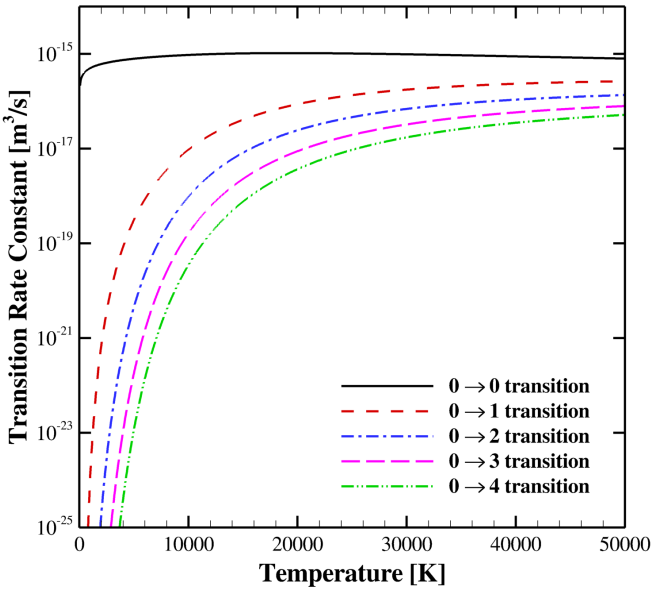


Figure 5.13: Example FHO model transition rates as a function of temperature.

To verify the FHO model implementation, we conducted a series of zero dimensional equilibrium state simulation corresponding to different temperature, using the state-to-state DSMC collision model with the FHO model VT transition probability for nitrogen gas. The AHO model is used to calculate the quantized vibrational energy. In these simulations, the translational and vibrational temperature of the system are initially set at the same value, and the molecules in the gas are then allowed to collide with each other, according to the state-to-state collision cross-sections. The gas density is set as $0.01 kg/m^3$. The DSMC simulated transition rate for each transition is then obtained from the simulation. The simulated transition rates are then compared with those calculated using state-to-state cross-sections according to Eq. 5.18. Results for representative transitions are shown in Fig. 5.14. Overall, DSMC correctly simulates the transition rate obtained from integrating collision cross-section.

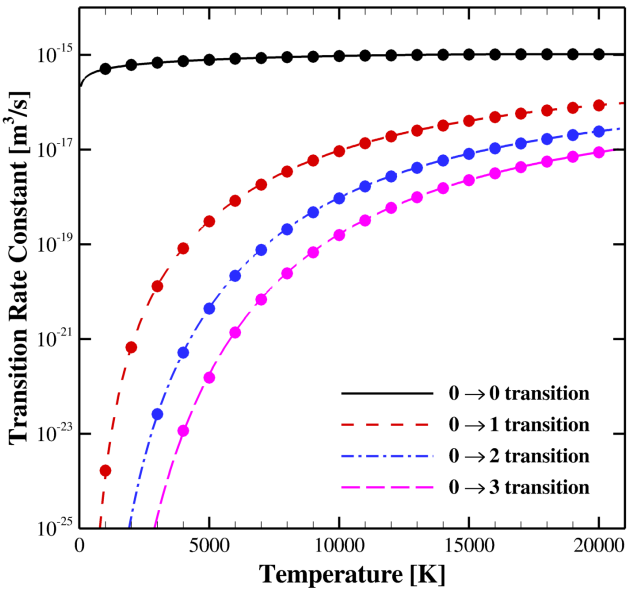


Figure 5.14: Comparison of DSMC simulated transition rate constant with that obtained from integrating collision cross-section. The lines are the transition rate constant calculated from state-to-state collision cross-section, while the symbols are the corresponding DSMC simulation results. The results are shown for representative transitions.

We then use the translational relaxation simulation [87] to establish the viscosity of the currently developed state-to-state model. In the translational relaxation simulation, the initial vibrational energy of each molecule is sampled from an equilibrium temperature T (according to the Boltzmann energy distribution). One component (x-component) of each molecule velocity is set at some specified value, while the other two components of the molecule velocity are sampled from the Maxwell-Boltzmann distribution at the same temperature T. The velocities are specified such that the average translational energy of the system corresponds to temperature T. The system are then allowed to relax toward equilibrium state. The x-component temperature T_x relaxes exponentially toward its equilibrium value $T_x(\infty) = T$. This process can be modeled by the following equation,

$$T_x(t) = T_x(\infty) + (T_x(0) - T_x(\infty))\exp\left(-\frac{t}{\tau_t}\right) \tag{5.107}$$

The characteristic translational relaxation constant, τ_t, is linked to the viscosity μ [87]. Specifically, we have,

$$\mu = p\tau_t, \tag{5.108}$$

where p is the pressure of the system corresponding to equilibrium temperature T. By fitting DSMC simulation result of translational relaxation process according to Eq. 5.107, we can obtain the translational relaxation constant τ_t at each temperature, and hence the viscosity in the DSMC state-to-state model. Example translational relaxation simulation result at $T = 5000$K is shown in Fig. 5.15, together with the corresponding fit using Eq. 5.107.

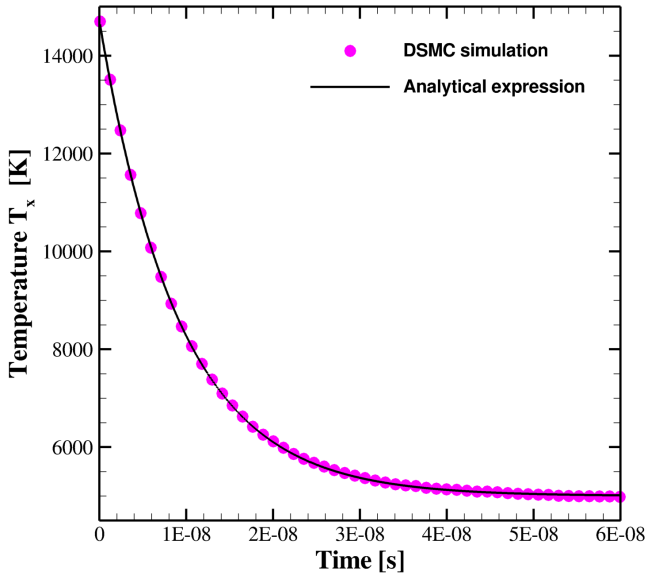


Figure 5.15: Translational relaxation simulation using the state-to-state DSMC collision model with FHO model cross-sections, at temperature $T = 5000$K. The DSMC result (symbol) is fitted exponentially (line) to obtain the translation relaxation constant, τ_t.

A series of translational relaxation simulations are conducted corresponding to different temperature T. We then obtain the viscosity of the DSMC state-to-state model using the FHO model transition probabilities. The results are shown in Fig. 5.16. Overall, the viscosity simulated by the state-to-state DSMC model is close to the VHS model viscosity (Eq. 5.38). Since the state-specific total cross-sections, $\sigma_i(g)$ of the DSMC model are linked to the VHS model total cross-sections, $\sigma_{T,VHS}(g)$ by construction, we

expect such agreement between the DSMC simulated viscosity and VHS model viscosity. The slight difference in the viscosity is due to the difference between $\sigma_{T,VHS}(g)$ and $\sigma_i(g)$.

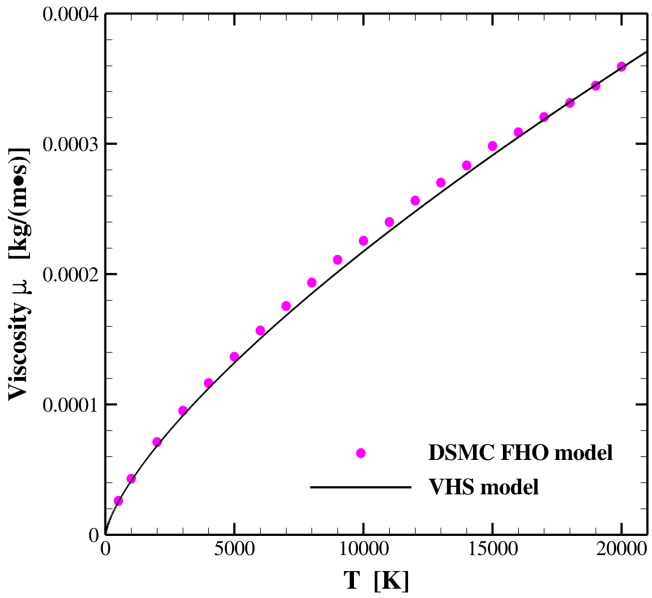


Figure 5.16: Comparison of viscosity simulated by the DSMC state-to-state model using the FHO model transition probabilities, with the VHS model viscosity.

Zero dimensional isothermal relaxation simulation are then conducted using the state-to-state DSMC collision model for nitrogen gas. The density is set at $0.1kg/m^3$. The results are shown in Fig. 5.17 for different equilibrium temperatures, with an initial vibrational temperature of $T_{vib} = 2000$K. By fitting DSMC vibration relaxation curve with the Landau-Teller equation solution (Eq. 2.1), we can then obtain the characteristic vibrational relaxation constant, τ_v, for the DSMC state-to-state model. The results are plotted in Fig. 5.18 as a function of temperature. DSMC results are also compared with the Millikan-White correlation [69] and its two variants [47, 73]. Overall, τ_v from DSMC simulation captures the general trend of the Millikan-White correlation.

Results from DSMC simulations are also compared with master equation calculation, to further validate the implementation of the state-to-state DSMC collision model. In the master equation calculation, transition rates (as a function of temperature) are directly used. The transition rates are obtained by integrating the state-to-state collision

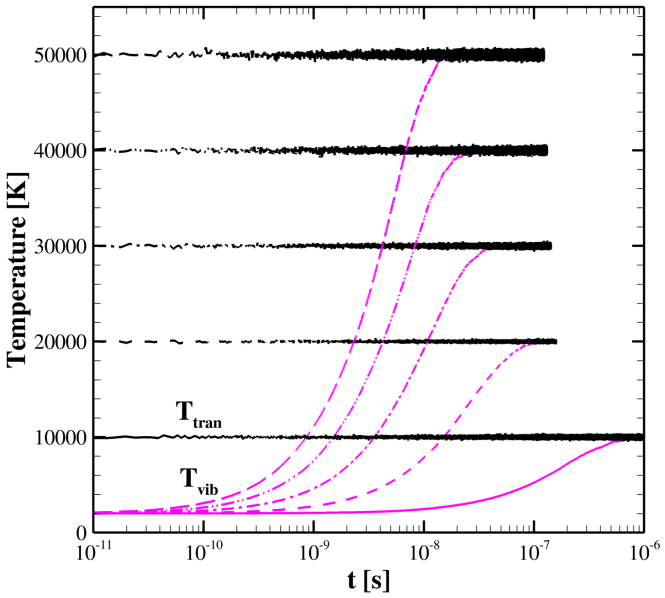


Figure 5.17: Isothermal vibrational relaxation of the state-to-state DSMC collision model with FHO model transition probability.

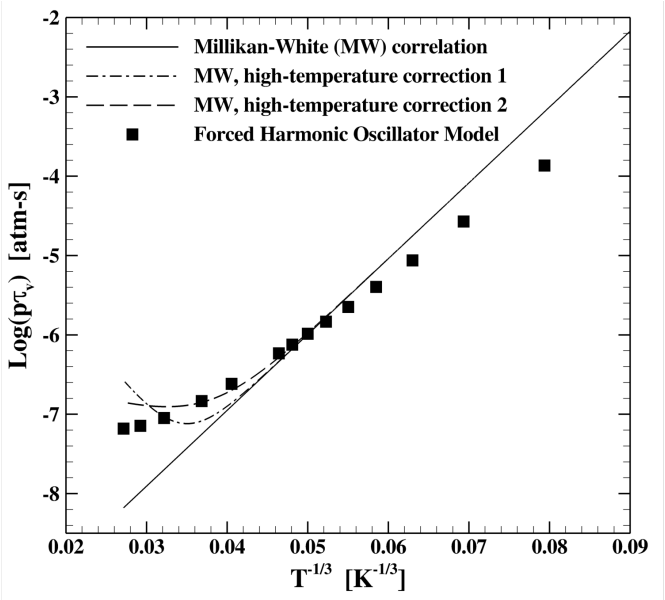
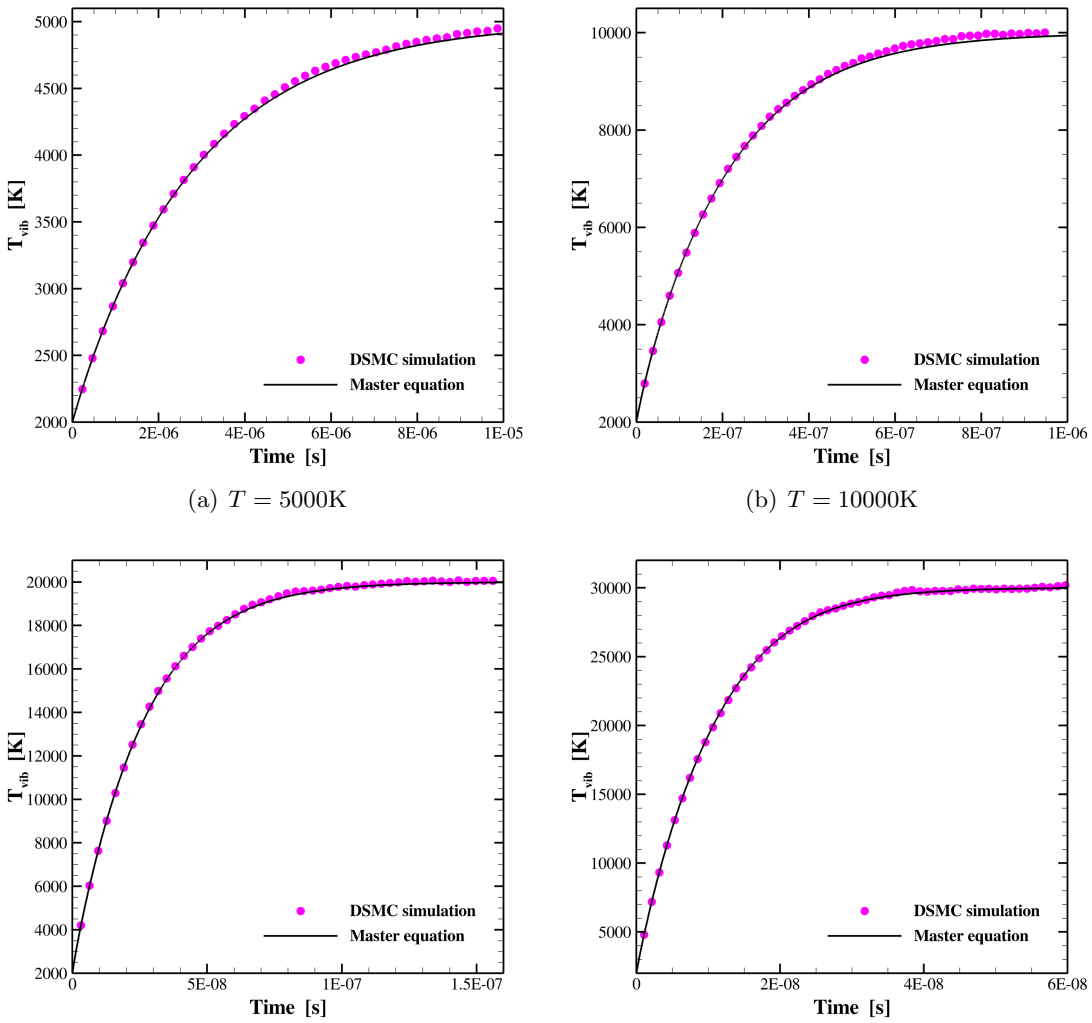


Figure 5.18: Characteristic vibrational relaxation constant obtained from state-to-state DSMC model. The DSMC results (symbol) are compared with Millikan-White correlation.

cross-sections according to Eq. 5.18. The results are shown in Fig. 5.19 for different equilibrium temperatures. Overall, the agreement is very good. This further validates the currently developed DSMC state-to-state collision model.



(a) $T = 5000$K (b) $T = 10000$K

(c) $T = 20000$K (d) $T = 30000$K

Figure 5.19: Isothermal vibrational relaxation of state-to-state DSMC collision model with the FHO model transition probability. The results are compared with the corresponding results obtained by master equation calculations.

5.5 Chapter conclusions

In this chapter, we proposed a general framework for implementing state-to-state collision models in the DSMC method. Several conclusions are drawn.

1) By examining a simple state-to-state model with analytical transition probabilities (or cross-sections), where the microscopic reversibility and detailed balance relationships are inherently satisfied [36], we identify the important factors in developing a general state-to-state collision model. Specifically, the state-to-state collision cross-section database should satisfy microscopic reversibility. When such a relationship is not satisfied, modification to the database may be required. For this situation, we propose an iterative method to arrive at a consistent set of state-to-state cross-sections that satisfy microscopic reversibility. When the microscopic reversibility is satisfied between each collision and its reverse collision process, the transition rate will also satisfy the detailed balance relation under thermal equilibrium condition macroscopically.

2) We developed a state-to-state DSMC collision model and algorithm that ensures the correct transition rate is simulated in the DSMC method. This then guarantees detailed balance is satisfied in actual DSMC simulations. Key aspects in developing a state-to-state collision model, is to propose efficient algorithms to correctly simulate the specified state-to-state collision cross-section and transition rate in the DSMC method. These includes the calculation of desired collision rate in the DSMC method, the selection of potential collision pairs, and the procedure to perform actual collision processes. To achieve these, we proposed the detailed implementation of a general rovibrational state-to-state collision model. The proposed model is demonstrated using constructed state-resolved collision probabilities.

3) With the algorithms verified, we then developed a vibrational state-to-state DSMC collision model using the transition probability data provided by the Forced Harmonic Oscillator (FHO) model [3, 68, 77, 78]. The DSMC simulation results are verified by corresponding master equation simulation results. The viscosity of the DSMC state-to-state model is also established through translational relaxation simulations.

Chapter 6

Robust Cut-cell Algorithms for DSMC Implementations Employing Multi-level Cartesian Grids

6.1 Chapter Introduction

Grid generation is requiring an ever greater percentage of total simulation turn-around time for complex three-dimensional fluid dynamic simulations. For low speed flows without strong gradients, the governing fluid equations (the Navier-Stokes equations) can be accurately discretized and solved on Cartesian-based grids, which dramatically simplify and even automate the grid generation process. However, for high-speed flows, grids typically must be aligned with strong gradients and surface geometry for accurate results. In contrast, the DSMC method [15], as a stochastic particle method that simulates the Boltzmann equation, does not require the discretization or evaluation of

This chapter is based on a paper published in Computers & Fluids: Chonglin Zhang and Thomas E. Schwartzentruber, Robust Cut-cell Algorithms for DSMC Implementations Employing Multi-level Cartesian Grids, Computers & Fluids 69, 122–135, 2012.

gradients. In DSMC, mass, momentum, and energy are transported through free molecular movement and collisions, and computational cells are used only to sort particles into near-neighbor lists. As a result, accurate DSMC simulations can be obtained on Cartesian-based grids for both low and high-speed flows. Furthermore, compared to DSMC implementations employing a body-fitted tetrahedral grid, the use of a Cartesian flow field grid greatly simplifies the grid generation process and adaptive mesh refinement, especially for simulations involving complex surface geometries [54, 64, 99].

Such a Cartesian-grid approach can, however, pose difficulties for the determination of the intersections between the background Cartesian flow field grid and the complex surface geometry where boundary conditions must be specified. Typically, complex surface mesh geometries are triangulated and are simply represented in the code by a large number of triangles. Any flow field grid cell that intersects with the surface geometry is called a "cut-cell". The key information required for each cut-cell includes a list of the triangulated surface elements that intersect with the cell and the volume of the cut-cell that remains in the flow field. The list of triangles is required for particle movement and detecting surface collisions and the volume is required to accurately set the local collision rate in the DSMC simulation.

The DSMC method typically uses a "ray-trace" method to move particles through the grid one cell at a time, by determining particle intersections with cell-faces. Thus, if a surface mesh "cuts" through a flow field cell, only the geometry of the surface mesh contained within that flow field cell is required by the ray-trace algorithm to detect a collision with the surface. As long as the surface mesh can be sorted within the flow field Cartesian cells, standard DSMC algorithms for particle movement and collisions require no modification for complex geometry. Additionally, in DSMC simulations, the probability with which two particles in a computational cell should be selected for collision, depends on the volume of the computational cell. However, if a Cartesian grid is used, the computational cells near the surface region are mostly cut-cells with complex shapes due to the intersection of regular shaped Cartesian cells with the surface geometry. The volume of these complex shaped cut-cells must be calculated accurately, in order to achieve the correct collision rates for simulated particles, the correct macroscopic properties, and hence accurate simulation results.

Furthermore, for an accurate and efficient DSMC simulation, the flow field grid must

be locally refined to the mean free path of the gas. As a result, it is desirable to decouple the flow field grid resolution from the surface geometry resolution. If DSMC grid cells are too large, non-local particles are selected for collisions resulting in numerical diffusion of gradients. If DSMC cells are too small, since a statistically relevant number of particles are required per cell, the number of simulated particles becomes far more than necessary, resulting in an accurate but highly inefficient simulation. For low density flows, such as rarefied flow around a low-orbiting satellite, the mean free path of the flow may be of the same order of magnitude as the satellite itself. Thus the optimal size of the DSMC flow field cells may be much larger than fine geometry details of the satellite surface. For such low density flows, it is possible that many surface triangles intersect and cut a highly complex volume out of a single Cartesian flow field cell. In some cases, one large Cartesian cell may be divided into several flow volumes which don't connect with each other (termed "split-cells"). Such situations should be recognized by the geometry algorithms, and particles within each of the split cells should be treated separately as they may lie on the opposite side of a solid wall.

For steady-state DSMC simulations, cut-cell information must be determined at the beginning of the simulation and, if adaptive mesh refinement (AMR) is utilized, each time either the flow field grid or triangulated surface mesh is modified (perhaps 5-10 times during a typical simulation). For unsteady DSMC simulations, where a changing density field requires continual adaptive mesh refinement, the cut-cell information must be computed as often as every few DSMC time steps. Further details of the DSMC Cartesian grid structure, surface representation, adaptive mesh refinement, and particle ray-tracing techniques will be discussed later in this chapter.

The purpose of this chapter is to provide a comprehensive discussion of various cut-cell algorithms applicable for use in a Cartesian grid based DSMC implementation. This chapter completely details all aspects of the algorithms and tests each on a variety of complex geometries. The geometrical algorithms are general to any Cartesian grid structure and discussion of their implementation in DSMC codes employing single or multi-level Cartesian grids is also included. In general, the cut-cell algorithms can be divided into the following components:

- Sorting all surface triangles into the appropriate flow field Cartesian cells.

- Calculating the volume of the cut-cells accurately and efficiently.
- Identifying all split-cells resulting from single Cartesian cells that are divided into separate volumes by the surface geometry.

Specifically, in section 6.2, three existing algorithms for the sorting of surface triangles are described and their relative efficiency is compared for various complex geometries. Then, in section 6.3, newly developed algorithms for computing cut-cell volumes and split-cells are presented and tested for accuracy and efficiency. Integration of these algorithms into DSMC simulations is described in section 6.4, and chapter conclusions are presented in section 6.5.

6.2 Surface Mesh Sorting

Sorting triangulated surface elements into the Cartesian flow field cells that they intersect with, ultimately produces a list of cell faces and triangulated surface faces that particles in a given cell may cross, or reflect-from, during the ray-trace particle movement procedure. This list of cell faces and triangulated surface elements also facilitates the calculation of the cut-cell volume (discussed later in section 6.3). It is important to note that for the ray-trace algorithm, the triangulated surface elements sorted within a Cartesian cell do not need to be clipped. The triangles can extend well beyond the faces of the Cartesian cell. If a particle crosses a Cartesian cell face before hitting the triangle element, the ray-trace algorithm will naturally move the particle to the neighboring Cartesian cell and continue using the new cell's list of surface triangles. In this manner, surface triangles only need to be sorted into all Cartesian cells that they intersect with, and therefore the same triangle may be sorted into multiple Cartesian cells.

A surface triangle should be sorted into a Cartesian cell, as long as any part of it lies within the volume region formed by the Cartesian cell. Thus, the core algorithm involves determining whether a specific Cartesian cell intersects with a specific surface triangle. The intersection of a triangle and a Cartesian cell (referred to as an "axis aligned cube" or "axis aligned bounding box") fits into the general category of an Object/Object Intersection test. A variety of methods exist for handling this task. In this chapter, three different methods are investigated. The first method is the most intuitive and straightforward to implement. It is based on determining the intersection of a line segment with

a square or triangle by using the signed tetrahedral volume [5]. The second method, called the separating axis theorem (SAT) [6], takes a significantly different and somewhat non-intuitive approach. The third method, the Voorhies algorithm [90], expands on the signed tetrahedral volume method to gain efficiency. Both SAT and Voorhies methods come from the computer graphics research community. For completeness, we give a detailed description of all three algorithms and later ensure consistency of the results and compare efficiencies.

6.2.1 Signed Tetrahedral Volume Approach

A triangle represented by T, should be sorted into the corresponding Cartesian cell C, as long as they intersect in the region formed by C,

$$T \cap C \neq \emptyset \tag{6.1}$$

otherwise T should not be sorted into C. Three situations exist for the relative position between a Cartesian cell C and a triangle T:

(1) $T \cap C = \emptyset$. T and C don't intersect with each other.

(2) $T \subset C$. T is completely contained in C.

(3) $T \cap C \neq \emptyset$, and $T \not\subseteq C$. T intersects with C, but T is not completely contained in C.

The first step is to quickly accept or reject trivial intersection situations. Specifically, an intersection is immediately confirmed if any of the 3 triangle vertices lie inside the cubic region. In contrast, if all 3 triangle vertices lie on the same side of a Cartesian cell face (any of the 6 faces) *and* they lie on the side opposite to the cube itself, then an intersection is trivially rejected (the triangle does not intersect the Cartesian cell). Each algorithm described in this section first applies these trivial acceptance/rejection tests to a given Cartesian cell and surface triangle pair. If still unresolved, each method proceeds to determine intersection in different manners.

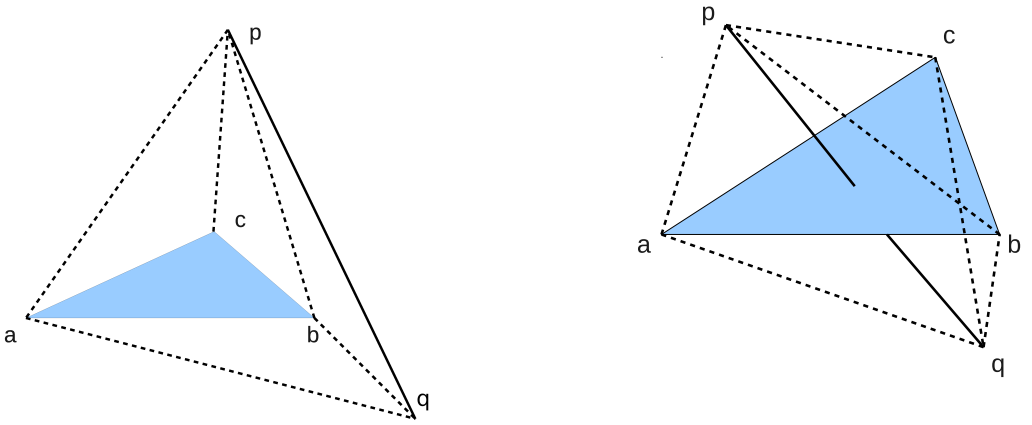
For the signed tetrahedral volume method, the algorithm proceeds to check whether the triangle intersects with any of the 6 individual faces of the Cartesian cell or not. If no

intersection is found, then this corresponds to situation (1), otherwise this corresponds to situation (3). To check whether a triangle intersects with a face of the Cartesian cell, this is equivalent to check whether one edge of the triangle intersects with the face, or one edge of the face intersects with the triangle. As a face of the Cartesian cell can be divided into two triangles, the problem can then be transformed to determine whether a general line segment intersects with a triangle in three-dimensional space. This is done by the signed tetrahedral volume [5] approach as discussed below.

For a general tetrahedron formed by four vertices m, n, r, s, denote their coordinates by (m_x, m_y, m_z), (n_x, n_y, n_z), (r_x, r_y, r_z) and (s_x, s_y, s_z). The signed volume of the tetrahedron $V(m, n, r, s)$ [5] is given by:

$$V(m,n,r,s) = \frac{1}{6}\begin{vmatrix} m_x & m_y & m_z & 1 \\ n_x & n_y & n_z & 1 \\ r_x & r_y & r_z & 1 \\ s_x & s_y & s_z & 1 \end{vmatrix}, \tag{6.2}$$

where it is noted that the order of the four vertices affect the value of $V(m, n, r, s)$.



(a) l_{pq} does not intersect with T_{abc}

(b) l_{pq} intersects with T_{abc}

Figure 6.1: Signed tetrahedral volume method to detect an intersection between l_{pq} and T_{abc}

For a line segment l_{pq} and a triangle T_{abc} (refer to Fig. 6.1), let the coordinates of the end points of l_{pq} be (p_x, p_y, p_z) and (q_x, q_y, q_z), and the coordinates of the three vertices of T_{abc} be (a_x, a_y, a_z), (b_x, b_y, b_z) and (c_x, c_y, c_z). Whether l_{pq} intersects with T_{abc} can be achieved via the following procedures:

(1) If $V(p, a, b, c)$ and $V(q, a, b, c)$ have different signs, this means that l_{pq} intersects with the plane formed by T_{abc}, then continue to step (2); otherwise, l_{pq} doesn't intersect with T_{abc}, stop.

(2) If $V(p, a, b, q)$, $V(p, b, c, q)$ and $V(p, c, a, q)$ have same sign, then l_{pq} intersects with T_{abc}; otherwise, l_{pq} does not intersect with T_{abc}.

As shown in Fig. 6.1(b), both conditions above are satisfied, and l_{pq} intersects with T_{abc}. While in Fig. 6.1(a), although l_{pq} intersects with the plane formed by T_{abc}, the second condition is not satisfied and l_{pq} does not intersect with T_{abc}.

6.2.2 Separating Axis Theorem

The separating axis theorem (SAT) [6] states that: two convex polyhedrons, A and B, are disjoint (do not intersect) if they can be separated along either an axis parallel to the normal of a face of either A or B, or along an axis formed by the cross product of an edge from A and an edge from B. A separating axis separating A and B means that the projections of A and B onto the axis do not overlap with each other.

To determine the intersection of a triangle and a cube, we first apply the trivial acceptance/rejection tests described in section 6.2.1. For unresolved cases, based on SAT, we test for each of the 13 possible axes to determine if they are separating axes or not. The axes tested include the 3 normals of the cubic faces, the normal of the triangle, and the 9 axes resulting from cross products between the cubic edges and triangle edges.

For a specific triangle and a specific cube, if all 13 tests are passed, then there is no separating axis and the triangle intersects with the cube. On the other hand, if at any point a separating axis is found, then the algorithm terminates and the triangle does not intersect with the cube.

6.2.3 Voorhies Algorithm

The Voorhies algorithm [90] is divided into three main tasks, described below. Similar to the previous two methods described above, the trivial acceptance/rejection tests discussed in section 6.2.1 are applied first. For unresolved intersection cases, the following procedure is then followed:

(1) Test the triangle vertices against additional planes formed by the Cartesian cell. Specifically, compare the 3 triangle vertices against 12 planes, each of which pass through one of the 12 cubic edges and form a 45° angle with the edge's adjacent face. These planes are called edge-planes. Next, compare the 3 triangle vertices against 8 additional planes, each of which pass through one of the 8 cubic corners (cubic vertices) and are perpendicular to the corresponding cubic diagonal. These planes are called corner-planes. If all 3 triangle vertices lie outside the same edge-plane or corner-plane, the triangle does not intersect with the Cartesian cell. Otherwise, the algorithm proceeds with additional tests.

(2) Check whether any of the 3 triangle edges intersect with any of the 6 faces of the cube [90]. If any intersection is found then the triangle intersects with the Cartesian cell. Otherwise the algorithm proceeds with additional tests.

(3) Check whether the 4 diagonals of the cube intersect the triangle [90]. If any intersection is found then the triangle intersects with the Cartesian cell. Otherwise no more additional tests are required and the triangle does not intersect with the Cartesian cell.

It should be mentioned that, in the three steps described above, bit-wise operations [4, 5, 90] are used when possible, to reduce the time cost. The bit-wise operation assigns a bit code to each point in space based on its location, and use logical "AND" and "OR" operations to determine the position of these points relative to the other planes of interest.

6.2.4 Time costs for an idealized sorting problem

As the three algorithms use different strategies and operations to determine intersection, and may exit at different parts of the algorithm, their time cost will be different for the

same Cartesian cell and triangle pair. However, it is possible that the relative costs may also be case dependent where one algorithm may perform better than another depending on the exact configuration of the cell and the triangle.

To get an estimation of the relative time costs, we perform intersection tests of a unit cube with arbitrarily generated triangles [6]. The unit cube is centered at the origin $(0, 0, 0)$, and has a length of 1. The vertices of each triangle are randomly generated within a cube of length 4 centered at the origin $(0, 0, 0)$. We sequentially perform 100 tests for each algorithm. For each test, we evaluate 1 million arbitrarily generated triangles against the unit cube. First, we compare the intersection results of the three algorithms and it is verified that the results precisely agree with each other. By averaging the 100 tests, we get the averaged time cost of performing 1 million triangle/cube intersection tests for each algorithm. The results are displayed in Table 6.1. The time costs of the three algorithms are of the same order for the current test. In this test, approximately half of the triangle and Cartesian cell pairs are determined to intersect with each other.

Table 6.1: The time costs of performing an intersection test of 1 million arbitrarily generated triangles against a unit Cartesian cell.

Algorithm number*	Average time (s) †	Average time per pair of triangle and Cartesian cell (s)
1	1.02	1.02×10^{-6}
2	0.21	2.1×10^{-7}
3	0.42	4.2×10^{-7}

* Algorithm 1: Signed Tetrahedral Volume Algorithm; 2: Separating Axis Theorem; 3: Voorhies Algorithm.

† The tests are conducted on a quad-core Intel Xeon W3520 CPU at 2.67GHz. Other time costs studies in this paper are conducted on the same platform.

6.2.5 Sorting on multi-level Cartesian grids

For DSMC simulations involving large numbers of Cartesian flow field cells, a good data structure is beneficial for time efficiency when sorting surface elements. In the worst situation, the time cost scales with the product of the number of the Cartesian cells and the number of surface triangles.

In this chapter, we take advantage of the three-level Cartesian grid data structure used in the MGDS DSMC code [43], which is shown in Fig. 6.2. In this data structure, the size of the L1 cells are specified according to a user-defined fraction of the free-stream mean free path, and the L1 cell sizes do not change during the simulation. Each L1 cell can be refined into any number of L2 cells, each of which can each be refined into any number of L3 cells. Specifically, $dx_1 = r_2 \times r_3 \times dx_3$. Here r_2 and r_3 are the refinement ratios of L2 and L3 cells,

$$r_2 = \frac{dx_1}{dx_2},\ r_3 = \frac{dx_2}{dx_3}\ , \tag{6.3}$$

and dx_1, dx_2 and dx_3 are the size of the L1, L2, and L3 cells.

Sorting on such multi-level Cartesian grids has similar benefits to using an alternating digital tree (ADT) data structure technique [5, 18]. The algorithm we use to sort the triangles is listed below:

- First loop over all L1 cells. For a specific L1 cell C_i, find all the triangles that intersect with it using one of the three algorithms described above. These triangles are expressed as $T_i = \{T_i^1, T_i^2, ..., T_i^{M_i}\}$.

- Second, for each C_i, loop over all of its L2 cells. For a specific L2 cell $C_{i,j}$ contained in C_i, find all the triangles intersecting with it from $T_i = \{T_i^1, T_i^2, ..., T_i^{M_i}\}$. The resulting triangles intersecting with $C_{i,j}$ are a subset of T_i and are expressed as $T_{i,j} = \{T_{i,j}^1, T_{i,j}^2, ..., T_{i,j}^{M_{i,j}}\}$.

- Third, for each $C_{i,j}$, loop over every L3 cell contained within it. For a specific L3 cell $C_{i,j,k}$, find all the triangles intersecting with it from $T_{i,j}$. The resulting triangles intersecting with $C_{i,j}$ are a subset of $T_{i,j}$, and are expressed as $T_{i,j,k} = \{T_{i,j,k}^1, T_{i,j,k}^2, ..., T_{i,j,k}^{M_{i,j,k}}\}$.

- After looping over all L3 cells $C_{i,j,k}$ contained in the L1 cell C_i, and finding the triangles intersecting with them, continue to the next L1 cell, and repeat the above three procedures.

Since the L1 cell size remains fixed during a MGDS DSMC simulation, each L1 cell can maintain a list of the triangles contained within it. As a result, subsequent calls to the

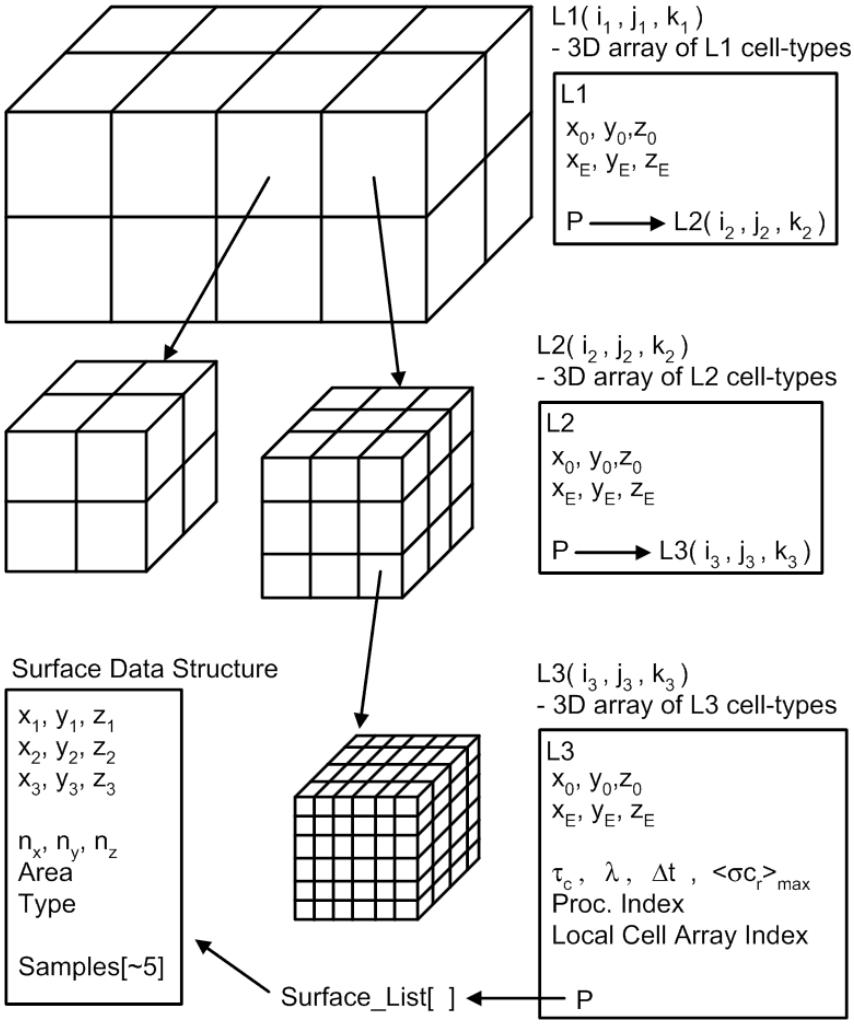


Figure 6.2: The data structure of the three-level Cartesian grid MGDS DSMC code.

triangle sorting algorithms (for example during adaptive mesh refinement) can proceed directly to process the L2 cells, potentially reducing the time cost significantly.

6.2.6 Sorting time costs for realistic geometries

To estimate the time cost of the sorting algorithm, we make the following assumptions:

(i) The average time cost to test the intersection of one pair of Cartesian cell and triangle is C.

(ii) Each L1 cell of the three-level Cartesian grid has the same refinement ratios: r_2 and r_3. The total number of L1, L2 and L3 cells are N_1, N_2 and N.

(ii) The total number of surface triangles is M. They are uniformly distributed over the entire Cartesian cell region. The L1, and L2 cells contain M_1 and M_2 triangles, respectively.

Based on these assumptions, we have,

$$N_1 = \frac{N}{r_2^3 r_3^3},\ N_2 = \frac{N}{r_3^3} \tag{6.4}$$

and approximately,

$$M_1 = \frac{M}{N_1},\ M_2 = \frac{M}{N_1 r_2^3} \tag{6.5}$$

The total time cost of the initial sorting call is thus,

$$t_1 = CN_1(M + r_2^3(M_1 + r_3^3 M_2)) = C(N\frac{M}{r_2^3 r_3^3} + M(r_2^3 + r_3^3)). \tag{6.6}$$

Since, usually we have $N > M$, then for large N, t_1 can be approximated as $t_1^{(N>M)} = CN\frac{M}{r_2^3 r_3^3}$, which scales linearly with N and M. Similarly, the time cost of the subsequent call to the sorting algorithm is,

$$t_2 = CM(r_2^3 + r_3^3). \tag{6.7}$$

As can be seen from Eq. 6.7, t_2 doesn't depend on N theoretically. The time ratio or speed-up for the subsequent call compared to the first call is

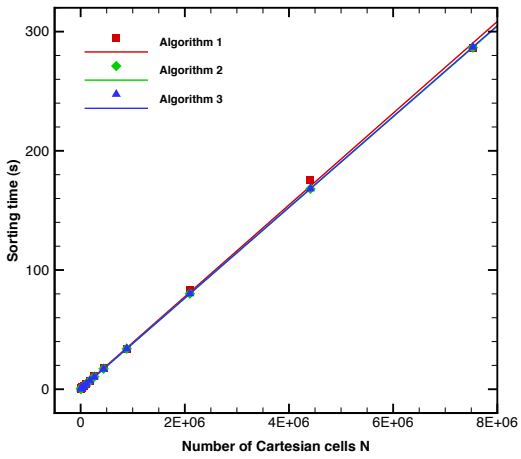
$$t_r = \frac{t_1}{t_2} = 1 + \frac{N}{r_2^3 r_3^3 (r_2^3 + r_3^3)}. \tag{6.8}$$

We then run a series of triangle sorting tests where N (the total number of L3 cells) is varied and M (the total number of surface geometry triangles) is fixed for both two-level and three-level Cartesian grids. The first geometry chosen for the surface

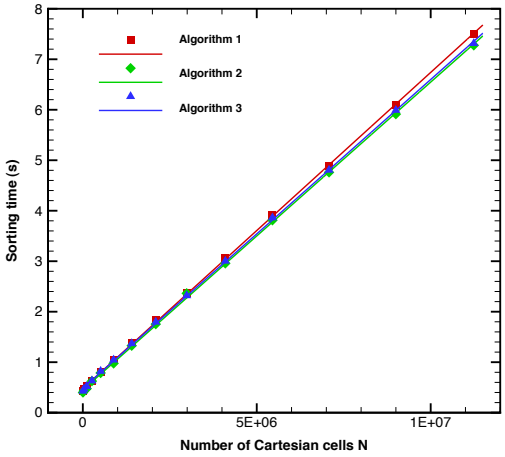
mesh is a geometry resembling the MIR Space Station [65] (shown later in Figs. 6.6 and 6.16) which contains 53,290 triangles. The second geometry chosen is a planetary probe geometry [70] (shown later in Fig. 6.11), which contains 9,568 triangles. The two refinement ratios considered are $r_2 = 1, r_3 = 4$ and $r_2 = 4, r_3 = 4$, respectively. The former corresponds to a two level Cartesian grid, while the latter corresponds to a three level Cartesian grid. It is important to note that the flow field Cartesian grids are uniform and that for a specific value of N, the finest level cells in both two-level and three-level Cartesian grids are identical, only their storage and organization in the geometry data structure are different.

We plot the total time cost for the initial sorting task on both two and three level Cartesian grids in Fig. 6.3, for both MIR and planetary probe surface meshes, in function of N for each of the three sorting methods. As can be seen from Figs. 6.3(a) and 6.3(b), the sorting time for the MIR Space Station grid scales linearly with N. Also, the sorting time required for the three level grid (Fig. 6.3(b)) has a much lower time cost than required by the two level grid (Fig. 6.3(a)). The same trends are seen in Figs. 6.3(c) and 6.3(d) for the planetary probe surface mesh, where it is also apparent that the time cost required for the Probe geometry is much less than that for the MIR geometry, as the time cost scales roughly linearly with the number of triangles M for fixed N. In contrast to the time costs for the idealized sorting problem (Table 6.1), Figs. 6.3(a)-6.3(d) reveal that all three sorting methods require virtually the same computational time for these realistic geometries. The reason is that each method is able to quickly accept/reject the trivial cases discussed in section 6.2.1. Unlike the idealized sorting problem, realistic geometries have inherent order and locality of triangles and thus there are a large number of trivial rejections. There are however situations where a difference in sorting time between methods is noticeable as seen in Fig. 6.3(d).

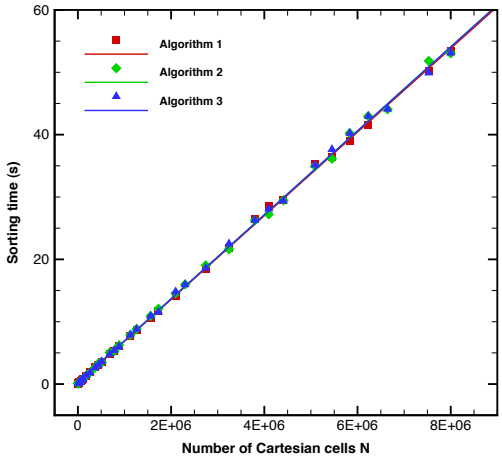
We expect that subsequent calls to the sorting algorithm will have a lower time cost since triangles have already been sorted into the L1 cells. This is confirmed in Figs. 6.4(a) and 6.4(c) where the new time costs are plotted for each algorithm versus N for the two level Cartesian grid with mesh refinement ratios $r_2 = 1, r_3 = 4$. The first observation is that by comparing Figs. 6.4(a) and 6.4(c) with Figs. 6.3(a) and 6.3(c), the time costs required for the second sorting call is reduced by roughly two orders of magnitude when the number of flow field cells surpasses a few million. In



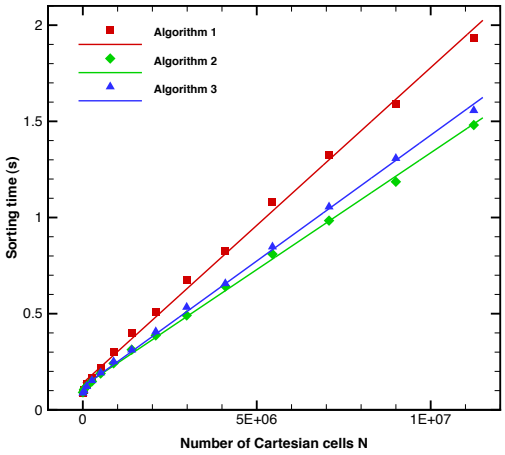
(a) All algorithms with mesh refinement ratio $r_2 = 1, r_3 = 4$ for MIR geometry.



(b) All algorithms with mesh refinement ratio $r_2 = 4, r_3 = 4$ for MIR geometry.



(c) All algorithms with mesh refinement ratio $r_2 = 1, r_3 = 4$ for Probe geometry.



(d) All algorithms with mesh refinement ratio $r_2 = 4, r_3 = 4$ for Probe geometry.

Figure 6.3: The time costs of the triangle sorting routine as a function of the number of Cartesian cells N. The results correspond to the initial call to this routine using two different mesh refinement ratios. Results are shown for each of the three algorithms for both MIR and Probe geometries. The lines in these figures are linear fits to the data points.

Figs. 6.4(a) and 6.4(c), for sorting algorithms 2 and 3, the time cost scales only weakly with N, for large N. This is the beneficial result of subsequent sorting calls when using a multi-level Cartesian grid. Since subsequent calls test each Cartesian cell with far

fewer triangles, there are far fewer trivial acceptances and rejections and the various methods now require different time costs. For subsequent sorting calls, methods 2 and 3 are seen to be significantly faster than method 1. Finally, although not shown, the same trend is found when a three level Cartesian grid is used. In this case, the first call to the sorting routine is already very fast (as seen in Figs. 6.3(b) and 6.3(d)) with a significant speedup for subsequent calls.

An alternate technique to increase the efficiency of the sorting algorithms is to approximately pre-sort triangles into Cartesian cells. Here, the pre-sorting algorithm loops over all surface triangles. For each triangle, its axis aligned bounding box is first determined using the lower and upper coordinates of its three vertices. This bounding box is then tested with every L1 Cartesian cell for intersection. If the bounding box intersects with one L1 Cartesian cell, then the triangle is added to that Cartesian cell's list of triangles. In this manner, all triangles are approximately pre-sorted into L1 Cartesian cells with the number of operations scaling with M. The sorting efficiency obtained when such pre-sorting is used, is very similar to the sorting efficiency for a second sorting call. Time costs for the sorting algorithms including both pre-sorting and the first call to the sorting routine are shown in Figs. 6.4(b) and 6.4(d) and are indeed seen to be similar to the results shown in Figs. 6.4(a) and 6.4(c).

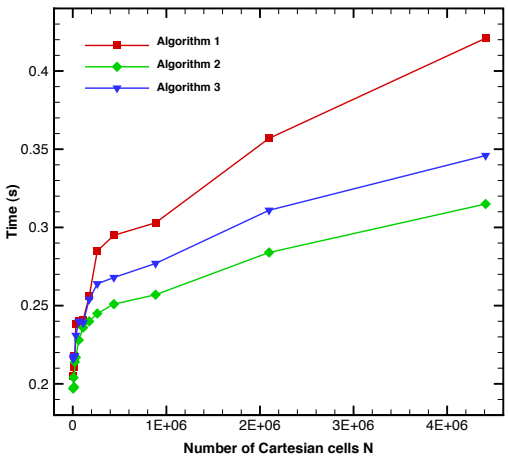
6.3 Cut-cell Volume Calculation

When surface triangles intersect with a Cartesian cell and result in cut-cell, the portions of surface triangles inside the Cartesian cell combined with the Cartesian cell faces form one or multiple polyhedrons located in the flow field region, as shown in Fig. 6.6. Depending on the number of isolated polyhedrons formed, we define two types of cut-cells:

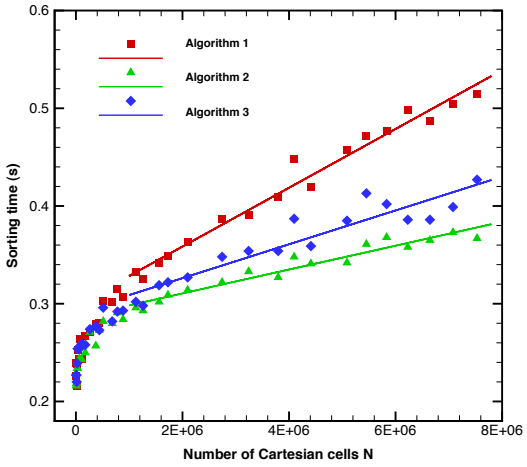
(1) Regular cut-cell, a cut-cell containing only one polyhedron in the flow field region.

(2) Split cut-cell (or simply called split-cell), a cut-cell containing multiple polyhedrons in the flow field region.

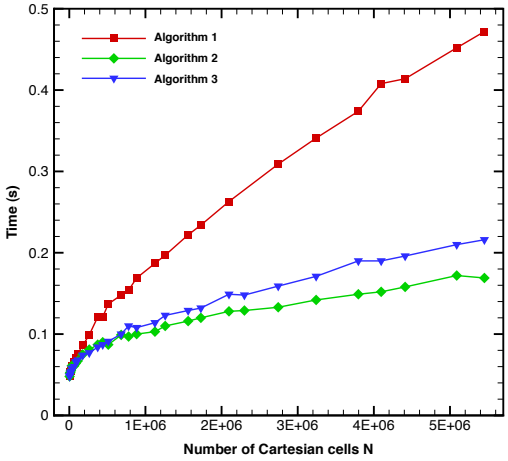
In this section we present two methods, including all algorithmic steps, that can be used to calculate the cut-cell volumes. Here, by cut-cell volume, we mean the part of



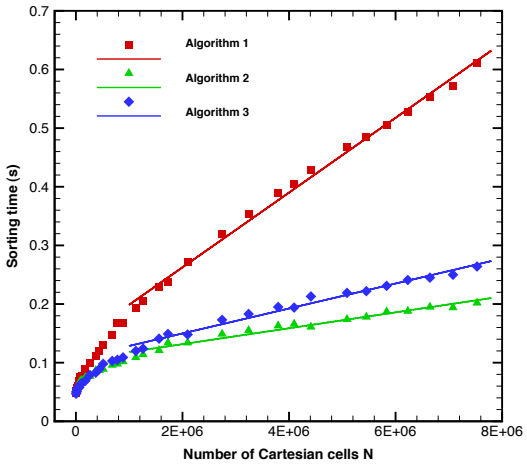
(a) Time costs for the MIR geometry during a second call to the sorting routine.



(b) Time costs for the MIR geometry including both the pre-sort routine followed by the first call to the sorting routine.



(c) Time costs for the Probe geometry during a second call to the sorting routine.



(d) Time costs for the Probe geometry including both the pre-sort routine followed by the first call to the sorting routine.

Figure 6.4: The time costs of the triangle sorting routine as a function of the number of Cartesian cells N. All results correspond to a two level Cartesian grid. Results are shown for each of the three methods for both MIR and Probe geometries.

the volume inside the flow field region. The first method is indirect, and uses a Monte Carlo sampling technique, which is quite simple to implement and robust, however is

not capable of identifying split-cells. The second method directly computes the cut-cell volumes by forming corresponding polyhedron/polyhedrons and is naturally able to detect and compute split-cell volumes.

6.3.1 Indirect method – Monte Carlo sampling

Suppose a Cartesian cell $C_{i,j,k}$ with volume V, is cut through by surface triangles $T_{i,j,k} = \{T^1_{i,j,k}, T^2_{i,j,k}, ..., T^{M_{i,j,k}}_{i,j,k}\}$, and the volume of the resulting cut-cell is V_c. By randomly generating N_P number of points inside the Cartesian cell $C_{i,j,k}$, and counting the total number of points inside the flow field region, the volume of the cut-cell can be estimated as:

$$V_c = V\frac{N_c}{N_P} \tag{6.9}$$

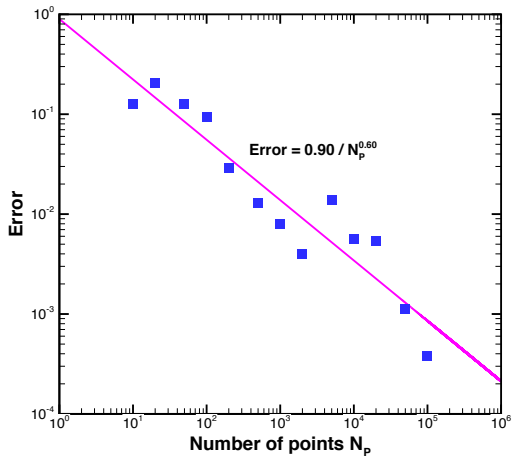
where, N_c is the number of points inside the flow field. The sampled volume, V_c, converges to its exact value as the number of Monte Carlo points used increases. When N_P number of points are used, the difference between the exact value and calculated value of volume is inversely proportional to the square root of N_P, i.e, error $\propto \frac{1}{\sqrt{N_P}}$. A test calculation was performed for a Cartesian cell that is cut by a sphere of radius $0.9dx$, centered at one vertex of the Cartesian cell. The portion of the sphere inside the Cartesian cell contains roughly 200 triangulated surface elements. The expected error convergence of the Monte Carlo method is verified in Fig. 6.5(a), which compares the error of calculated volume using this method (V_{cal}) with that of the second, direct method (V_{true}) described in the next subsection. The volume from second method is exact to within machine precision.

Determining whether a randomly generated point P is inside the flow field part of a cut-cell, can be achieved via the following procedures [4, 5]:

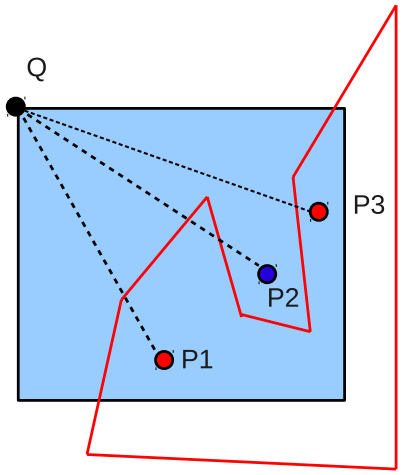
(1) Choosing a known point Q, which is inside the flow field.

(2) Counting the number of intersections between the line segment l_{PQ} and the triangles representing the surface that have already been sorted into the Cartesian cell.

(3) If the number of intersection points is even, then P is inside the flow field; otherwise P is outside the flow field, and is inside the region formed by the surface.



(a) The volume calculation error of the Monte Carlo method. Error = $\frac{|V_{cal}-V_{true}|}{V_{true}}$



(b) Determining whether a point is inside the flow field region.

Figure 6.5: Monte Carlo sampling technique to determine cut-volumes.

The above procedure can be seen from a two dimensional analogy in Fig. 6.5(b), where Q is the known point that is inside the flow field, and P_1, P_2 and P_3 are three test points. As P_1Q, P_2Q and P_3Q intersect with the triangle surface by 1 time, 2 times, and 1 time, respectively, only P_2 is inside the flow field, while P_1 and P_3 are not. When a point Q inside the flow field doesn't exist, we choose a point that is known to lie inside the surface region, and the logic is similar to that discussed above.

6.3.2 Direct method – forming Polyhedron/Polyhedrons

To determine all the polyhedrons contained in one cut-cell, thus identifying potential split-cells, and to accurately calculate the volume of each polyhedron, we propose a direct method in this section. The algorithms introduced here operate on individual Cartesian cells, one at a time, and use a bottom-up approach to form polyhedron. Specifically, the edges of a polyhedron are found first; then the polyhedron faces are formed using these edges; and finally the polyhedron is determined by closing all of the polyhedron faces. By counting the number of polyhedrons, we naturally determine whether a cut-cell is a regular cut-cell or a split-cell. If all of the faces of each polyhedron

are known, its volume can be calculated directly. The input of the algorithms are the Cartesian cell and the list of triangles which intersect with the Cartesian cell (determined by the algorithms of section 6.2). The output of the algorithms are the number of polyhedrons contained in the cut-cell, the faces of each polyhedron, and the volume of each polyhedron.

The algorithms are composed of 4 major tasks, and are completely detailed below. The first task of the algorithms, i.e. the surface triangle clipping, is achieved using the Sutherland-Hodgmann (S-H) mesh clipping algorithm [5] from the computer graphics field. The remaining three tasks are accomplished using new algorithms proposed in this paper. An illustrative flowchart portraying each task for an example split-cell from a simulation involving the MIR Space Station geometry is displayed in Fig. 6.6.

Before describing the detailed algorithms, we first define some terminologies that will be used later.

(1) Triangle-polygon (TP). Following the definition used by Aftosmis *et al.* [5], we call the polygon resulting from clipping a triangle against a Cartesian cell triangle-polygon. The triangle-polygon is the portion of the triangle inside the Cartesian cell.

(2) Face-polygon (FP). A face-polygon is either the entire Cartesian cell face (CF) if no triangle-polygon's boundary edge is contained in the face (type 1), or it is part of the Cartesian cell face and is composed of triangle-polygons' boundary edges and the Cartesian cell edges (type 2). Examples of these two types of face-polygons are shown in Fig. 6.6. The type 2 face-polygon is not convex in general, due to the inclusion of the triangle-polygon's boundary edge.

6.3.2.1 Surface Triangle Clipping

The starting point is a Cartesian cell $C_{i,j,k}$ and a list of triangles intersecting with it (denoted as $T_{i,j,k} = \{T^1_{i,j,k}, T^2_{i,j,k}, ..., T^{M_{i,j,k}}_{i,j,k}\}$, where $M_{i,j,k}$ is the number of triangles intersecting with $C_{i,j,k}$). When a triangle $T^l_{i,j,k}$ intersects with $C_{i,j,k}$, only the part of $T^l_{i,j,k}$ inside $C_{i,j,k}$ (known as a triangle-polygon defined above) is necessary to form the polyhedron. Due to its efficiency and easy implementation, we use the S-H mesh clipping algorithm [5] to clip every triangle intersecting with $C_{i,j,k}$, and only leave the

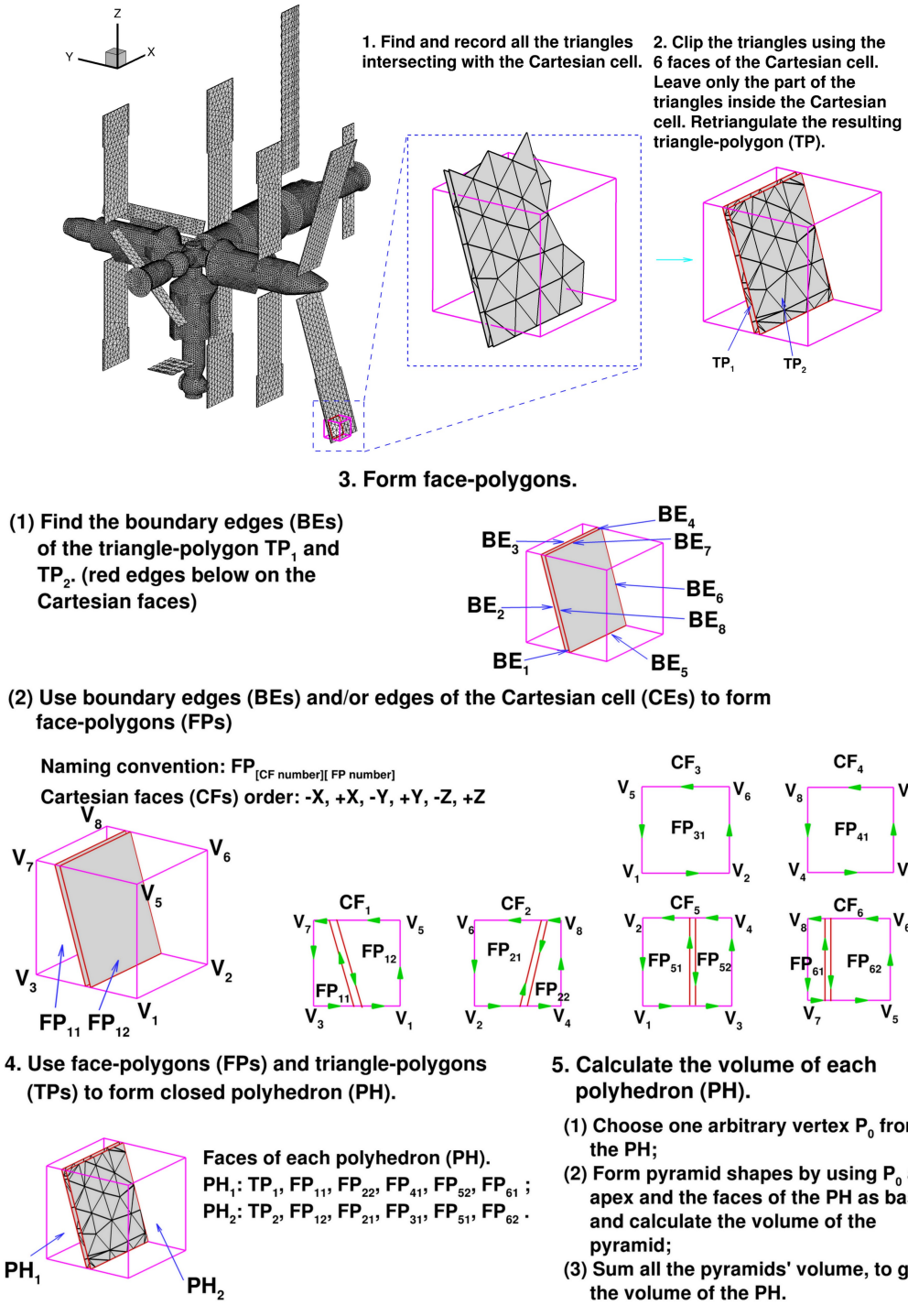


Figure 6.6: A detailed schematic of the entire cut-cell algorithm.

part inside $C_{i,j,k}$, i.e. the triangle-polygon (the details of the S-H algorithm can be found in Ref. [5]).

The triangle-polygon is usually a convex polygon, as the clipping window $C_{i,j,k}$ is convex. To simplify the representation of each triangle-polygon, it is further triangulated. All of the triangle-polygons created from clipping the triangles intersecting with $C_{i,j,k}$, are further grouped together based on their connectivity, to form one or several clipped surface regions inside the Cartesian cell $C_{i,j,k}$. To avoid confusion, each of these surface regions is also called a triangle-polygon, although it is actually a three dimensional surface region represented by triangles. Two triangle-polygons are shown in Fig. 6.6 for the demo split-cell.

The number of triangle-polygons contained in one cut-cell, can help determine the cut-cell type. If the number of triangle-polygons is 1, then there is only one surface region inside the cut-cell, and the cell is a regular cut-cell. On the other hand, if the number of triangle-polygons is greater than 1, the cut-cell may contain only one flow field polyhedron, and is a regular cut-cell, or may contain multiple polyhedrons, as the demo cut-cell shown in Fig. 6.6. The former situation would correspond to Fig. 6.6 if the flow field were to lie in the thin region *between* the surfaces.

6.3.2.2 Forming Face-polygon

As one face-polygon is a two dimensional geometry lying in the Cartesian cell face, and is composed of triangle-polygons' boundary edges and Cartesian cell edges, we use these edges to construct the face-polygon in two steps.

6.3.2.2.1 Finding triangle-polygon's boundary edge As the only difference between the two types of face-polygon is whether they contain any boundary edges of the triangle-polygon, we first determine these boundary edges. Because each triangle-polygon's boundary edge belongs to only one triangle, and is not shared by multiple triangles, it can be found by looping over all the triangles' edges contained in the triangle-polygon, and selecting those edges not shared by multiple triangles. Since each boundary edge is contained in only one of the 6 Cartesian cell faces, after all the boundary edges for each triangle-polygon are found, we further link each boundary edge to one Cartesian cell face. This is achieved by comparing the coordinates of the

boundary edge and the coordinates of the Cartesian cell face.

6.3.2.2.2 Using triangle-polygon's boundary edges and/or Cartesian cell edges to form face-polygon At this point, we have a list of boundary edges contained in each Cartesian cell face. We use these edges and the Cartesian cell edges to form the required face-polygon, which will be a face of the polyhedron to be formed. For a face of Cartesian cell $C_{i,j,k}$ without a boundary edge, the face itself is a type 1 face-polygon as defined earlier. On the other hand, a face containing triangle-polygons' boundary edges will include multiple type 2 face-polygons, each of which contains at least one boundary edge. As our purpose is to form flow-field polyhedron, we only construct flow-field face-polygons, and ignore all the face-polygons inside the surface region. The procedures to form the type 2 face-polygon are discussed below. The input is a list of triangle-polygon's boundary edges $B_i = \{BE_i^1, BE_i^2, ..., BE_i^{n_i}\}$ contained in each Cartesian cell face $CF_i (i = 1, 2, ..., 6)$, the 4 edges $E_i^1, E_i^2, E_i^3, E_i^4$ of each CF_i, and the 8 vertices $V_1, V_2, ..., V_8$ of the Cartesian cell.

(1) For face CF_i with at least one boundary edge, start a new face-polygon FP_m $(1 \leq m \leq$ number of face-polygon in $CF_i)$. Choose an edge BE_i^l $(1 \leq l \leq n_i)$ from B_i, let it be one edge of FP_m, and mark it as used. Denote the starting vertex of the first edge of FP_m as V_0. Following the direction of BE_i^l, its two vertices are named P_1 and P_2, with P_1 the starting vertex and P_2 the ending vertex. Here, the phrase "the direction of BE_i^l" means the direction consistent with the outward normal of the triangle-polygon in which BE_i^l is contained. The meaning of this phrase is implied for the reminder of this part.

(2) Check if P_2 is on an edge E_i^j $(1 \leq j \leq 4)$ of face CF_i. The direction of E_i^j is based on the outward normal of CF_i. Jump to (3) or (4) depending on whether P_2 is on an edge of face CF_i.

(3) If P_2 is on E_i^j, then the next edge of FP_m will be part of E_i^j. There are two situations. Situation 1 is that there is no other unused edge from B_i with a starting vertex on E_i^j. Situation 2 is that there is at least one unused edge from B_i with a starting vertex on E_i^j.

(i) For situation 1, the new edge connects P_2 and V_s ($1 \leq s \leq 8$). V_s is the Cartesian vertex on edge E_i^j that ensures line P_2V_s has the same direction as E_i^j. The new edge is added to the edge list of FP_m. Then, the next edge of FP_m will lie on a different face edge E_i^k ($k \neq j$) with one vertex as V_s. E_i^k is the edge connecting E_i^j and has the same direction as E_i^j when starting from V_s. Return to the beginning of (3) with a different face edge E_i^k and the new starting vertex V_s.

(ii) For situation 2, the new edge connects P_2 and P_3. P_3 is on the edge E_i^j, and is also the starting vertex of another unused boundary edge BE_i^m ($1 \leq m \leq n_i$) from B_i. When there are multiple boundary edges having starting vertex on E_i^j, P_3 is the one nearest to P_2 and satisfies the condition that line P_2P_3 moves in the same direction as E_i^j. Add the new edge BE_i^m to the edge list of CF_m, mark it as used, and return to (2) with P_3.

(iii) Before jumping back to previous step as stated in (i) and (ii), we also need to check if the ending vertex of the newly added edge (vertex V_s in (i) and vertex P_3 in (ii)) is the same as V_0. If yes, the current face-polygon FP_m should be closed and the algorithm returns to (1), and resumes with a new face-polygon.

(4) If P_2 is not on any edge E_i^j ($1 \leq j \leq 4$) of face CF_i, the next edge should still be a boundary edge from B_i. Select from B_i the unused edge BE_i^k ($1 \leq k \leq n_i$) with a starting vertex the same as P_2, add BE_i^k to the edge list of FP_i, and mark BE_i^k as used. If no new edge can be added, continue to (5).

(5) Check if the ending vertex of the newly added edge is the same as vertex V_0.

(i) If yes, then close the current face-polygon FP_m, return to (1) and start a new face-polygon.

(ii) If not, denote the two vertices of the new edge BE_i^k as P_1 and P_2 in order, and return to (2).

(6) After all faces $CF_1, CF_2, ..., CF_6$ with boundary edges have been visited, all type 2 face-polygons are determined. The remaining Cartesian cell faces are type 1 face-polygon.

Examples of constructed face-polygons are shown in Fig. 6.6 for reference. After constructing the face-polygon, its area is calculated using the two dimensional form of Eq. 6.2.

6.3.2.3 Forming Polyhedron

After finding all the face-polygons ($FP_1, FP_2, ..., FP_m$) and all the triangle-polygons ($TP = \{TP_1, TP_2, ..., TP_n\}$) contained in one cut-cell, we form new polyhedron as follows:

(1) Starting from one triangle-polygon (TP_i), add all the triangles in TP_i to the face list (F_s) of the new polyhedron (PH_s), and mark this TP_i as used.

(2) Loop over all the unused face-polygons. If a face-polygon (FP_j) shares a common vertex with the faces in F_s, add FP_j to F_s, and mark it as used.

(3) After no new face-polygon can be added, loop over all the unused elements of TP. For an element TP_j of TP, check if any boundary vertex of TP_j is the same as one vertex of a face in F_s. If yes, add all the triangles in TP_j to F_s, and mark it as used.

(4) After no new element of TP can be added to F_s, the polyhedron PH_s is formed. Continue to form new polyhedron from an unused element of TP and repeat (1), (2), (3), (4).

6.3.2.4 Polyhedron Volume Calculation

The polyhedron contained in a cut-cell has two types of faces, the face-polygon and the triangles contained in the triangle-polygon. With all faces of a polyhedron known, its volume can be calculated by dividing the polyhedron into smaller pyramid shaped polyhedrons, each of which shares a common apex and uses one of the polyhedron face as its base. The process to calculate the volume of each polyhedron is as follows:

(1) Choose a point from the vertex list of the polyhedron (called P_0), and let P_0 be the apex of all the pyramid shaped polyhedrons.

(2) Loop over all triangles contained in the triangle-polygon faces of the polyhedron. For a triangle T_i, let its 3 vertices be P_1, P_2, P_3. Form a pyramid shaped polyhedron (tetrahedron) by using the four vertices P_0, P_1, P_2, P_3, and let P_0 be its apex. Use Eq. 6.2 to get the signed volume of the tetrahedron. Since the triangle has an outward normal direction, and P_0, P_1, P_2 are ordered with this convention, the signed volume we need is $V(P_0, P_1, P_3, P_2)$.

(3) Loop over each face-polygon contained in the polyhedron. For a face-polygon FP_i, calculate the volume of the pyramid formed by P_0 and FP_i. Choose a vertex of FP_i as P_j. Denote the vector formed by P_j and P_0 as $\vec{l}_{P_0P_j}$ pointing from P_0 to P_j, and the outward normal of FP_i as $\vec{n}_i$ (same as the outward normal of the Cartesian face, which FP_i lies in). The volume of the pyramid is $V_{P_0,FP_i} = \frac{1}{3}S_i\vec{l}_{P_0P_j} \cdot \vec{n}_i$, where S_i is the area of FP_i.

(4) Sum over all of the above computed volumes to determine the volume of the polyhedron.

6.3.3 Robust handling of special cases

When calculating the volume of the polyhedron inside a cut-cell using the direct method, we rely on the spatial resolution and relative position of different geometrical objects to construct the correct polyhedron. However, when a surface triangle's vertex is very close to the Cartesian cell face, or a surface triangle's edge is very close to the Cartesian cell edge, the S-H clipping algorithm may result in a very small triangle-polygon when compared with the size of the Cartesian cell itself (possibly many orders of magnitude smaller). In section 6.3.2.2.1, "*Finding triangle polygon's boundary edge*", the algorithm attempts to determine if any of the triangle polygon's edges lie on a Cartesian cell face. A finite tolerance must be used to determine this and if an extremely small triangle polygon is created (by S-H clipping) that lies next to a cell face, it is possible that it is ignored. As a result, the polyhedron volume may not be closed and the algorithms may fail. Such round-off errors are robustly handled with the following simple modification.

Before performing S-H mesh clipping, we first examine the vertices and edges of the surface triangles against the Cartesian cell faces and edges. If the distance between any triangle vertex and a Cartesian cell face (d_1) is smaller than a specified tolerance,

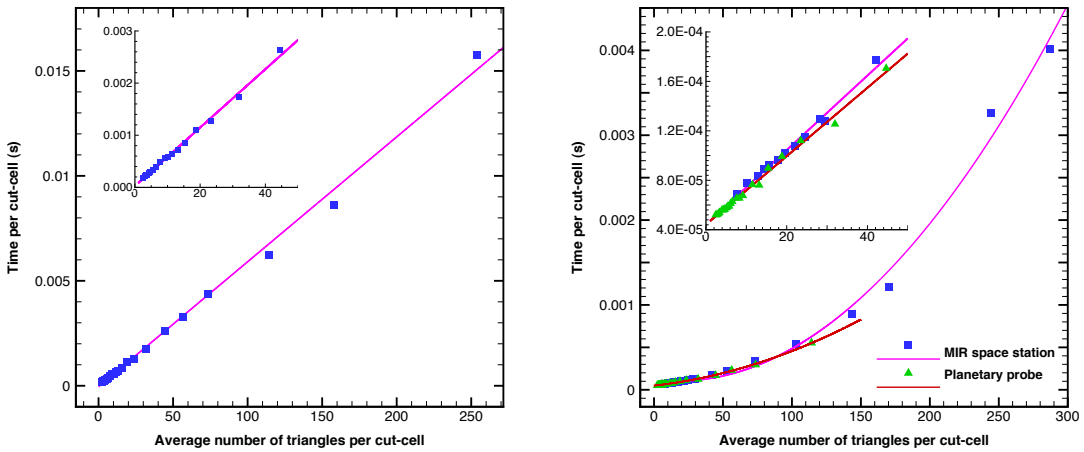
d_ϵ, then the vertex is moved for d_ϵ in the face outward normal direction. Similarly, when the distance between a surface triangle edge and a Cartesian cell edge (d_2) is smaller than d_ϵ, one vertex of the triangle edge is moved in either face outward normal direction for a distance that ensures $d_2 \geq d_\epsilon$. All the surface triangles sharing the same vertex are modified accordingly. This artificially enlarges any triangles that may cause roundoff issues. We use a small tolerance of $d_\epsilon = 10^{-4} \times \Delta x$ (where Δx is the size of the Cartesian cell). Such modification to the surface geometry rarely happens, and due to the small tolerance we set, the polyhedron volume change is completely negligible. Also, this modification is only used to achieve robustness of the volume calculation, and is not used in other parts of the DSMC method such as the ray-trace method and surface property sampling.

6.3.4 Time efficiency of the volume calculation algorithms

As the cut-cell volume calculation is performed for each cut-cell separately, the total time cost of the volume calculation routine scales linearly with the number of cut-cells involved. The time required to calculate one cut-cell volume depends mainly on the number of triangles contained in the cut-cell and the number of face-polygons in the cut-cell. The number of face-polygons in one cut-cell is somewhat related to the number of triangles. As a result, here we only consider the effect of the number of triangles on the volume calculation.

We use both the MIR Space Station and planetary probe geometries and place the surface meshes inside regions of uniform Cartesian grid. Using the cut-cell sorting routine, the triangles are sorted into corresponding Cartesian cells and are ready for cut-cell volume calculation. By uniformly changing the sizes of all Cartesian cells, the number of triangles contained in each cut-cell can be varied, since a smaller Cartesian cell would contain fewer triangles.

Suppose the average number of triangles in each cut-cell is N_T, and the number of cut-cells is N_c. We record the time cost of the cut-cell volume calculation routine for different numbers of Cartesian cells N, which corresponds to different N_c and N_T pairs. We plot the average time required to perform the volume calculation per cut-cell as a function of the average number of triangles contained in each cut-cell in Fig. 6.7 for each geometry. For the Monte Carlo sampled volume technique, only the Probe geometry was



(a) Monte Carlo method using planetary probe as test case.

(b) Direct Method using MIR space station and planetary probe as test cases.

Figure 6.7: Average time cost of calculating the volume of one cut-cell as a function of the average number of triangles contained in the cut-cell.

tested, and the results are plotted in Fig. 6.7(a). The results for both MIR and Probe geometries using the direct volume calculation method are presented in Fig. 6.7(b). It is observed that the Monte Carlo technique scales linearly with N_T, however, takes more than an order of magnitude longer for the same N_T compared with the direct method. Even more importantly, the results presented for the Monte Carlo technique employ only 1000 Monte Carlo points per cell and thus the sampled volumes only approximate the exact volumes. For the direct method, it can be seen from Fig. 6.7(b) that, overall, the time required for the cut-cell volume calculation scales as N_T^2 when N_T is greater than 50. However, if N_T is less than 50, the time required for cut-cell volume calculation scales linearly with N_T. In typical DSMC simulations, $N_T < 50$. As an example, if on average a cut-cell has $N_T = 10$, the time required for one cut-cell volume calculation is approximately 8×10^{-5}s. This means we can compute 12,500 cut-volumes in one second (on the platform noted in Table 6.1).

6.3.5 Cut-cell volume distribution

A cut-cell is only part of the original Cartesian cell, and has a volume ratio of less than 1. The volume ratio is defined as *volume ratio = Cut-cell volume / Cartesian*

cell volume. As the number of particles in each simulation cell is directly related to its volume, a small volume ratio may result in a cut-cell with no enough particles to maintain a correct collision statistics. This may lead to inaccurate simulation result and affect the surface heating rate. To see the cut-cell volume distribution in a typical DSMC simulation, we conduct a simulation of very rarefied flow over the MIR space station. The conditions of the simulation are shown in Table 6.4.

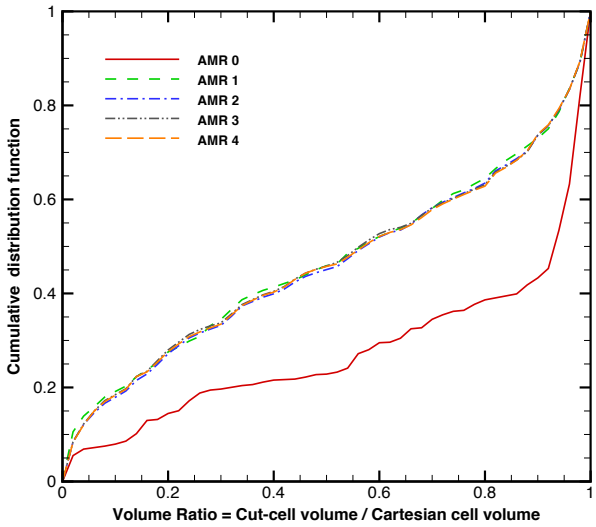


Figure 6.8: The cumulative distribution function (CDF) of the cut-cell volume ratio, where the volume ratio = Cut-cell volume / Cartesian cell volume.

The Cumulative distribution function (CDF) of the cut-cell volume ratio is plotted in Fig. 6.8. The cut-cell volume only include the volume of the flow field part Cartesian cell. This figure describes the distribution of the cut-cell volume relative to the full volume of the original Cartesian cell. A low volume ratio implies a small cut-cell compared to the original full Cartesian cell. This may cause some problems to the DSMC collision particle selection process, and therefore affect the accuracy of the simulation results near the wall/surface region, where all the cut-cells lie. In the figure, "AMR" 0 refers to the initial call to the cut-cell volume calculation routine, "AMR 1" refers to the call to the cut-cell volume calculation routine after the first Adaptive Mesh Refinement (AMR), and "AMR 2 " refers to the call to the cut-cell volume calculation routine after the second AMR, and similarly for all the other lines in the figure. After the first call to

the cut-cell volume calculation routine, the CDF reaches a relative steady distribution, which agrees with the DSMC simulation results that the flow field Cartesian meshes do not change significantly after each AMR calls except the first one. The CDF at each point indicates the percentage of all cut-cells with volume ratio less than a specified value. As can be seen from Fig. 6.8, in the simulation conducted here, 20% of the cut-cells have a volume ratio of less than 0.2. This may affect the accuracy of the simulation result. It should be noted, different simulation may result in a different cut-cell volume distribution. However, further studies are necessary to quantify the effect of small volume ratio cut-cells.

6.3.6 Verification of split-cell volume calculation

In order to identify split-cells and calculate the volume of each split-cell, we use the direct volume calculation method (method 2 in section 6.3.2), since the Monte Carlo sampling technique (method 1 in section 6.3.1) is incapable of detecting spit-cells. To verify the implementation and accuracy of the direct method for split-cells volume calculation, we place a triangulated sphere in the center of a Cartesian cell to create a number of split-cells. Specifically, 8 separated regions are formed inside the flow field. The volume calculation algorithm introduced in this paper is able to identify these split-cells and form 8 separated polyhedrons to represent the 8 flow field regions. Considering that the sphere is actually triangulated, the computed volumes of these regions are confirmed to be very close, but not exactly equal, to the analytical result assuming a perfectly smooth sphere. The radius of the sphere is $R = 0.75dx$, dx is the length of the cube, and $dx = 1$. The computed volumes of the 8 regions are listed in Table 6.2.

6.4 Cut-cell Algorithms Demonstrations

6.4.1 Planetary probe simulation

The cut-cell algorithms discussed in the previous sections are implemented in the MGDS code, a multi-level Cartesian grid based DSMC implementation [43]. Simulation of Mach 20.2, diatomic nitrogen (N_2) gas flow over a spherical-blunted planetary probe

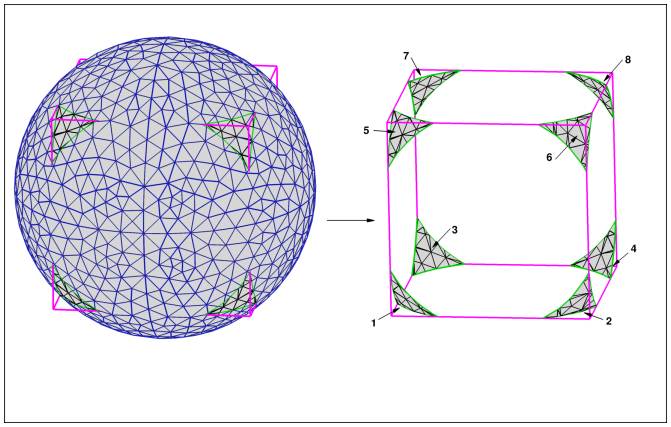


Figure 6.9: Split-cell demonstration. A Cartesian cell is cut-through by a sphere, and the result is a cut-cell with 8 separated flow field regions.

Table 6.2: Calculated split-cell volumes.

Region	Volume ($\times 10^{-3}$)
1	1.6138
2	1.6155
3	1.6164
4	1.6471
5	1.6138
6	1.6155
7	1.6164
8	1.6471

is conducted to demonstrate the capability of the cut-cell algorithms to handle complex boundaries between the surface mesh and the background Cartesian mesh. The conditions of the simulation are listed in Table 6.3, which corresponds to one of the experimental conditions of the CNRS wind tunnel test [7–9]. Here, M_∞, T_∞, ρ_∞, and V_∞, are the free stream Mach number, temperature, density, and velocity, respectively. The wall temperature boundary condition is fixed at a constant value of T_w. The geometry of the planetary probe is shown in Fig. 6.10 [75].

During the DSMC simulation, the three level Cartesian grid is automatically adapted to the local mean free path. Each time the background mesh is adapted, the surface mesh is re-cut from the three level Cartesian grid. All surface triangles are sorted into

Table 6.3: Conditions for the planetary probe simulation

Gas	M_∞	$T_\infty(K)$	$\rho_\infty(kg/m^3)$	$V_\infty(m/s)$	$T_w(K)$
N_2	20.2	13.3	1.73×10^{-5}	1502	300

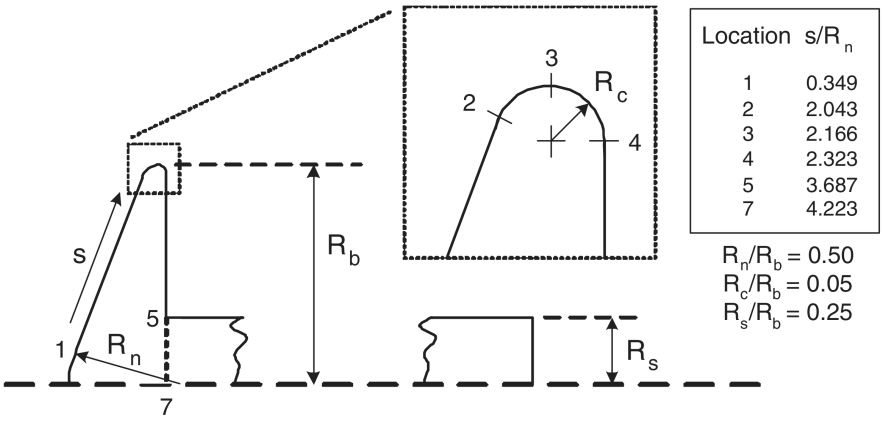


Figure 6.10: The cross-section of the planetary probe at the symmetry plane; the probe has a base diameter of $R_b = 0.05m$.

appropriate cells for use by the ray-trace movement routine and all cut-volumes are computed and stored for use in the collision rate and macroscopic sampling routines. Since for the planetary probe flow, the mean free path becomes quite small near the surface, there are a large number of cut-cells due to the existence of the probe surface. This can be seen from Fig. 6.11, which shows the adapted flow field Cartesian mesh on the symmetry plane of the planetary probe at 10 degrees angle of attack at steady-state. For this case the total number of L3 cells is approximately 622,000. Among these L3 cells, 83,000 of them are cut-cells, and only 15 of these cut-cells are split-cells. The probe surface cuts cells into a wide variety of volumes spanning from large cut-volumes to extremely small cut-volumes, where the surface may barely cut through a Cartesian cell. The collision rates within these cells are dependent on the cut-volumes and the ray-trace algorithm must successfully detect all surface collisions. Thus this test case represents a non-trivial test of the cut-cell algorithms and their incorporation with a DSMC code.

The flow field properties along the stagnation line of the probe at an angle of attack

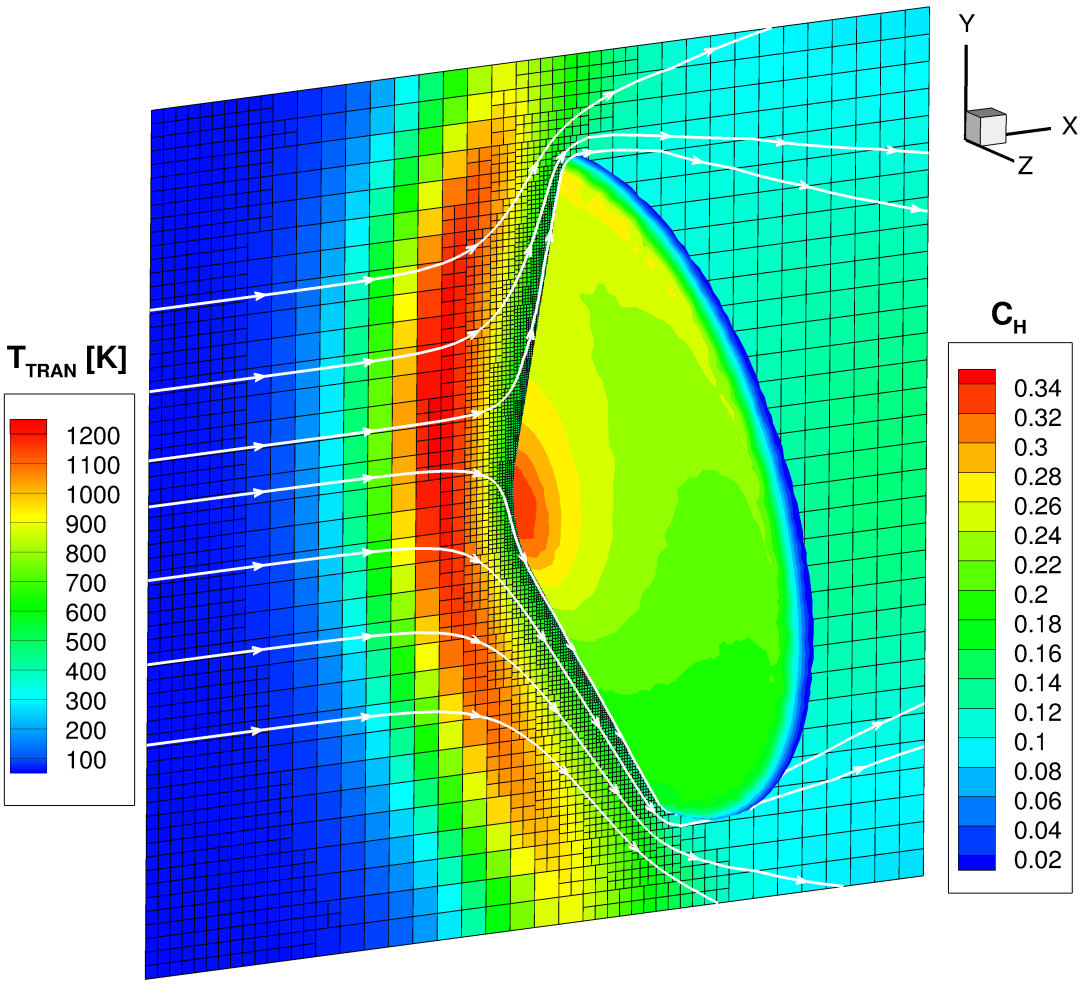


Figure 6.11: The translational temperature along the symmetry plane of the probe at angle of attack $\alpha = 10°$, together with heat transfer coefficient $C_H = 2q/(\rho_\infty V_\infty^3)$, where q is the local heat transfer rate to the probe surface. The cross-section of the Cartesian cells on the symmetry plane of the probe is also shown in the figure by black lines.

$\alpha = 0°$ is plotted in Fig. 6.12. The heat flux along the probe surface on the symmetry plane is also plotted in Fig. 6.13, at the same angle of attack $\alpha = 0°$. Experimental result [9] is also plotted in Fig. 6.13, and good agreement is obtained. Thus the existence of large numbers of cut-cells next to the surface of the planetary probe, does not affect the accuracy of the heat flux prediction.

Furthermore, the density contours are plotted in Fig. 6.14 and Fig. 6.15 for two

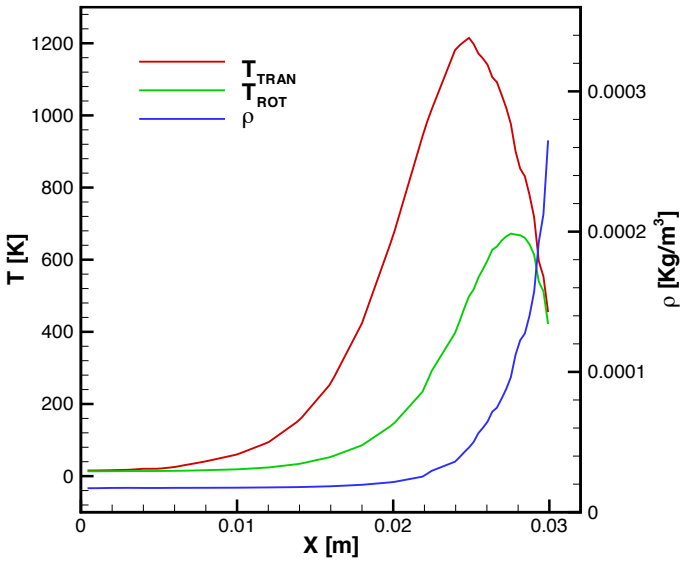


Figure 6.12: The translational temperature T_{TRAN}, rotational temperature T_{ROT} and density ρ, along the stagnation streamline for the probe without sting support at an angle of attack $\alpha = 0°$.

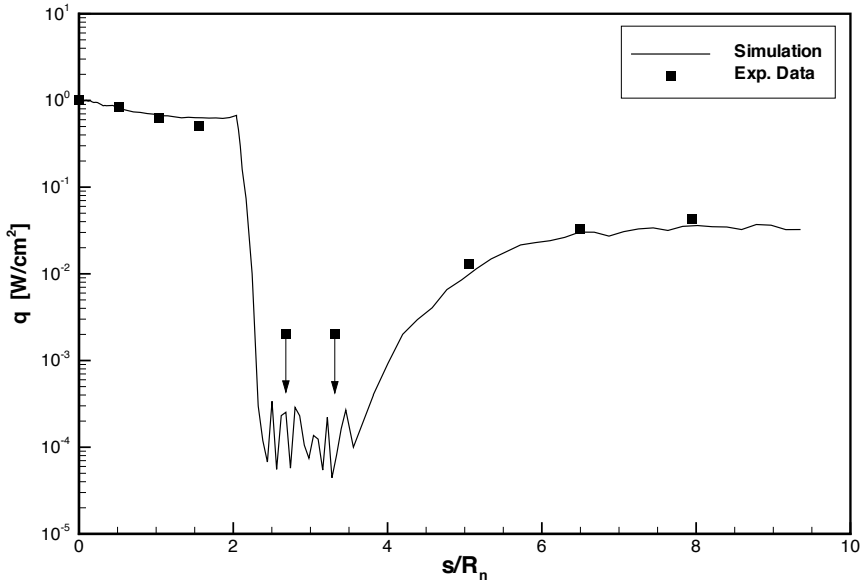


Figure 6.13: The heat flux along the probe surface on the symmetry plane, at an angle of attack $\alpha = 0°$; the closed squares are the experimental data.

different angles of attack conditions. The density contours predicted by DSMC simulation are normalized by the free stream density value, and compared with experimental measurement. Overall, the agreement between DSMC simulation and experimental measurement is very good, which further shows that the use of decoupled flow filed Cartesian mesh and surface triangle mesh, hence the existence of large numbers of cut-cells next to the surface of the planetary probe, does not affect the accuracy of flow field simulation.

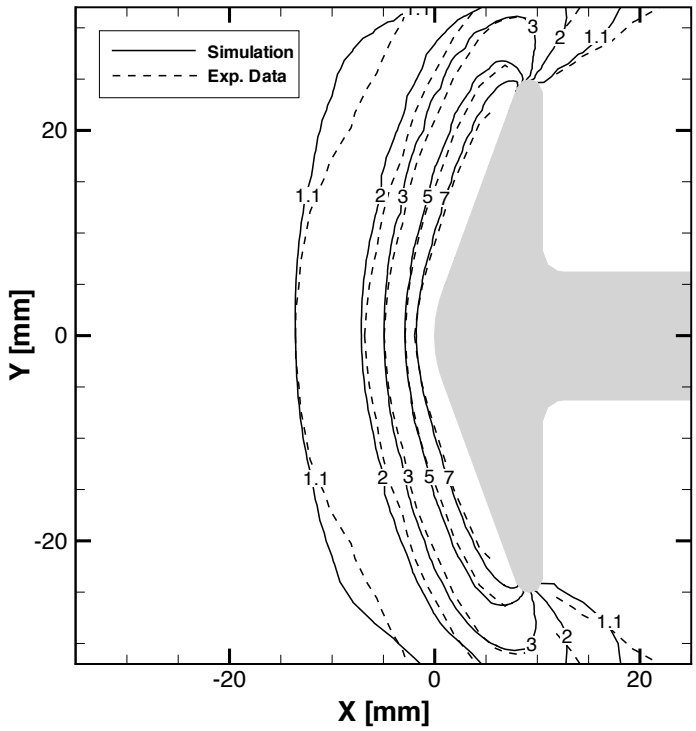


Figure 6.14: Contours of density normalized by the free stream density value. Comparison of DSMC simulation with experimental measurement at an angle of attack $\alpha = 0°$.

6.4.2 MIR space station simulation

To further verify and demonstrate the capability of handling complex geometries using the currently proposed cut-cell algorithms, we conduct a simulation of rarefied flow over the MIR space station geometry. The conditions of the simulation are listed in

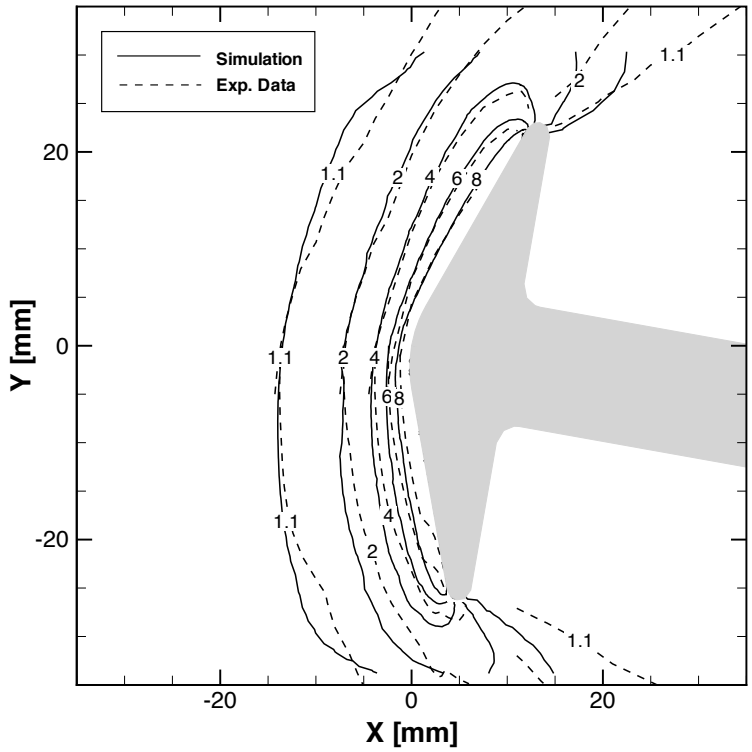


Figure 6.15: Contours of density normalized by the free stream density value. Comparison of DSMC simulation with experimental measurement an angle of attack $\alpha = 10°$.

Table 6.4. Similarly to the planetary probe simulation, AMR combined with the cut-cell algorithms, are used to automatically regenerate the flow field mesh by adaption to the local mean free path, and re-cut the surface mesh during the DSMC simulation. In Fig. 6.16, the final adapted flow-field mesh is plotted on two planes approximately passing through the center of the station. The translational temperature (T) field is also plotted on the two planes while the surface heating rate (q) contours are shown on the MIR Space Station geometry in Fig. 6.16.

Table 6.4: Conditions for the MIR space station simulation

Gas	M_∞	$T_\infty(K)$	$\rho_\infty(kg/m^3)$	$V_\infty(m/s)$	$T_w(K)$	simulation domain size (m)
N_2	20	50	4.5×10^{-8}	2883.4	200	$80 \times 40 \times 40$

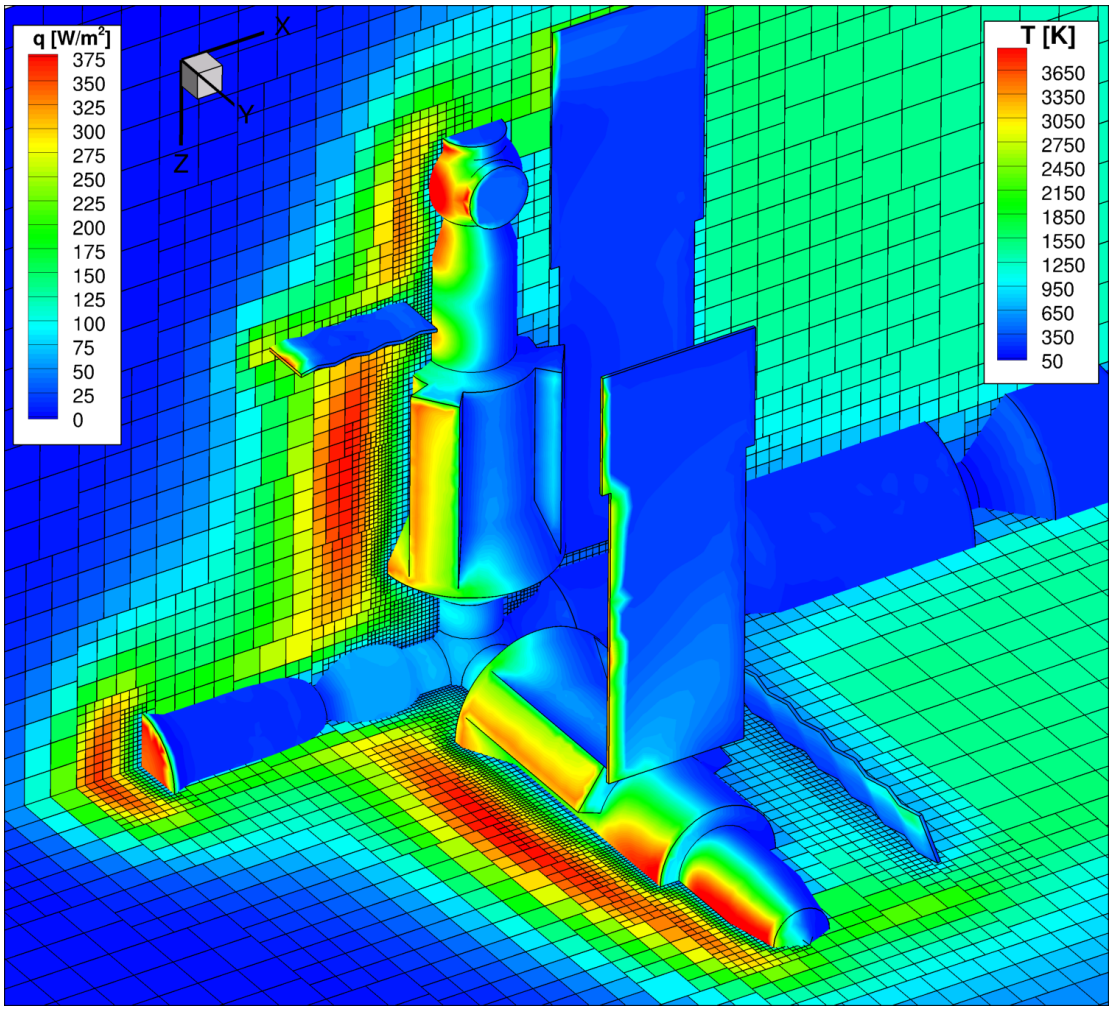


Figure 6.16: The translational temperature (T) on two planes passing roughly through the center of the MIR geometry, along with the heat transfer rate (q, energy transferred per unit area per unit time) to the surface. The flow is moving in the positive x-direction.

As can be seen from Fig. 6.16, the surface heating is relatively low due to the rarefied flow conditions. For this case, the total number of L3 cells in the simulation is approximately 624,000, and the L3 cell size is adapted to 0.56 of the local mean free path everywhere. Among these L3 cells, 57324 of them are cut-cells, and roughly 5572 of these cut-cells are split-cells. Because the flow has a very low density of $\rho = 4.5 \times 10^{-8} kg/m^3$, the mean free path in the flow field is quite large. As a result, the size of the Cartesian

cell is larger than that of the surface triangles in some flow field regions. Such a situation causes the large amount of split-cells in this simulation, when the Cartesian cells are cut-through by the relatively thin triangulated surface components, such as the solar panels. Although there is no experimental data to match this geometry and conditions, the coupling between the DSMC solver and cut-cell routines is found to be robust, and the heat flux contours are smooth with no indication of adverse effects from the cut-volumes.

6.5 Chapter Conclusions

In this chapter, a complete framework of cut-cell algorithms used to handle the decoupled flow field and surface meshes, is detailed for general Cartesian grid based DSMC implementations. A self consistent computer program is written (in Fortran 90) to implement the algorithms discussed in this chapter. The program could be easily incorporated into general three-dimensional Cartesian grid based DSMC simulation code. The cut-cell algorithms are decomposed into two major parts. The first part identifies the cut-cells, and records all the surface triangles intersecting with them. The major task is to determine the intersection of a triangle and a Cartesian cell. For this task, three existing algorithms are detailed and implemented; the signed tetrahedral volume method, the separating axis theorem (SAT) method, and the algorithm of Voorhies. The second major task performed by the algorithms is to calculate the cut-cell volumes. When a Cartesian cell is cut-through by the surface triangles, the resulting cut-cell is usually very complex and poses difficulties in determining its volume in an accurate and robust manner. New algorithms are presented for this task: a Monte Carlo volume sampling technique, and a direct volume calculation technique that forms polyhedron/polyhedrons and then determines the volume directly. All algorithms are tested on various constructed test cases and also for two realistic geometries: a planetary probe geometry and a geometry resembling the MIR Space Station. Algorithms are compared for accuracy and efficiency, and a number of conclusions are drawn.

1) For real geometries and grids, the three sorting algorithms are found to be equally efficient due to the large number of trivial sorting acceptances/rejections that each method handles efficiently. However, this may not be general to all problems and an

example where the methods require significantly different time costs is highlighted. If properly incorporated, the triangulated surface meshes considered can be sorted into Cartesian cells roughly two orders of magnitude faster if using a three level Cartesian grid versus a two level Cartesian grid. Furthermore, if subsequent cut-cell treatment is required during the simulation, for example after periodic adaptive mesh refinement, the time required for sorting algorithms can be dramatically reduced. For subsequent calls, the time required by the sorting algorithm no longer scales with the total number of Cartesian cells, but rather with the total number of surface triangles. Similar efficiency is found if a quick pre-sorting technique is applied prior to calling the sorting algorithm. When pre-sorting and/or subsequent cut-cell operations are employed, the 3 different methods now require significantly different amounts of time, since there are no longer as many trivial acceptances/rejections. In all cases where the 3 methods required different times, the SAT method was most efficient, followed closely by the Voorhies method, and the signed tetrahedral volume method was significantly slower.

2) The first cut-volume algorithm investigated is based on Monte Carlo sampling. It is very easy to implement and robust for complex cut-volumes. However, this method is found to be over an order of magnitude slower than the more direct volume calculation method and only approximates the correct cut-volume to a level corresponding to the number of Monte Carlo sample points used. Furthermore, the Monte Carlo volume sampling is incapable of detecting split-cells. The more direct volume calculation method is verified to be accurate using several test cases and is verified to correctly detect all split-cells and accurately compute these multiple flow field volumes. The time cost for the direct volume calculation method is found to scale linearly with the average number of triangles per cut-cell, N_T, for $N_T < 50$ which is typically the case for a DSMC simulation.

3) The combined sorting routines and cut-volume routines are incorporated into a 3D DSMC code and tested on two complex flow configurations: hypersonic flow over a planetary probe geometry and hypersonic flow over a geometry resembling the MIR Space Station. The DSMC code uses a three-level Cartesian grid and is capable of performing adaptive mesh refinement (AMR) during the simulation in order to adapt the flow field mesh to the local mean free path. At the beginning of each simulation, the surface mesh is cut from the background Cartesian mesh, the triangulated surface

elements are sorted into appropriate Cartesian cells for use by the ray-trace movement algorithm, and the cut-volumes are stored for use in computing local collision rates and sampling macroscopic variables. As the simulation proceeds, AMR is performed periodically, after which the cut-cell algorithms are called to re-cut the surface from the new background Cartesian grid and provide updated data for un-interrupted DSMC simulation. Based on the results of these test cases, the cut-cell algorithms are found to be robust and accurate, as the surface heat flux results are smooth and in agreement with experimental data. It is noted that although the algorithms discussed in this chapter assume a cubic Cartesian cell, extension to non-cubic Cartesian cells requires only minor adjustments to the cut-cell algorithms. The computer program written is very general, and works for both cubic and non-cubic Cartesian cells.

Since most DSMC implementations would cut a surface mesh out of the flow field mesh only a few times (perhaps only once) during the entire simulation, the time costs for both the sorting and cut-volume routines are completely negligible. Even for large parallel simulations, these algorithms could be performed on a single CPU as long as memory is not an issue. Furthermore, since these algorithms only rely on individual Cartesian cells and local triangles, they could be parallelized without much difficulty; further reducing the time cost. What is most interesting is that even for unsteady DSMC simulations, where a surface mesh may need to be re-cut from an ever-changing flow field mesh (frequent adaptive mesh refinement), these algorithms seem fast enough to be feasible. As an example, for the planetary probe geometry, the time required to initially sort the mesh and compute cut-volumes is approximately 60% of the time required for a single DSMC time step. For subsequent cut-cell treatment, the time drops to 57% of a single time step. For the MIR geometry, subsequent cut-cell treatment requires roughly 150% of the time for one DSMC time step. Considering that unsteady DSMC simulations require substantially more computational time per time step than the steady-state flows studied in this paper, the efficiency of the cut-cell algorithms appears quite feasible for enabling unsteady DSMC flows on Cartesian-based grids.

Chapter 7

Conclusions

7.1 Summary of the Thesis

This thesis makes a number of contributions regarding the implementation of advanced physical models in the direct simulation Monte Carlo (DSMC) method. Although DSMC algorithms and collision models for simple monatomic gases have been well-established, physical models for diatomic gas mixtures including rotational-vibrational-chemical nonequilibrium are less established. Specifically, inconsistencies between thermochemical nonequilibrium collision models are evident in the literature. Many models report issues in satisfying detailed balance and equipartition of energy, implementations of these models into the DSMC method are inconsistent and often not reported, the connection between continuum models and advanced DSMC models in the near-equilibrium limit is not rigorously demonstrated, and although state-resolved DSMC models have been presented, a detailed model formulation and implementation has not been presented in the literature. The research comprising this thesis systematically addresses all of these issues and rigorously describes how advanced physical models should be implemented within the DSMC method in a consistent manner and also presents new DSMC algorithms and collision models for gases in thermochemical nonequilibrium.

In Chapter 2, the issue of how particles are randomly selected for collisions within a given DSMC cell during a given time step was investigated. Especially for gas mixtures, the algorithm used to select collision pairs and update the internal energies of either both particles or just one particle involved in the collision, has a direct effect on the

simulated internal energy transfer rates. The same internal energy relaxation model with the same parameter inputs may lead to different simulated energy transfer rates if different selection procedures are used. This subtle, but important, aspect of DSMC is crucial for the transferability of DSMC collision models between DSMC implementations (codes) and also for obtaining consistency with continuum simulations (CFD) in the near-equilibrium limit. Chapter 2 presented a modification to an existing inelastic collision selection procedure, which is referred to as the sequential probability selection procedure. Simple expressions for the inelastic collision probabilities used in DSMC simulations are detailed, which no longer require the solution of a set of quadratic equations for each collision (as previous methods did). The modified procedure was analytically proven to be equivalent to both the original procedure of Haas *et al.* [48] and the modified framework of Gimelshein *et al.* [44]. We then demonstrated the ability to accurately simulate prescribed phenomenological relaxation rates for gas mixtures in thermal nonequilibrium. Accuracy of the modified selection procedure was verified through comparison with analytical solutions for rotational and vibrational isothermal relaxations.

In Chapter 3, existing phenomenological models for rotational-vibrational-chemical nonequilibrium flows were analyzed. Such existing models have not been consistently compared in the literature. Often the selection procedure used is not identified and different combinations of rotation, vibration, and dissociation models have been used in the comparisons, which makes conclusions regarding the differences of a single model (for example, the vibration model only) difficult to interpret. Chapter 3 detailed the existing phenomenological models for modeling rotational, vibrational relaxation, and chemical reaction processes in the DSMC method. Using the sequential probability selection procedure as a consistent framework, existing DSMC models for rotational and vibrational relaxation were implemented in a modular fashion within a three-dimensional Cartesian grid based DSMC code [43]. The rotational and vibrational models were compared with each other using both isothermal and adiabatic relaxation calculations as well as one dimensional stagnation line simulations for hypersonic flow conditions. When used for the stagnation line simulation, all three vibrational relaxation models predicted virtually the same stagnation point pressure and heat flux. Next, three existing dissociation models for the DSMC method were implemented in a modular fashion and compared

using one dimensional stagnation line simulations. When using the same reaction rate constant (k_f), we found that the TCE model give a higher dissociation rate of N_2 compared to the VFD model using a vibrational favoring parameter of $\phi = 3$ [47]. The QK dissociation model predicted the highest N_2 dissociation rate of any model. All models predicted the same stagnation pressure (refer to Table 3.2).

In Chapter 4, the physical mechanisms of rotational excitation and relaxation were analyzed and a new rotational model (for nitrogen) was developed for the DSMC method. Previously, the translational-rotational relaxation rate has been modeled as either a function of the translational temperature or simply modeled as a constant value. However, prior studies have also highlighted that the rate of rotational energy relaxation may be a strong function of the direction towards the equilibrium state (compressing flows vs. expanding flows) and also the magnitude of the deviation from equilibrium. Chapter 4 presented a new nonequilibrium-direction-dependent (NDD) rotational energy exchange model for DSMC and multi-temperature Navier-Stokes methods. The DSMC NDD model is based only on collision-quantities (translational and rotational energy of the collision pair ε_t, ε_r) and reduces to a rotational collision number model in the continuum limit (Eq. 4.5), and is applicable for use with the Jeans relaxation equation in CFD calculations. The model is formulated based on recent Molecular Dynamics (MD) simulations of rotational relaxation in nitrogen (Valentini *et al.*, Phys. Fluids 24, 106101 (2012) [87]) and accounts for the dependence of the relaxation rate on the direction to the equilibrium state. This enables a single parameterization of the model to accurately simulate rotational relaxation in nitrogen for both compressing and expanding flows, unlike the widely used Parker model. The DSMC NDD model is simple to implement compared to previously developed phenomenological models, and has good numerical efficiency. More importantly, it accurately reproduces a range of pure MD solutions including zero dimensional isothermal relaxations, one dimensional normal shock wave simulations, and one dimensional expansion simulations [87]. In developing the new DSMC NDD model, a general form for the energy distribution function that should be sampled for post-collision states (using the Borgnakke-Larsen approach) was also presented. This general formulation ensures detailed balance and equipartition of energy at equilibrium for any collision-quantity based DSMC model and also explains

the behavior of prior rotational models in the literature [1, 21, 22, 33, 95]. Finally, the increased accuracy of a collision-quantity based model compared to a cell-averaged model was demonstrated by comparing rotational energy distribution functions (rdfs) within a shock wave against a pure MD solution. The rdfs simulated by the collision-quantity dependent NDD model were in better agreement with the MD solution, compared to the cell-averaged version of the NDD model.

In Chapter 5, the formulation and implementation of state-to-state models in the DSMC method was investigated. Only very recently have realistic state-to-state models for DSMC been presented in the literature [27, 58], and no rigorous model framework has been presented that addresses microscopic reversibility, detailed balance, the connection to gas transport properties, and the overall consistency of the state-to-state cross section database. Chapter 5 presented a general framework for implementing state-to-state collision models in the DSMC method. First, a simple state-to-state model with analytical transition probabilities (or cross sections) was detailed, where the microscopic reversibility and detailed balance relationships are inherently satisfied [36]. Through the presentation of this simple model, we identified the important factors in developing a general state-to-state collision model. Specifically, the state-to-state collision cross section database should satisfy microscopic reversibility. Especially when the database consists of transition probabilities only, with no information regarding the total cross section (i.e. the viscosity), the individual cross sections may require modification for consistency. One key aspect in developing a state-to-state DSMC collision model, is to propose efficient DSMC algorithms to correctly simulate the specified state-to-state collision cross section and transition rate. This includes the calculation of desired collision rates in the DSMC method, the selection of potential collision pairs, and the procedure to perform actual collision processes. To achieve these tasks in a computationally efficient manner, we proposed the detailed implementation of a general rovibrational state-to-state collision model for the DSMC method. The proposed model was first demonstrated using qualitatively-constructed state-to-state collision probabilities. With the algorithms verified, we then developed a vibrational state-to-state DSMC collision model using the transition probability data provided by the forced harmonic oscillator (FHO) model [3, 68, 77, 78, 82, 96]. The state-to-state DSMC simulation results were verified by corresponding master equation simulation results.

In Chapter 6, the issue of grid generation and complex geometry associated with the DSMC method was investigated. A significant obstacle for new DSMC developers is the fact that the computational grid must be adapted to the local mean-free-path (i.e. the density field), which is a result of the simulation itself. This requires adaptive mesh refinement (AMR) in order to avoid significant user-time for iterative grid generation. The requirement of AMR and also the desirability of decoupling the flow field grid (adapted to the local mean-free-path) from the body-surface mesh naturally leads to the use of Cartesian flow grids and "cut-cell" algorithms to embed complex geometry. Although the implementation of multi-level Cartesian flow grids is straight-forward, general algorithms for cutting complex 3D geometry from a background grid are quite challenging and tedious in terms of implementation. Chapter 6 detailed a complete framework of cut-cell algorithms used to handle the decoupled flow field grids and surface meshes, for general Cartesian grid based DSMC implementations. A self consistent computer program was written (in Fortran 90) to implement the algorithms discussed in this chapter. The cut-cell algorithms were decomposed into two major parts. The first part identifies the cut-cells, and records all the surface triangles intersecting with them. The major task was to determine the intersection of a triangle and a Cartesian cell. For this task, three existing algorithms were detailed and implemented; the signed tetrahedral volume method, the separating axis theorem (SAT) method, and the algorithm of Voorhies. The second major task was to calculate the cut-cell volumes. When a Cartesian cell is cut-through by surface triangles, the resulting cut-cell can be very complex and poses difficulties in determining its volume in an accurate and robust manner. New algorithms were presented for this task: a Monte Carlo volume sampling technique, and a direct volume calculation technique that forms polyhedron/polyhedrons and then determines the volume directly. All algorithms were tested on various constructed test cases and also for two realistic geometries: a planetary probe geometry and a geometry resembling the MIR Space Station.

7.2 Contributions of the Thesis

In this thesis, we have made the following contributions:

1. We developed a modification to an existing inelastic collision selection procedure

suitable for modeling the internal energy exchange processes of gas mixtures in direct simulation Monte Carlo (DSMC). The model is simple to implement, computationally efficient, and if adopted would enable more consistent transferability of DSMC collision models between DSMC codes and more consistent comparisons with CFD solutions in the near-equilibrium limit.

2. Using the above modified selection procedure, we implemented a range of existing DSMC phenomenological models for rotational/vibrational relaxation and dissociation reactions in a three dimensional DSMC code. The models were compared in a consistent manner through isothermal and adiabatic relaxation calculations, as well as 1D stagnation line simulations for hypersonic flow.

3. We developed a new rotational energy exchange model for both the DSMC method and multi-temperature Navier-Stokes methods. The DSMC model (called the NDD model) is based only on collision-quantities and reduces to a rotational collision number in the continuum limit, applicable for use with the Jeans relaxation equation in CFD. The model was formulated based on recent Molecular Dynamics (MD) simulations of rotational relaxation in nitrogen (Valentini *et al.*, Phys. Fluids 24, 106101 (2012)) and accounts for the dependence of the relaxation rate on the direction to the equilibrium state. This enables a single parameterization of the model to accurately simulate rotational relaxation in both compressing and expanding flows, unlike the widely used Parker model.

4. While developing the DSMC NDD model above, a general form for the energy distribution function that should be sampled for post-collision states (using the popular Borgnakke-Larsen approach) was developed. This general formulation ensures detailed balance and equipartition of energy at equilibrium for any collision-quantity based DSMC model and also explains the behavior of prior rotational models presented in the literature.

5. We developed a general framework for implementing rovibrational state-to-state collision models in the DSMC method that achieves microscopic reversibility, detailed balance, and equipartition of energy under equilibrium conditions. The framework links the state-to-state cross sections to the gas transport properties,

or if the cross section database is incomplete, allows for the incorporation of a desired phenomenological viscosity model. Three implementations were considered in order to achieve computational efficiency and the ability to simulate rare transition events.

6. Using the state-to-state model framework above, we developed a vibrational state-to-state DSMC collision model using the transition probability data provided by the forced harmonic oscillator (FHO) model [3, 68, 77, 78]. DSMC simulation results were verified by corresponding master equation simulation results.

7. To handle the automatic grid generation of complex geometries coupled with multi-level Cartesian grids, a set of robust cut-cell algorithms were developed and implemented. The algorithms are robust for very complex 3D geometries and computationally efficient.

Bibliography

[1] T. Abe. Direct Simulation Monte Carlo Method for Internal-Translational Energy Exchange in Nonequilibrium Flow. In *Rarefied Gas Dynamics - Theory and Simulations; Proceedings of the 18th International Symposium on Rarefied Gas Dynamics*, pages 103–113. University of British Columbia, Vancouver, Canada, 1994.

[2] T. Abe. Inelastic Collision Model for Vibrational-Translational and Vibrational-Vibrational Energy Transfer in the Direct Simulation Monte Carlo Method. *Physics of Fluids*, 6(9):3175–3179, 1994.

[3] I. V. Adamovich, S. O. Macheret, J. W. Rich, and C. E. Treanor. Vibrational Energy Transfter Rates Using a Forced Harmonic Oscillator Model. *Jounal of Thermophysics and Heat Transfer*, 12(1):57–65, 1998.

[4] M. J. Aftosmis. *Solution Adaptive Cartesian Grid Methods for Aerodynamic Flows with Complex Geometries*. Lecture notes for 28th Computational Fluid Dynamics Lecture Series, Von Karman Institute of Fluid Dynamics, Belgium, 1997.

[5] M. J. Aftosmis, M. J. Berger, and J. E. Melton. Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry. *AIAA Journal*, 36(6):952–960, 1998.

[6] T. Akenine-Möller. Fast 3D Triangle-Box Overlap Testing. *Journal of Graphics Tools*, 6(1):29–33, 2001.

[7] J. Allegre, D. Bisch, and J. C. Lengrand. Experimental Rarefied Aerodynamic

Forces at Hypersonic Conditions over 70-Degree Blunted Cone. *Journal of Spacecraft and Rockets*, 34(6):719–723, 1997.

[8] J. Allegre, D. Bisch, and J. C. Lengrand. Experimental Rarefied Density Flowfields at Hypersonic Conditions over 70-Degree Blunted Cone. *Journal of Spacecraft and Rockets*, 34(6):714–718, 1997.

[9] J. Allegre, D. Bisch, and J. C. Lengrand. Experimental Rarefied Heat Transfer at Hypersonic Conditions over 70-Degree Blunted Cone. *Journal of Spacecraft and Rockets*, 34(6):724–728, 1997.

[10] B. K. Annis and A. P. Malinauskas. Temperature Dependence of Rotational Collision Numbers from Thermal Transpiration. *The Journal of Chemical Physics*, 54(11):4763–4768, 1971.

[11] J. P. Appleton, M. Steinberg, and D. J. Liquornik. Shock-Tube Study of Nitrogen Dissociation using Vacuum-Ultraviolet Light Absorption. *The Journal of Chemical Physics*, 48(2):599–608, 1968.

[12] D. Baganoff and R. K. Hanson. Shock-tube Study of Nitrogen Dissociation Rates using Pressure Measurements. *AIAA Journal*, 10(2):211–215, 1972.

[13] F. Bergemann and I. D. Boyd. New Discrete Vibrational Energy Model for the Direct Simulation Monte Carlo Method. In *Rarefied gas dynamics: experimental techniques and physical systems*. AIAA, 1994.

[14] G. D. Billing and L. Wang. Semiclassical Calculations of Transport Coefficients and Rotational Relaxation of Nitrogen at High Temperatures. *The Journal of Physical Chemistry*, 96(6):2572–2575, 1992.

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

[16] G. A. Bird. *Sophisticated DSMC*. DSMC07 Workshop, Sante Fe, 2007.

[17] G. A. Bird. The Q-K model for gas-phase chemical reaction rates. *Physics of Fluids*, 23:106101, 2011.

[18] J. Bonet and J. Peraire. An Alternating Digital Tree (ADT) Algorithm for 3D Geometric Searching and Intersection Problems. *International Journal for Numerical Methods in Engineering*, 31:1–17, 1991.

[19] C. Borgnakke and P. S. Larsen. Statistical Collision Model for Monte Carlo Simulation of Polyatomic Gas Mixture. *Journal of Computational Physics*, 18(4):405–420, 1975.

[20] J. Bourgat, L. Desvillettes, P. Letallec, and B. Perthame. Microreversible Collisions for Polyatomic Gases and Boltzmann's Theorem. *European Journal of Mechanics B*, 13(2):237–254, 1994.

[21] I. D. Boyd. Analysis of Rotational Nonequilibrium in Standing Shock Waves of Nitrogen. *AIAA Journal*, 28(11):1997–1999, 1990.

[22] I. D. Boyd. Rotational-Translational Energy Transfer in Rarefied Nonequilibrium Flows. *Physics of Fluids A*, 2:447–552, 1990.

[23] I. D. Boyd. Analysis of vibrational-translational energy transfer using the direct simulation Monte Carlo method. *Physics of Fluids A*, 3(7):1785–1971, 1991.

[24] I. D. Boyd. Analysis of vibration-dissociation-recombination processes behind strong shock waves of nitrogen. *Physics of Fluids A*, 4(1):178–185, 1992.

[25] I. D. Boyd. Relaxation of Discrete Rotational Energy Distributions using a Monte Carlo Method. *Physics of Fluids A: Fluid Dynamics*, 5(9):2278–2286, 1993.

[26] I. D. Boyd. Temperature Dependence of Rotational Relaxation in Shock Waves of Nitrogen. *Journal of Fluid Mechanics*, 246:343–360, 1993.

[27] I. D. Boyd and E. Josyula. State Resolved Vibrational Relaxation Modelling for Strongly Nonequilibrium Flows. *Physics of Fluids*, 23:7101, 2011.

[28] D. Bruno, M. Capitelli, V. Cervellera, S. Longo, E. V. Kustova, and E. A. Nagnibeda. Calculation of Transport Coefficients with Vibrational Nonequilibrium. *Journal of Thermophysics and Heat Transfer*, 15(1):70–75, 2001.

[29] D. Bruno, M. Capitelli, and S. Longo. DSMC Modelling of Vibrational and Chemical Kinetics for a Reacting Gas Mixture. *Chemical Physics Letters*, 289(1-2):141–149, 1998.

[30] E. H. Carnevale, C. Carey, and G. Larson. Ultrasonic Determination of Rotational Collision Numbers and Vibrational Relaxation Times of Polyatomic Gases at High Temperatures. *The Journal of Chemical Physics*, 47(8):2829–2835, 1967.

[31] G. Chaban, R. Jaffe, D. W. Schwenke, and W. Huo. Dissociation Cross Sections and Rate Coefficients for Nitrogen from Accurate Theoretical Calculations. *AIAA paper*, 2008-1209.

[32] S. Chapan and T. G. Cowling. *The Mathematical Theory of Non-uniform Gases.* Cambridge University Press, London, 1960.

[33] I. Choquet. Thermal Nonequilibrium Modelling using the Direct Simulation Monte Carlo Method: Application to Rotational Energy. *Physics of Fluids*, 6:4042–4053, 1994.

[34] I. Choquet. Vibrational Nonequilibrium Modelling using Direct Simulation. Part I: Continuous Internal Energies. *Journal of Thermophysics and Heat Transfer*, 9:446, 1995.

[35] S. Dietrich and I. D. Boyd. Scalar and Parallel Optimized Implementation of the Direct Simulation Monte Carlo Method. *Journal of Computational Physics*, 126(2):328 – 342, 1996.

[36] S. M. Dunn and J. B. Anderson. Direct monte carlo simulation of chemical reaction systems: Internal energy transfer and an energy-dependent unimolecular reaction. *The Journal of Chemical Physics*, 99(9):6607–6612, 1993.

[37] F. Esposito, I. Armenise, and M. Capitelli. N-N2 state to state vibrational-relaxation and dissociation rates based on quasiclassical calculations. *Chemical Physics*, 331(1):1–8, 2006.

[38] F. Esposito, I. Armenise, G. Capitta, and M. Capitelli. O-O2 state-to-state vibrational relaxation and dissociation rates based on quasiclassical calculations. *Chemical Physics*, 351:91–98, 2008.

[39] K. Fujita and T. Abe. Coupled Rotational-Vibration-Dissociation Kinetics of Nitrogen using QCT Models. In *36th AIAA Thermophysics Conference, AIAA 2003-3779*, Orlando, Florida, 2003.

[40] M. A. Gallis, R. B. Bond, and J. R. Torczynski. A kinetic-theory approach for computing chemical-reaction rates in upper-atmosphere hypersonic flows. *The Journal of Chemical Physics*, 131(12):124311, 2009.

[41] M. A. Gallis, R. B. Bond, and J. R. Torczynski. Assessment of Collisional-Energy-Based Models for Atmospheric Species Reactions in Hypersonic Flows. *Journal of Thermophysics and Heat Transfer*, 24(2):241–253, 2010.

[42] G. Ganzi and S. I. Sandler. Determination of Thermal Transport Properties from Thermal Transpiration Measurements. *The Journal of Chemical Physics*, 55(1):132–140, 1971.

[43] D. Gao, C. Zhang, and T. E. Schwartzentruber. Particle Simulations of Planetary Probe Flows Employing Automated Mesh Refinement. *Journal of Spacecraft and Rockets*, 48(3):397–405, May-June 2011.

[44] N. E. Gimelshein, S. F. Gimelshein, and D. A. Levin. Vibrational Relaxation Rates in the Direct Simulation Monte Carlo Method. *Physics of Fluids*, 14(12):4452–4455, 2002.

[45] S. F. Gimelshein, N. E. Gimelshein, D. A. Levin, M. S. Ivanov, and I. J. Wysong. On the use of Chemical Reaction Rates with Discrete Internal Energies in the Direct Simulation Monte Carlo Method. *Physics of Fluids*, 16(7):2442–2451, 2004.

[46] S. F. Gimelshein, M. S. Ivanov, G. N. Markelov, and Y. E. Gorbachev. Determination of the Scalar Friction Factor for Non-spherical Particles and Aggregates Across the Entire Knudsen Number Range by Direct Simulation Monte Carlo (DSMC). *Journal of Thermophysics and Heat Transfer*, 12(4):489–495, 1998.

[47] B. L. Haas and I. D. Boyd. Models for Direct Monte Carlo Simulation of Coupled Vibration-Dissociation. *Physics of Fluids A*, 5:478, 1993.

[48] B. L. Haas, D. B. Hash, G. A. Bird, F. E. Lumpkin, and H. A. Hassan. Rates of Thermal Relaxation in Direct Simulation Monte Carlo Methods. *Physics of Fluids*, 6(6):2191–2201, 1994.

[49] D. B. Hash, H. A. Hassan, and J. N. Moss. Direct Simulation of Diatomic Gases using the Generalized Hard Sphere Model. *Journal of Thermophysics and Heat Transfer*, 8(4):758–764, 1994.

[50] R. N. Healy and T. S. Storvick. Rotational Collision Number and Eucken Factors from Thermal Transpiration Measurements. *The Journal of Chemical Physics*, 50(3):1419–1427, 1969.

[51] G. Herzberg. *Molecular Spectra and Molecular Structure: Spectra of Diatomic Molecules*. D. Van Nostrand Company, Inc., Princeton, N.J., 1950.

[52] C. N. Hinshelwood. *The Kinetics of Chemical Change*. Oxford University Press, Oxford, UK, 1940.

[53] M. S. Ivanov and S. F. Gimelshein. Computational Hypersonic Rarefied Flows. *Annual Review of Fluid Mechanics*, 30(1):469–505, 1998.

[54] M. S. I. M. Ivanov, A. Kashkovsky, S. Gimelshein, G. Markelov, A. Alexeenko, Y. Bondar, G. Zhukova, S. Nikiforov, and P. Vaschenkov. SMILE System for 2D/3D DSMC Computations. *Proceedings of the 25th International Symposium on Rarefied Gas Dynamics*, July 2006.

[55] R. Jaffe, D. W. Schwenke, G. Chaban, and W. Huo. Vibrational and Rotational Excitation and Relaxation of Nitrogen from Accurate Theoretical Calculations. *AIAA paper*, 2008-1208.

[56] R. B. B. Joseph O. Hirschfelder, Charles Francis Curtiss. *Molecular Theory of Gases and Liquids*. Wiley, New York, 1965.

[57] D. Kewley and H. Hornung. Free-piston Shock-tube Study of Nitrogen Dissociation. *Chemical Physics Letters*, 25(4):531 – 536, 1974.

[58] J. G. Kim and I. D. Boyd. State-resolved master equation analysis of thermochemical nonequilibrium of nitrogen. *Chemical Physics*, 415(0):237 – 246, 2013.

[59] P. Kistemaker, A. Tom, and A. D. Vries. Rotational Relaxation Numbers for the Isotopic Molecules of N2 and Co. *Physica*, 48(3):414–424, 1970.

[60] K. Koura. Statistical Inelastic Cross-section Model for the Monte Carlo Simulation of Molecules with Discrete Internal Energy. *Physics of Fluids A: Fluid Dynamics*, 4(8):1782–1788, 1992.

[61] K. Koura. Statistical Inelastic Cross-section Model for the Monte Carlo Simulation of Molecules with Continuous Internal Energy. *Physics of Fluids A: Fluid Dynamics*, 5(3):778–780, 1993.

[62] K. Koura. A Set of Model Cross Sections for the Monte Carlo Simulation of Rarefied Real Gases: Atom–Diatom Collisions. *Physics of Fluids*, 6(10):3473–3486, 1994.

[63] K. Koura. A Generalization for Parker Rotational Relaxation Model based on Variable Soft Sphere Collision Model. *Physics of Fluids*, 8(5):1336–1337, 1996.

[64] G. J. LeBeau. A Parallel Implementation of the Direct Simulation Monte Carlo Method. *Computer Methods in Applied Mechanics and Engineering*, 174:319–377, 1999.

[65] G. J. LeBeau and F. E. Lumpkin. Application Highlights of the DSMC Analysis Code (DAC) Software for Simulating Rarefied Flows. *Computer Methods in Applied Mechanics and Engineering*, 191:595–609, 2001.

[66] J. A. Lordi and R. E. Mates. Rotational Relaxation in Nonpolar Diatomic Gases. *Physics of Fluids*, 13(2):291–308, 1970.

[67] F. E. Lumpkin, B. L. Haas, and lain D. Boyd. Resolution of Differences Between Collision Number Definitions in Particle and Continuum Simulations. *Physics of Fluids A*, 3(9):2282–2284, 1991.

[68] S. O. Macheret and I. V. Adamovich. Semiclassical Modeling of State-specific Dissociation Rates in Diatomic Gases. *Jounal of Chemical Physics*, 113(17):7351–7361, 2000.

[69] R. C. Millikan and D. R. White. Systematics of Vibrational Relaxation. *Journal of Chemical Physics*, 39(12):3209–3213, 1963.

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

[71] C. Nyeland and G. D. Billing. Transport Coefficients of Diatomic Gases: Internal-state Analysis for Rotational and Vibrational Degrees of Freedom. *The Journal of Physical Chemistry*, 92(7):1752–1755, 1988.

[72] M. Panesi, R. L. Jaffe, D. W. Schwenke, and T. E. Magin. Rovibrational internal energy transfer and dissociation of $N_2(^1\Sigma_g^+)$ - $N(^4S_u)$ system in hypersonic flows. *The Journal of Chemical Physics*, 138(4):044312, 2013.

[73] C. Park. Review of Chemical-Kinetic Problems of Future NASA Missions, I: Earth Entries. *Journal of Thermophysics and Heat Transfer*, 7(3):385–398, 1993.

[74] J. G. Parker. Rotational and Vibrational Relaxation in Diatomic Gases. *Physics of Fluids*, 2(4):449–462, 1959.

[75] T. E. Schwartzentruber, L. C. Scalabrin, and I. D. Boyd. Multiscale Particle-Continuum Simulations of Hypersonic Flow over a Planetery Probe. *Jounal of Spacecraft adn Rockets.*, 45(6):11–31, 2008.

[76] R. Sharafutdinov, A. Belikov, M. Strekalov, and A. Storozhev. State-to-state Rate Constants and Rotational Relaxation Time in Nitrogen. *Chemical Physics*, 207(1):193–201, 1996.

[77] M. L. Silva, V. Guerra, and J. Loureiro. Nonequilibrium Dissociation Processes in Hyperbolic Atmospheric Entries. *Journal of Thermophysics and Heat Transfer*, 21(2):303–310, 2007.

[78] M. L. Silva, V. Guerra, and J. Loureiro. State-Resolved Dissociation Rates for Extremely Nonequilibrium Atmospheric Entries. *Journal of Thermophysics and Heat Transfer*, 21(1):40–49, 2007.

[79] G. O. Sitz and R. L. Farrow. Pump–probe Measurements of State-to-state Rotational Energy Transfer Rates in N_2 (v=1). *The Journal of Chemical Physics*, 93(11):7883–7893, 1990.

[80] J. I. Steinfeld, J. S. Francisco, and W. L. Hase. *Chemical Kinetics and Dynamics*. Prentice Hall, Englewood Cliffs, N.J., 1998.

[81] T. Tokumasu and Y. Matsumoto. Dynamic Molecular Collision (DMC) Model for Rarefied Gas Flow Simulations by the DSMC Method. *Physics of Fluids*, 11(7):1907–1920, 1999.

[82] C. E. Treanor. Vibrational Energy Transfer in High-Energy Collisions. *The Journal of Chemical Physics*, 43(2):532–538, 1965.

[83] C. E. Treanor and P. V. Marrone. Effect of Dissociation on the Rate of Vibrational Relaxation. *Physics of Fluids*, 5(9):1022–1026, 1962.

[84] P. Valentini and T. E. Schwartzentruber. A Combined Event-Driven/Time-Driven Molecular Dynamics Algorithm for the Simulation of Shock Waves in Rarefied Gases. *Journal of Computational Physics*, 228(23):8766–8778, Dec 2009.

[85] P. Valentini and T. E. Schwartzentruber. Large-scale Molecular Dynamics Simulations of Normal Shock Waves in Dilute Argon. *Physics of Fluids*, 21(6):066101, June 2009.

[86] P. Valentini, P. A. Tump, C. Zhang, and T. E. Schwartzentruber. Molecular Dynamics Simulations of Shock Waves in Mixtures of Noble Gases. *Journal of Thermophysics and Heat Transfer*, 27(2):226–234, 2013.

[87] P. Valentini, C. Zhang, and T. E. Schwartzentruber. Molecular Dynamics Simulation of Rotational Relaxation in Nitrogen: Implications for Rotational Collision Number Models. *Physics of Fluids*, 24(10):106101, 2012.

[88] P. Vijayakumar, Q. Sun, and I. D. Boyd. Vibrational-Translational Energy Exchange Models for the Direct Simulation Monte Carlo Method. *Physics of Fluids*, 11(8):2117–2126, 1999.

[89] W. G. Vincenti and J. C. H. Kruger. *Introduction to Physical Gas Dynamics*. Wiley, New York, 1967.

[90] D. Voorhies. Triangle-Cube Intersection. In *Graphic Gems III*, pages 236–239. Academic Press, 1992.

[91] D. C. Wadsworth and I. J. Wysong. Vibrational favoring Effect in DSMC Dissociation Models. *Physics of Fluids*, 9(12):3873–3884, 1997.

[92] D. L. Willauer and P. L. Varghese. Direct Simulation of Rotational Relaxation using State-to-state Cross Sections. *Journal of Thermophysics and Heat Transfer*, 7(1):49 – 54, 1993.

[93] D. L. Willauer and P. L. Varghese. Direct Simulation of Rotational Relaxation using State-to-state Cross Sections. *Journal of Thermophys and Heat Transfer*, 7(1):49–54, 1993.

[94] I. J. Wysong, R. A. Dressler, Y. H. Chiu, and I. D. Boyd. Direct Simulation Monte Carlo Dissociation Model Evaluation: Comparison to Measured Cross Sections. *Journal of Thermophysics and Heat Transfer*, 16(1):83–93, 2002.

[95] I. J. Wysong and D. C. Wadsworth. Assessment of Direct Simulation Monte Carlo Phenomenological Rotational Relaxation Models. *Physics of Fluids*, 10(11):2983–2994, 1998.

[96] A. Zelechow, D. Rapp, and T. E. Sharp. Vibrational–Vibrational–Translational Energy Transfer between Two Diatomic Molecules. *The Journal of Chemical Physics*, 49(1):286–299, 1968.

[97] C. Zhang and T. E. Schwartzentruber. Numerical Assessment of Vibration and Dissociation Models in DSMC for Hypersonic Stagnation Line Flows. In *43rd AIAA Thermophysics Conference, AIAA 2012-2992*, New Orleans, Louisiana, 2012.

[98] C. Zhang and T. E. Schwartzentruber. Robust Cut-cell Algorithms for DSMC Implementations Employing Multi-level Cartesian Grids. *Computers & Fluids*, 69:122–135, 2012.

[99] C. Zhang, T. Thajudeen, C. Larriba, T. E. Schwartzentruber, and C. J. H. Jr. Determination of the Scalar Friction Factor for Non-spherical Particles and Aggregates Across the Entire Knudsen Number Range by Direct Simulation Monte Carlo (DSMC). *Aerosol Science and Technology*, 46(10):1065–1078, 2012.

[100] C. Zhang, P. Valentini, and T. E. Schwartzentruber. A Nonequilibrium-Direction-Dependent Rotational Energy Model for use in Continuum and Stochastic Molecular Simulation. In *51st AIAA Aerospace Sciences Meeting, AIAA 2013-1202*, Grapevine, Texas, 2013.

Appendix A

Determination of Rotational Inelastic Collision Probability in NDD Model

A.1 Determination of $\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r)$ from rotational collision number $Z_{rot}(T_t, T_r)$

As discussed in Chapter 4, to get the collision energy dependent inelastic collision probability for DSMC simulation, we need to solve the following equation, and determine a unique solution for $\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r)$. What follows in this section is a derivation to determine $\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r)$.

$$\frac{1}{Z_r^{\infty}}\left(1+\frac{T^*}{T_t}\right)\left(\frac{T_t}{T_r}\right)^n = \int_0^{\infty}\int_0^{\infty} \tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t, \varepsilon_r; T_t, T_r) d\varepsilon_t d\varepsilon_r \tag{A.1}$$

Here, the joint distribution function of the collision energies $(\varepsilon_t, \varepsilon_r)$ is $f(\varepsilon_t, \varepsilon_r; T_t, T_r)$, and is given in Eq. 4.7. The probability that a collision pair has energies which lie in the range $[\varepsilon_t, \varepsilon_t + d\varepsilon_t]$ and $[\varepsilon_r, \varepsilon_r + d\varepsilon_r]$ is $f(\varepsilon_t, \varepsilon_r; T_t, T_r) d\varepsilon_t d\varepsilon_r$, and the joint distribution function satisfies the following equation,

$$\int_0^{\infty}\int_0^{\infty} f(\varepsilon_t, \varepsilon_r; T_t, T_r) d\varepsilon_t d\varepsilon_r = 1 \ . \tag{A.2}$$

For a quantity Q associated with the collision process that depends on both ε_t and ε_r, the average of Q is,

$$< Q >= \int_0^\infty \int_0^\infty Q(\varepsilon_t, \varepsilon_r) f(\varepsilon_t, \varepsilon_r; T_t, T_r) d\varepsilon_t d\varepsilon_r \ . \tag{A.3}$$

Specifically, assuming that Q can be written in the form,

$$Q(\varepsilon_t, \varepsilon_r) = A\varepsilon_t^l \varepsilon_r^m \ , \tag{A.4}$$

where A, m and n are some real constants, we have, by substituting the appropriate expressions,

$$\begin{aligned} < Q >= & \int_0^\infty \int_0^\infty A\varepsilon_t^l \varepsilon_r^m f(\varepsilon_t, \varepsilon_r; T_t, T_r) d\varepsilon_t d\varepsilon_r \\ = & A \int_0^\infty \varepsilon_t^l f(\varepsilon_t; T_t) d\varepsilon_t \int_0^\infty \varepsilon_r^m f(\varepsilon_r; T_r) d\varepsilon_r \\ = & A I_{\varepsilon_t} I_{\varepsilon_r} \ , \end{aligned} \tag{A.5}$$

where,

$$\begin{aligned} I_{\varepsilon_t} &= \int_0^\infty \frac{\varepsilon_t^l}{\Gamma(\frac{\zeta_t}{2}) k_B T_t} \left(\frac{\varepsilon_t}{k_B T_t} \right)^{\frac{\zeta_t}{2}-1} e^{-\frac{\varepsilon_t}{k_B T_t}} d\varepsilon_t \\ &= \frac{(k_B T_t)^l}{\Gamma(\frac{\zeta_t}{2})} \int_0^\infty \left(\frac{\varepsilon_t}{k_B T_t} \right)^{\frac{\zeta_t}{2}+l-1} e^{-\frac{\varepsilon_t}{k_B T_t}} d\frac{\varepsilon_t}{k_B T_t} \\ &= \frac{(k_B T_t)^l}{\Gamma(\frac{\zeta_t}{2})} \int_0^\infty x^{\frac{\zeta_t}{2}+l-1} e^{-x} dx \\ &= \frac{\Gamma(\frac{\zeta_t}{2}+l)}{\Gamma(\frac{\zeta_t}{2})} (k_B T_t)^l \ . \end{aligned} \tag{A.6}$$

Through the variable substitution $x = \frac{\varepsilon_t}{k_B T_t}$, similarly we have,

$$I_{\varepsilon_r} = \frac{\Gamma(\frac{\zeta_r}{2}+m)}{\Gamma(\frac{\zeta_r}{2})} (k_B T_r)^m \ . \tag{A.7}$$

As a result, for $Q(\varepsilon_t, \varepsilon_r) = A\varepsilon_t^l \varepsilon_r^m$, we have,

$$< Q >= A\frac{\Gamma(\frac{\zeta_t}{2}+l)\Gamma(\frac{\zeta_r}{2}+m)}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})}(k_B T_t)^l (k_B T_r)^m \tag{A.8}$$

By observing and comparing the left hand side of Eq. A.1 and the right hand side of Eq. A.8, $\tilde{p}_{rot}$ should have the following form,

$$\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) = \frac{a}{Z_r^\infty}\left(1 + b\frac{k_B T^*}{\varepsilon_t}\right)\left(\frac{\varepsilon_t}{\varepsilon_r}\right)^n \ , \tag{A.9}$$

where a and b are constants to be determined. Substituting Eq. A.9 into Eq. A.1, and performing the integration similarly as that for $< Q >$, we then obtain for $n \neq 1$,

$$a = \frac{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})}{\Gamma(\frac{\zeta_t}{2}+n)\Gamma(\frac{\zeta_r}{2}-n)} \ , \ b = \frac{\zeta_t}{2} + n - 1 \ . \tag{A.10}$$

The final form of $\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r)$ is therefore,

$$\tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) = \frac{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})}{\Gamma(\frac{\zeta_t}{2}+n)\Gamma(\frac{\zeta_r}{2}-n)Z_r^\infty}\left[1 + (\frac{\zeta_t}{2} + n - 1)\frac{k_B T^*}{\varepsilon_t}\right]\left(\frac{\varepsilon_t}{\varepsilon_r}\right)^n \ . \tag{A.11}$$

A.2 The connection factor between rotational inelastic collision probability $p_{rot}(\varepsilon_t, \varepsilon_r)$ and continuum collision number $Z_{rot}(T_t, T_r)$

In order to understand why a connection factor is necessary and determine its value, we conduct an approximate mathematical analysis similar to that used by Haas *et al.* [48], which equates the energy gain simulated by DSMC to that modeled by the Jeans equation.

Suppose a system is in a nonequilibrium state ($T_t \neq T_r$) associated with ζ_t and ζ_r degrees of freedom, and is relaxing toward equilibrium. At time t, assume the average relative translational and rotational energies of the system are $\hat{\varepsilon}_t(t)$ and $\hat{\varepsilon}_r(t)$, respectively. During a simulation time step, Δt, the fraction of particles undergoing collision is $\frac{\Delta t}{\tau_c}$, where τ_c is the mean collision time. For a selected collision pair with

energy ε_t and ε_r, the probability that they will undergo inelastic collision during Δt is: $p_{in} = \frac{\Delta t}{\tau_c} p_{rot}(\varepsilon_t, \varepsilon_r)$. The probability they will not undergo inelastic collision during Δt is: $p_{el} = (1 - \frac{\Delta t}{\tau_c} p_{rot}(\varepsilon_t, \varepsilon_r))$.

If the collision pair undergoes inelastic collision, the post inelastic collision rotational energy ε_r' should be sampled from the distribution function we have discussed in Chapter 4, section 4.2.3.1: $f(\varepsilon_t; T_c) f(\varepsilon_r; T_c) p_{rot}(\varepsilon_t, \varepsilon_r)$, where T_c is the assumed equilibrium state temperature. Using Eq. 4.6 and Eq. 4.10, we have,

$$\frac{1}{Z_{rot}(T_c, T_c)} = \int_0^\infty \int_0^\infty \frac{p_{rot}(\varepsilon_t, \varepsilon_r)}{C} f(\varepsilon_t; T_c) f(\varepsilon_r; T_c) d\varepsilon_t \varepsilon_r \ , \tag{A.12}$$

where we have assumed C to be a function of ε_t and ε_r when p_{rot} is not a constant. From Eq. A.12, we can then assume that the normalized distribution function of $f(\varepsilon_t; T_c) f(\varepsilon_r; T_c) p_{rot}(\varepsilon_t, \varepsilon_r)$ is,

$$F(\varepsilon_t, \varepsilon_r; T_c) = \frac{Z_{rot}(T_c, T_c)}{C} f(\varepsilon_t; T_c) f(\varepsilon_r; T_c) p_{rot}(\varepsilon_t, \varepsilon_r) \ , \tag{A.13}$$

which represents the probability distribution of the post inelastic collision translational and rotational energy when using the general Borgnakke-Larsen (BL) model [19].

In the inelastic collision situation, suppose the post collision rotational energy is ε_r' and the total collision energy is $\varepsilon_c = \varepsilon_t + \varepsilon_r$. Suppose for a fixed total collision energy ε_c, the expectation of ε_r' is $< \varepsilon_r >$, then based on the definition of $F(\varepsilon_t, \varepsilon_r; T_c)$, we should have the following equation hold,

$$\int_0^\infty \int_0^{\varepsilon_c} \varepsilon_r' F(\varepsilon_c - \varepsilon_r', \varepsilon_r'; T_c) d\varepsilon_r' d\varepsilon_c = \int_0^\infty < \varepsilon_r > f(\varepsilon_c; T_c) d\varepsilon_c \ . \tag{A.14}$$

This equation represents the average post inelastic collision rotational energy for all collision pairs (with same or different ε_c), and thus $< \varepsilon_r >$ should have the following form,

$$< \varepsilon_r > = \frac{1}{f(\varepsilon_c; T_c)} \int_0^{\varepsilon_c} \varepsilon_r' F(\varepsilon_c - \varepsilon_r', \varepsilon_r'; T_c) d\varepsilon_r' \ . \tag{A.15}$$

For the situation when p_{rot} is a constant, we have

$$
\begin{aligned}
&< \varepsilon_r > \\
=&\frac{1}{f(\varepsilon_c;T_c)} \int_0^{\varepsilon_c} \varepsilon_r' F(\varepsilon_c - \varepsilon_r', \varepsilon_r'; T_c) d\varepsilon_r' \\
=&\frac{1}{f(\varepsilon_c;T_c)} \int_0^{\varepsilon_c} \frac{\varepsilon_r' e^{-\frac{-\varepsilon_c}{k_B T_c}}}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})(k_B T_c)^2} \left(\frac{\varepsilon_c - \varepsilon_r'}{k_B T_c}\right)^{\frac{\zeta_t}{2}-1} \left(\frac{\varepsilon_r'}{k_B T_c}\right)^{\frac{\zeta_r}{2}-1} \left(\frac{Z_{rot}}{C} p_{rot}\right) d\varepsilon_r' \\
=&\frac{1}{f(\varepsilon_c;T_c)} \int_0^{\varepsilon_c} \frac{\varepsilon_r' e^{-\frac{-\varepsilon_c}{k_B T_c}}}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})(k_B T_c)^2} \left(\frac{\varepsilon_c - \varepsilon_r'}{k_B T_c}\right)^{\frac{\zeta_t}{2}-1} \left(\frac{\varepsilon_r'}{k_B T_c}\right)^{\frac{\zeta_r}{2}-1} d\varepsilon_r' \\
=&\frac{1}{f(\varepsilon_c;T_c)} \frac{e^{-\frac{-\varepsilon_c}{k_B T_c}}}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})} \left(\frac{\varepsilon_c}{k_B T_c}\right)^{\frac{\zeta_t+\zeta_r}{2}} \int_0^{\varepsilon_c} \left(1 - \frac{\varepsilon_r'}{\varepsilon_c}\right)^{\frac{\zeta_t}{2}-1} \left(\frac{\varepsilon_r'}{\varepsilon_c}\right)^{\frac{\zeta_r}{2}} d\frac{\varepsilon_r'}{\varepsilon_c} \\
=&\frac{\Gamma(\frac{\zeta_t+\zeta_r}{2})}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})} \varepsilon_c \int_0^1 (1-x)^{\frac{\zeta_t}{2}-1} x^{\frac{\zeta_r}{2}} dx \\
=&\frac{\Gamma(\frac{\zeta_t+\zeta_r}{2})}{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2})} \frac{\Gamma(\frac{\zeta_t}{2})\Gamma(\frac{\zeta_r}{2}+1)}{\Gamma(\frac{\zeta_t+\zeta_r}{2}+1)} \varepsilon_c \\
=&\frac{\zeta_r}{\zeta_t+\zeta_r} \varepsilon_c \ .
\end{aligned}
\tag{A.16}
$$

Note that the above equation involves the Beta function $B(p,q) = \int_0^1 (1-x)^{p-1} x^{q-1} dx = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)}$. The result $< \varepsilon_r >= \frac{\zeta_r}{\zeta_t+\zeta_r}\varepsilon_c$ is physically understood, since the role of the BL model is to establish equilibrium between the translational and rotational energy modes.

For the situation when $p_{rot} = p_{rot}(\varepsilon_t, \varepsilon_r)$, and is not a constant, we write $< \varepsilon_r >$ in a similar form,

$$
< \varepsilon_r >= \frac{\zeta_r'(\varepsilon_c)}{\zeta_t'(\varepsilon_c) + \zeta_r'(\varepsilon_c)} \varepsilon_c \ , \tag{A.17}
$$

where $\zeta_r'(\varepsilon_c)$ and $\zeta_t'(\varepsilon_c)$ are functions of ε_c. When p_{rot} is a constant, $\zeta_r'(\varepsilon_c)$ and $\zeta_t'(\varepsilon_c)$ will simplify to ζ_r and ζ_t.

If the collision pair does not undergo inelastic collision, then the rotational energy after collision is still ε_r. Combining the two types of collisions, for a collision pair with

energy $(\varepsilon_t, \varepsilon_r)$ at time t, the expected value of rotational energy at time $t + \Delta t$ is then,

$$\begin{aligned}\varepsilon_r(t+\Delta t) &= p_{in} < \varepsilon_r > + p_{el}\varepsilon_r \\ &= \frac{\Delta t}{\tau_c} p_{rot}(\varepsilon_t, \varepsilon_r) \frac{\zeta_r'}{\zeta_t' + \zeta_r'}(\varepsilon_t + \varepsilon_r) + \left(1 - \frac{\Delta t}{\tau_c} p_{rot}(\varepsilon_t, \varepsilon_r)\right) \varepsilon_r\end{aligned} \tag{A.18}$$

The average rotational energy at time $t + \Delta t$ is then,

$$\begin{aligned}&\hat{\varepsilon}_r(t+\Delta t) \\ =&\int_0^\infty \int_0^\infty \varepsilon_r(t+\Delta t) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ =&\int_0^\infty \int_0^\infty \varepsilon_r f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ &+ \frac{\Delta t}{\tau_c} \int_0^\infty \int_0^\infty \left(\frac{\zeta_r'}{\zeta_t' + \zeta_r'}(\varepsilon_t + \varepsilon_r) - \varepsilon_r\right) p_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ =&\hat{\varepsilon}_r(t) + \frac{\Delta t}{\tau_c} \int_0^\infty \int_0^\infty \frac{\zeta_t'}{\zeta_t' + \zeta_r'} \left(\frac{\zeta_r'}{\zeta_t'}\varepsilon_t - \varepsilon_r\right) p_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ =&\hat{\varepsilon}_r(t) + I(\varepsilon_t, \varepsilon_r; T_t, T_r, \zeta_t', \zeta_r')\Delta t \ ,\end{aligned} \tag{A.19}$$

where,

$$\begin{aligned}&I(\varepsilon_t, \varepsilon_r; T_t, T_r, \zeta_t', \zeta_r') \\ =&\frac{1}{\tau_c} \int_0^\infty \int_0^\infty \frac{\zeta_t'}{\zeta_t' + \zeta_r'} \left(\frac{\zeta_r'}{\zeta_t'}\varepsilon_t - \varepsilon_r\right) p_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \ .\end{aligned} \tag{A.20}$$

Finally, with a first order approximation and using Eq. A.19, we have,

$$\frac{d\hat{\varepsilon}_r(t)}{dt} \approx \frac{\hat{\varepsilon}_r(t+\Delta t) - \hat{\varepsilon}_r(t)}{\Delta t} = I(\varepsilon_t, \varepsilon_r; T_t, T_r, \zeta_t', \zeta_r') \tag{A.21}$$

1. Constant rotational collision probability

If p_{rot} is a constant, then $\zeta_t' = \zeta_t$, $\zeta_r' = \zeta_r$, and both are constant. The right hand

side of Eq. A.21 can be simplified as,

$$\begin{aligned} I(\varepsilon_t, \varepsilon_r; T_t, T_r, \zeta_t', \zeta_r') &= \frac{1}{\tau_c} \int_0^\infty \int_0^\infty \frac{\zeta_t}{\zeta_t + \zeta_r} \left(\frac{\zeta_r}{\zeta_t} \varepsilon_t - \varepsilon_r \right) p_{rot} f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ &= \frac{1}{\tau_c} \frac{\zeta_t}{\zeta_t + \zeta_r} \left(\frac{\zeta_r}{\zeta_t} \hat{\varepsilon}_t(t) - \hat{\varepsilon}_r(t) \right) p_{rot} \\ &= \frac{1}{\tau_c} \frac{\zeta_t}{\zeta_t + \zeta_r} \left(\hat{\varepsilon}_r^*(t) - \hat{\varepsilon}_r(t) \right) p_{rot} \ . \end{aligned} \tag{A.22}$$

Comparing the above equation with the right hand side of the Jeans equation (Eq. 4.1), we then have,

$$\frac{\hat{\varepsilon}_r^*(t) - \hat{\varepsilon}_r(t)}{Z_{rot} \tau_c} = \frac{1}{\tau_c} \frac{\zeta_t}{\zeta_t + \zeta_r} \left(\hat{\varepsilon}_r^*(t) - \hat{\varepsilon}_r(t) \right) p_{rot} \ , \tag{A.23}$$

or equivalently,

$$p_{rot} = \frac{\zeta_t + \zeta_r}{\zeta_t} \frac{1}{Z_{rot}} = C_a \tilde{p}_{rot} \ . \tag{A.24}$$

This then gives the analytical connection factor C_a (discussed in Chapter 4) for the constant rotational collision probability model, and is the same as has been discussed by Lumpkin *et al.* [67] and Haas *et al.* [48].

2. Nonequilibrium-Direction-Dependent collision probability

When $p_{rot} = p_{rot}(\varepsilon_t, \varepsilon_r)$, by equating the right hand side of Jeans equation (Eq. 4.1) and that of Eq. A.21, we then have,

$$\begin{aligned} &\frac{\hat{\varepsilon}_r^*(t) - \hat{\varepsilon}_r(t)}{Z_{rot}} \\ &= \int_0^\infty \int_0^\infty \frac{\zeta_t'}{\zeta_t' + \zeta_r'} \left(\frac{\zeta_r'}{\zeta_t'} \varepsilon_t - \varepsilon_r \right) p_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ &= \int_0^\infty \int_0^\infty \left(\frac{\zeta_r'}{\zeta_t' + \zeta_r'} \varepsilon_t - \frac{\zeta_t'}{\zeta_t' + \zeta_r'} \varepsilon_r \right) C_a C_n \tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \\ &= \int_0^\infty \int_0^\infty \left(\frac{\zeta_r'}{\zeta_t'} \varepsilon_t - \varepsilon_r \right) \frac{\zeta_t'/\zeta_t}{(\zeta_t' + \zeta_r')/(\zeta_t + \zeta_r)} C_n \tilde{p}_{rot}(\varepsilon_t, \varepsilon_r) f(\varepsilon_t; T_t) f(\varepsilon_r; T_r) d\varepsilon_t d\varepsilon_r \end{aligned} \tag{A.25}$$

In this case, it is hard to give an analytical expression for C (hence C_n) by equating the two sides of the above equation. However, since in the constant p_{rot} case, the connection factor is shown to be necessary from the above equation, when $p_{rot} = p_{rot}(\varepsilon_t, \varepsilon_r)$, we

anticipate a connection factor is also necessary and should have a different form due to the different expression in the integrand. Generally speaking, C (and also C_n) should be a function of both ε_t and ε_r. For the $(\varepsilon_t, \varepsilon_r)$ dependent DSMC collision probability model for nitrogen, we found that a constant value of C_n reproduces the MD data well, and approximates the Jeans equation in the continuum limit, as discussed in Chapter 4.