Spring 1-1-2012

High-Fidelity Dynamic Modeling of Spacecraft in the Continuum–Rarefied Transition Regime

Craig P. Transky
University of Colorado at Boulder, turansky@colorado.edu

High-Fidelity Dynamic Modeling of Spacecraft in the Continuum–Rarefied Transition Regime

by

Craig. P. Turansky

B.S., University of Colorado, Boulder, 2008

M.S., University of Colorado, Boulder, 2008

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Aerospace Engineering Sciences

2012

This thesis entitled:
High-Fidelity Dynamic Modeling of Spacecraft in the Continuum–Rarefied Transition Regime
written by Craig. P. Turansky
has been approved for the Department of Aerospace Engineering Sciences

Prof. Brian Argrow

Prof. Sedat Biringen

Prof. Jeffrey Forbes

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Turansky, Craig. P. (Ph.D., Aerospace Engineering)

High-Fidelity Dynamic Modeling of Spacecraft in the Continuum–Rarefied Transition Regime

Thesis directed by Prof. Brian Argrow

The state of the art of spacecraft rarefied aerodynamics seldom accounts for detailed rigid-body dynamics. In part because of computational constraints, simpler models based upon the ballistic and drag coefficients are employed. Of particular interest is the continuum-rarefied transition regime of Earth's thermosphere where gas dynamic simulation is difficult yet wherein many spacecraft operate. The feasibility of increasing the fidelity of modeling spacecraft dynamics is explored by coupling rarefied aerodynamics with rigid-body dynamics modeling similar to that traditionally used for aircraft in atmospheric flight. Presented is a framework of analysis and guiding principles which capitalize on the availability of increasing computational methods and resources. Aerodynamic force inputs for modeling spacecraft in two dimensions in a rarefied flow are provided by analytical equations in the free-molecular regime, and the direct simulation Monte Carlo method in the transition regime. The application of the direct simulation Monte Carlo method to this class of problems is examined in detail with a new code specifically designed for engineering-level rarefied aerodynamic analysis. Time-accurate simulations of two distinct geometries in low thermospheric flight and atmospheric entry are performed, demonstrating non-linear dynamics that cannot be predicted using simpler approaches. The results of this straightforward approach to the aero–orbital coupled-field problem highlight the possibilities for future improvements in drag prediction, control system design, and atmospheric science. Furthermore, a number of challenges for future work are identified in the hope of stimulating the development of a new subfield of spacecraft dynamics.

Acknowledgements

I would first extend great gratitude to Prof. Brian Argrow for taking me in as an academic refugee. Furthermore, I would like to acknowledge his years of sage advice, wisdom and patience as well as his humoring of my crazy scheming and tolerance of my nocturnal schedule.

Additional thanks to my RECUV labmates for providing both discussion and entertainment.

Thanks to:

H. Borowski for getting the signatures for this document while I was out of town.

J. Durrie for playing MTG with me in the lab.

M. Stachura for being Canadian.

This work was supported by:

The Neutral Atmosphere Density Interdisciplinary Research Program MURI under Grant FA9550-07-1-0565 sponsored by the Air Force Office of Scientific Research to the University of Colorado at Boulder.

University of Colorado Research Computing and the JANUS supercomputer. Funding for Janus was provided by NSF-MRI Grant CNS-0821794, the University of Colorado and the National Center for Atmospheric Research.

Contents

Tables

Table

Figures

Figure

Introduction

Spacecraft in planetary orbits are subject to gas forces originating from movement through the planet's atmosphere. This presents a problem in modeling spacecraft dynamics that has thus far been treated as mostly ballistic, meaning that aerodynamics is only represented in dynamic models in terms of a few basic inputs, namely the mass, projected area and possibly the surface properties of an object. Since the 1960s, effort has been made to model and estimate the drag on artificial satellites and space debris with the goal of predicting orbital decay and determining atmospheric density. This work presents the satellite drag class of problems in a new way that places a demand on higher fidelity modeling of spacecraft dynamics in orbit that extends beyond drag to include body dynamics and the coupling they bring to the fluid-body system in rarefied flows.

Many human-created satellites orbit at low altitudes; between 200km and 500km. For small objects such as spacecraft in this region, Earth's atmosphere begins to make a transition from a continuum gas to a rarefied one. This transition region is characterized by a number of parameters, all of which surround the elementary measure of rarefaction, the Knudsen number. The Knudsen number Kn is the ratio of the mean free path λ of molecules in a gas environment to some characteristic length L_c:

$$\mathrm{Kn} = \frac{\lambda}{L_c} \tag{1}$$

For Kn less than about 0.01, gas can be considered to be described using continuum models, namely the Navier-Stokes or Euler equations which rely on a differential, analytical method of modeling fluid transport and evolution. As Kn approaches infinity, molecules are assumed to never collide

with each other, a regime which is termed *free-molecular* (FM) flow. In reality, free-molecular flow is often an acceptable assumption to make for Kn $\gtrsim$ 100. The region between these two limits is the continuum-rarefied transition (often referred to as simply "transition") regime where the continuum assumption can no longer be made with confidence. Figure 1 shows the various regimes of fluid dynamics and their associated Knudsen numbers. Many spacecraft operate in

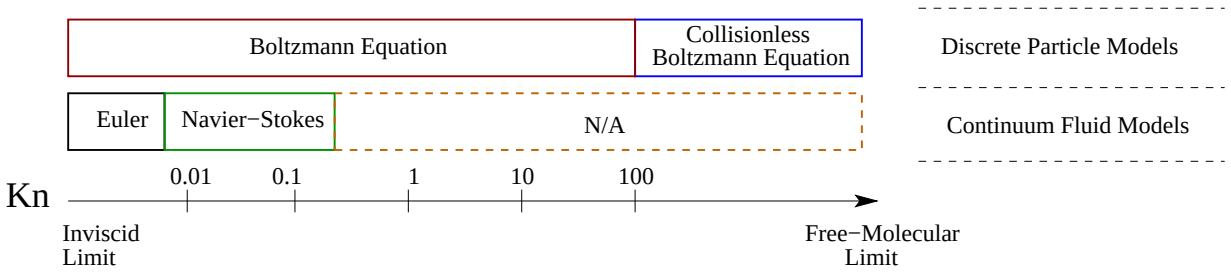


Figure 1: Ranges of Knudsen numbers over which various descriptions are valid.

the transition region of Earth's atmosphere or descend through it during re-entry, thus making understanding of the dynamic effects of the gas on them difficult. This is because tools exist for numerical simulation of both limits of the continuum, but few exist for transition. In the continuum regime, a wealth of traditional Computational Fluid Dynamics (CFD) methods exist, many of which solve an approximation of the Navier-Stokes (NS) equations or some form of general multi- or even single-fluid gas transport equations. In the highly rarefied and free-molecular regimes, the Direct Simulation Monte Carlo (DSMC) method, pioneered by G. A. Bird [3] has become the standard method of choice. There are also approaches for free-molecular flow that are entirely analytical which derive from kinetic theory. Part of the challenge of providing better models for spacecraft in transition flow involves developing new computational tools. The other part is developing methods to use the information these tools provide to refine how spacecraft motion is characterized and analyzed.

A New (Old) Problem in Satellite Dynamics

In the field of atmospheric flight, early models of fluid interactions were simple and focused on understanding lift and drag. For example, linear inviscid potential flow theory was often used to predict fluid forces until the advent of computers and CFD due to its simplicity, and mathematically complete and tractable nature. For early aircraft, the objective was simply to achieve sustained flight. As airplane technology improved, higher-performing aircraft were desired in order to meet the needs of commercial and civil industry, and the military. This requirement drove the development of better fluid models, so as to take the description of airplanes in flight from the realm of only inviscid drag and lift to, eventually, six degree-of-freedom (DOF) dynamic models and associated control theory deriving from more realistic fluid models–all of which were a consequence of the appearance of better mathematical and computational tools and improved understanding of the physics of fluids.

With the arrival of spacecraft, a similar evolution took place. However, the current level of sophistication in spacecraft dynamics is primarily a result of advances in control theory. Rarefied gas effects are introduced as low-order disturbances, often in the form of parameters such as the ballistic coefficient

$$B = \frac{C_D A}{m} \tag{2}$$

where A is some characteristic area, often a projected frontal area, and m is the mass of the spacecraft. The field of satellite aerodynamics has been traditionally focused on the computation of ballistic coefficients and drag coefficients for orbit prediction [14, 32, 17, 16]. Drag modeling, in the form of the drag coefficient C_D has been the focus of much of this field since the 1960s, with little attempt to treat spacecraft dynamics with the same rigor as aircraft dynamics. The reason for this oversight is partly due to the lack of available tools to describe rarefied gas dynamics. Now, tools such as DSMC and other discrete particle methods can provide the information needed to begin examining this problem in greater depth.

Specific Motivations

Although numerous reasons exist to obtain better models of any physical system, there is a set of motivations for satellite dynamics which may be recognized as being paramount. They are:

- Improvements in satellite ballistic drag modeling and subsequent prediction of orbital decay.
- Better atmospheric science due to improved measurement accuracy.
- Wide implications for spacecraft control system design and subsequent vehicle design.

Satellite Ballistic Drag

Most drag and ballistic coefficient models are developed to predict changes in orbital parameters [14] and to predict de-orbit trajectories [6]. Prediction of when and where spacecraft or other objects will impact the surface of the Earth remains difficult but necessary. Few atmospheric entries are predicted to acceptable accuracy in time or space. As examples, the lack of precision in the prediction of the entries of the UARS [68], ROSAT [20], and Phobos-Grunt [36] spacecraft refocused attention on the need for better models of both the upper atmosphere and the dynamics of low-orbital flight and atmospheric entry. The UARS spacecraft was only predicted to enter the "atmospheric interface"[50] with little other information about the remainder of its trajectory. Figure 2 shows the available track as observed by NASA, in which the official description of the craft's final location was *"...over a broad, remote ocean area in the Southern Hemisphere."* [50]. For the Phobos-Grunt spacecraft, predictions were similarly vague. Figure 3 shows a range of predictions of re-entry using Two-Line Element(TLE) data by Kelso[36]. These predictions use up-to-date observations of the actual orbit. By 12 January 2012, the predicted time was 16 January 2012, which was determined later to be the actual time. Yet, predictions continued after 12 January such that when 16 January was reached, current predictions of 18 January were made, when in fact, the spacecraft had already entered. Estimates span a range of roughly three days before

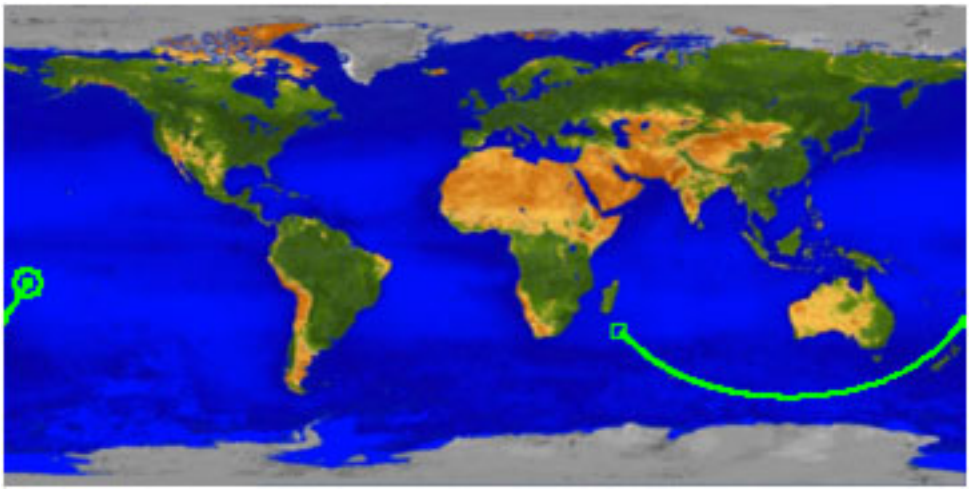


Figure 2: NASA UARS spacecraft final atmospheric re-entry track (image taken from NASA [50]).

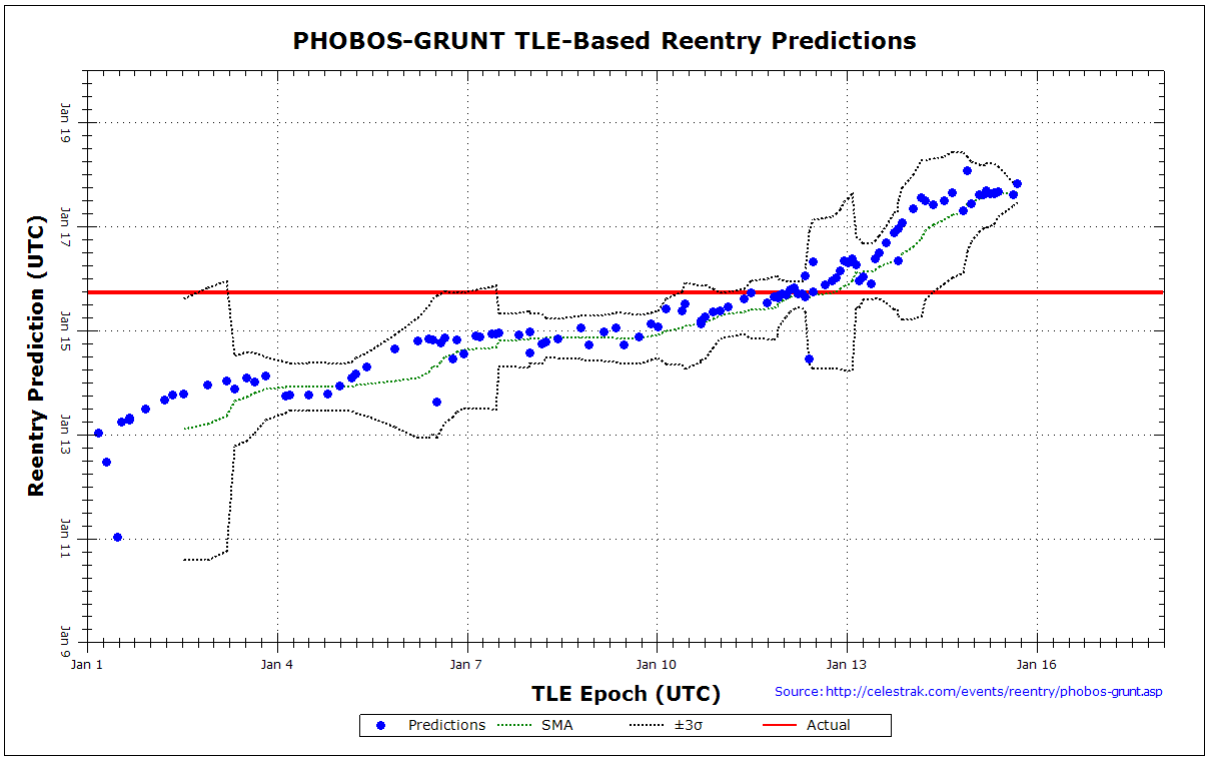


Figure 3: Phobos-Grunt spacecraft final atmospheric re-entry time predictions using TLEs (image by T.S. Kelso [36]).

and after the actual time of entry. The reasons for the lack of higher levels of accuracy in these types of estimations are well-articulated by Stansbery and Johnson [68]:

> *"The two principal and not completely independent influences on reentry prediction accuracy are atmospheric density and vehicle stability.*
> *...The initiation or change of tumbling modes can result in a change in the vehicle's drag profile, which, in turn, can affect the time and location of reentry."*

Since the ballistic and drag coefficients are usually computed from large data sets compiled over numerous orbits, it is obvious that such models cannot appropriately account for short-period attitude changes, and gas-surface interactions associated with rapidly changing altitude, orientation and speed. Furthermore, the drag profile mentioned here directly relates to body-dynamics of spacecraft in the transition flow experienced during re-entry.

Atmospheric Science

Spacecraft have been used as instruments for measuring atmospheric properties, most notably density. Atmospheric models are improved by inferring density from satellite drag measurements [32, 33, 37]. Furthermore, the physics of molecular interactions with surfaces of these spacecraft[57, 49, 15, 9] or general engineering approaches [21, 72] are studied in order to facilitate such measurements. Rigid-body dynamics are seldom included in these analyses. Better models of body dynamics and more accurate numerical techniques (e.g. DSMC) will yield improvements in density models and other necessary information about the upper atmosphere.

Controls and Design

Classical aircraft dynamics utilizes a single-point Taylor series expansion about some equilibrium condition, usually the "trimmed" flight condition. For cruise and mild maneuvers, departures from the equilibrium state are small enough that the equations of motion are effectively linear within a small space surrounding the equilibrium point. Many actively controlled spacecraft engage in maneuvers with large attitude changes that make single-

point expansion linear dynamics useless. Even when trimmed flight can be assumed, there exist few truly sufficient descriptions of spacecraft aerodynamics. Poorly-modeled natural dynamics makes it more difficult to understand and design control systems and prevents those systems from achieving high levels of efficiency. Instead of expending energy to correct or maintain a prescribed attitude which is being perturbed by aerodynamic forces, systems could be designed to cooperate with the natural dynamics rather than conflict with them. Much like atmospheric aircraft, spacecraft in low Earth orbit should be capable of making adjustments to their attitude using control surfaces or other novel methods. This has wide implications for the design of control systems themselves, and of spacecraft hardware, which are currently being overlooked.

Overview

Several decades of development of the direct simulation Monte Carlo method (DSMC) [42, 4, 54, 3] has enabled high-fidelity modeling of objects in rarefied flows, and now with reasonable computational turnaround. Classical methods for aircraft flight dynamics have long been established, particularly in the context of the linearized, rigid-body equations of motion (e.g., [65] and [23]). There appears to be an opportunity to combine DSMC aerodynamics with the methods of linearized flight dynamics for time-dependent and orbit-location-dependent satellite attitude dynamics and orbit prediction. The overall goal of this work may be stated as follows:

> *Descriptions of the motion of spacecraft or orbital debris of arbitrary geometry, mass and other physical properties should include body dynamics resulting from aerodynamic inputs in order to achieve a degree of accuracy similar to what is currently available in atmospheric aircraft flight. The requirements for being capable of creating models for this improvement are the creation of and improvement in rarefied gas dynamics simulation tools such as DSMC and the application of these tools as inputs to rigid-body dynamics in a manner similar to that seen in aircraft flight, but with specific consideration for spacecraft.*

Rather than promoting the use of any specific method for solving any specific problem, the intention of this work is to elevate to prominence the class of problems that arise in spacecraft flight

owing to aerodynamics in the rarefied gas environment. The increase in knowledge of gas kinetics, the availability of data and application of engineering-level problems subject to upper-atmospheric physics, and the appearance and maturity of tools such as DSMC necessitates a revisiting of this subject with a greater degree of scrutiny. Thus, while there is an abundance of information presented, most of it is presented by making qualitative arguments supported by a set of relevant, quantitative demonstrations.

Such demonstrations are in the form of coupling of analytical and computational (DSMC) rarefied gas dynamics with the methods of classical aircraft dynamics. The goal is to develop methods to generalize non-linear spacecraft aerodynamics which are not restricted to a single-point expansion, nor that treats aerodynamic forces as unstructured disturbances. This starts with a set of spacecraft parameters for arbitrary geometry and mass properties, with flight conditions based on orbital altitude and speed used as inputs to a simulation. As a demonstrative example, this family of methods is first applied to the classical NACA-0012 airfoil in two-dimensional (2D) free-molecular flow with a simple single degree-of-freedom dynamic model coupled with well-known analytical relations that describe molecular fluxes and forces for simple geometries. This is followed with dynamic simulations of the same airfoil geometry with a DSMC flow solver coupled with the same dynamic model. Additionally, one set of general conditions (per regime) that are common in low Earth orbit is imposed to limit the analysis to a computationally tractable parameter space.

Proposed is the creation of a set of tools that are centered around a new DSMC code (Section 1.2) and method for describing rigid-body dynamics of satellites in orbit, with emphasis on operation in the transition regime. The breakdown of this approach is shown schematically in Figure 4. The topmost goal is the aforementioned "new" problem. A six-DOF method may require considerable time, and thus, it is sufficient to provide a framework and set of guiding principles and examples in lower dimensions. The Tools half of the tree in Figure 4 relies on the improvement of DSMC as a simulation tool. In truth, DSMC is mature enough to be capable of providing much of the information required to complete the final goal. However, there are many smaller obstacles to its usage. One of these is the availability of suitable software implementations (codes) in the

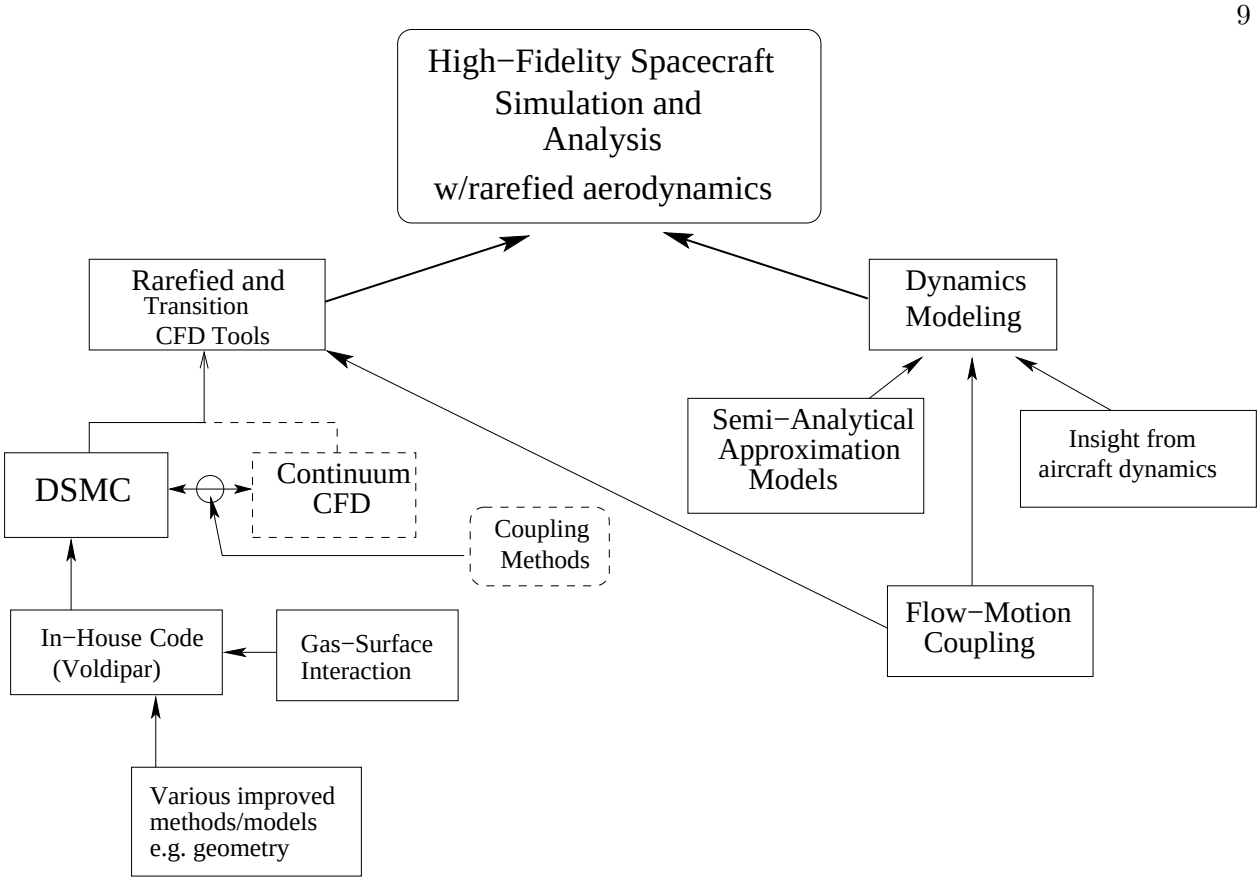


Figure 4: Schematic diagram of the hierarchy and relation of the elemental units of this research. Dashed boxes are optional but possibly very useful elements (DSMC - Direct Simulation Monte Carlo).

community. Codes by G.A. Bird are often considered suitable for benchmarking purposes but have limitations in inputs (namely, lack of certain gas-surface interactions) and usability. It was deemed worthwhile to develop a new code to suit the needs of research-level development. Thus, with complete control of the creation and development of such a new code, optimal simulation freedom is achieved. This code, currently in usable state for 2D problems is discussed in detail in Chapter 1. The Dynamics half consists of looking for ways to describe a spacecraft's rigid-body dynamics such that either analytical insight or simulation fidelity, or both can be achieved. Preliminary investigations begin with using the analytical free-molecular models in Chapter 3 to understand how to use inputs from a simulation like DSMC to create models of dynamics. This effort then extends to

discuss approximation methods like those introduced in Chapter 4 which seek to provide a single set of data that can be said to describe the motion of an arbitrary rigid body in rarefied and transition flow in a closed-form manner.

In Chapter 5 one of the most difficult problems in flow-motion coupling is addressed in the context of rarefied gas dynamics. The problem is how to accurately model the feedback system which arises in a rarefied gas system when an object is moving within it. This is of interest in that rigid-body dynamics naturally includes treatment of rates of linear motion and of rotation which create a challenge for modeling these systems. This class of problems falls under the general category of fluid-structure interaction or coupled-field problems. In order to proceed with simulations of rarefied aerodynamics of rigid bodies, the effect of rotation rates of a single DOF system are examined using numerical studies.

Chapter 6 demonstrates the application of methods from all previous chapters to present case studies of two objects in transition flow. The objects are simulated in the natural 2D orbital frame and incorporate body dynamics by inclusion of the single DOF given by the angle of attack. DSMC is used to generate high-resolution data sets which represent aerodynamic forces and moment acting on the objects as they orbit within the lower thermosphere. Time-accurate simulations of the objects' motion in orbit as they descend and enter Earth's atmosphere are given. Detailed dynamic characteristics of these systems are examined, thereby gaining insight into transition flow dynamics that has not been available before. Such simulations set a precedent for further work in rarefied aerodynamics of spacecraft. Finally, extensions to three-dimensional (3D) systems and other necessary directions for future work are discussed.

Chapter 1

Review of DSMC and Introduction to a New Code

The Direct Simulation Monte Carlo (DSMC) method is a numerical method originally developed by Bird [3] for simulating rarefied fluid flow. Its physically realistic but phenomenological approach, as an alternative to direct discretization of the Boltzmann equation, has contributed to its wide use in solving problems involving spacecraft in low-to-high earth orbit. It has also been used to develop new physical models for gas–surface interactions and intermolecular collisions as well as discrete particle computational techniques. Section 1.1 gives an introduction to DSMC in the context of motivations and top-level implementation details.

Bird's collection of DSMC codes has served as a standard for some time. His original demonstration codes [3] still serve as reasonable benchmarking tools, as do his "production" codes DS2V and DS3V. A number of codes have been developed to rival Bird's originals, each offering its own improvements upon the method. Some of these include MONACO by Dietrich and Boyd [19], DAC by NASA [42] and dsmcFOAM (part of the openFOAM open source CFD suite of software) by Scanlon et al. [62]. Many other codes have been developed but have less presence in the literature either due to propriety or being similar to and derived from the available codes. Each also presents its own set of challenges when using it for research. Some provide an interface that may not suit one's particular needs. Others may not contain sufficient capability to model certain physics such as gas-surface interactions, internal molecular energy modes or complex geometries. In order to have complete control and understanding of a DSMC code, one was written using Bird's test codes as an example and then extending their core concepts to create a more full-featured product. The

code, named Voldipar, was designed with the easy injection of new surface reflection models in mind. Its other specific features and methods will be discussed in Section 1.2.

1.1 A Review of the Motivations and Basic Principles of DSMC

DSMC is a method involving the phenomenological simulation of a finite set of discrete particles, each representing a large number F_N of physical particles in a system that evolves stochastically. The key to describing the motivation behind the development and usage of DSMC is in noting that it is "direct simulation". This means that DSMC does not attempt to approximate the Boltzmann equation (Equation 1.1) but rather provide an acceptable modeling of it through the imposition of a system of discrete particles that evolve according to first principles.

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c} \cdot \frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F} \cdot \frac{\partial(nf)}{\partial \boldsymbol{c}} = \Omega(f, f^*, f_1, f_1^*, n, c_r) \tag{1.1}$$

$$\Omega(f, f^*, f_1, f_1^*, n, c_r) = \int_{-\infty}^{\infty} \int_{0}^{4\pi} n^2 (f^* f_1^* - f f_1) c_r \sigma d\alpha d\boldsymbol{c}_1 \tag{1.2}$$

The Boltzmann equation is a partial differential equation in the variables n (number density) and f (velocity distribution function) with two three-dimensional independent variables $\boldsymbol{c}$ (velocity) and $\boldsymbol{r}$ (position). The large size of the independent variable space makes any discretization of the equation in 3D rather intractable. However, what presents more of an obstacle is the collision integral Ω, for which few approximations can be easily made for general cases. The collision integral is the driving element of the Boltzmann equation. It models the process by which molecular collisions of molecules in different velocity classes change the particle velocity distribution f; where f is the distribution of particles in velocity class $\mathbf{c}$ prior to collision and f_1 is the distribution of particles in velocity class $\mathbf{c}_1$ prior to collision. The post-collisional distributions f^* and f_1^* are the values of the distribution function for particles in post-collisional velocity classes $\mathbf{c}^*$ and $\mathbf{c}_1^*$, respectively. The variable c_r is the collisional relative speed and σ is the collision cross-section. It relatively unclear how to model the collision integral or *operator* for the reasons that it contains the mechanism for utilizing inverse collisions–something that is purely mathematical–and that the elements of it (e.g.

the distributions) are difficult to define for general cases. [1] The presence of an operator, the elements of which are effectively unknown, prevents any traditional discretization method from being implemented in any clear way. Approximations for the integral under specific assumptions may be derived, however, given a molecular collision potential function [66].

Thus, using the Boltzmann equation to simulate a discrete particle model of fluids proves difficult. Nevertheless, the equation itself remains a complicated way to represent what is essentially a simple process. Discrete particles still obey the laws of nature, and most of the modeling of individual particles interacting with each other and with their environment can be very accurately described using classical mechanics. It is this principle that defines the field of molecular dynamics. However, in molecular dynamics, relatively few particles are used, and often in the context of studying the physical details of their interactions (e.g. modeling chemical reactions) rather than in that of an engineering simulation of a large system (e.g. flow over a body in a fluid or *aerodynamics*). Since the latter context is what had been lacking prior to the development of DSMC, Bird's work sought to create a practical solution to solve engineering problems that would otherwise require the solution of the Boltzmann equation. The result of his work became a method, DSMC, which utilizes some clever means of averaging, application of a Chapman-Enskog expansion and a "small" sample of simulated particles. A direct simulation of this sort can be said to have three key requirements:

(1) A way to decouple collisions from advective motion

(2) A way to determine how many collisions should occur and when

(3) Ways to model

- Binary collisions
- Molecule translation
- Internal modes (e.g. vibrational, electrical, quantum)
- Surface interactions

[1] The collision integral is often deemed an operator as it acts on the distribution functions to change them. In Lattice-Boltzmann Methods (LBM) [69, 13, 41], various types of collision operators are chosen and studied, for example.

1.1.1 Decoupling Collisions and Advection

The first assumption that must be made to decouple the processes of collision and advection is that all collisions are binary; the dilute gas assumption. The next is that there is often a disparity in time scales between the two processes. If only two molecules are allowed to collide at a time, and the time interval over which this collision occurs is typically an order of magnitude smaller than the time required for any molecule in the system to traverse the distance to its next collision, then the Chapman-Enskog expansion [12] can be used to formally describe the separation of time scales as

$$f = f^{(0)} + \epsilon f^{(1)} + \epsilon^2 f^{(2)} + \mathcal{O}(\epsilon^3) \tag{1.3}$$

$$\frac{\partial}{\partial t} = \epsilon \frac{\partial}{\partial t_1} + \epsilon^2 \frac{\partial}{\partial t_2} + \mathcal{O}(\epsilon^3) \tag{1.4}$$

Equation 1.3 represents an expansion of the distribution function in terms of perturbations of the Maxwellian equilibrium $f^{(0)}$ and $\epsilon \approx \mathrm{Kn}$. Equation 1.4 is an expansion of a time derivative (suitable for transforming the Boltzmann equation) which separates it into a collision time, t_2 and advective time t_1. These equations are not explicitly required for the implementation of any DSMC algorithm. They merely show that a mathematical description of the desired decoupling exists. In LBM, however, they do explicitly appear and are used to manipulate the derivations of that method in order to prove its relation to the Euler and Navier-Stokes equations. The result of this apparent ability to decouple movement is that an algorithm consisting of two to three distinct parts emerges. They are

(1) Movement

(2) Collision

(3) Entry (possibly, may be incorporated)

Each of these steps occurs at each iteration of a DSMC simulation's main time loop. The reason the entry step is not necessarily distinct is that many authors choose to incorporate it into the movement step, thus somewhat obscuring its implementation. The calculation of a proper scheme

that models entering particles in a particular simulation can be nontrivial, however. Given the physical importance of performing entry correctly, it is fully enumerated here as its own distinct step. These three steps appear in all DSMC codes as well as in LBM codes, as LBM is a natural and conceptual analog of DSMC. In addition to the core three steps, a sampling step is often performed to calculate and output macroscopic field and surface values. This process is not physically imperative to the execution of the DSMC evolution itself, so it isn't considered to be a core algorithm element. It is true, however, that many gas physical models, such as energy redistribution or surface reflection models, may rely on macroscopic variables in real time (e.g. temperature, density). Thus sampling, while not part of the basic DSMC evolution, is almost always performed (if only to extract meaningful output data, an obvious desired result of the simulation). These core steps are shown in the source of Bird's `DSMC2` test code and in a debug build of Voldipar, for comparison in Figure 1.1.

1.1.2 Collision Mechanics and Details

Following from the identification of the DSMC core algorithm, and noting that the movement step is largely trivial (physically, but often not practically), the necessary information regarding collisions must be known in order to perform the collision step. This information includes: how many collisions occur and where, between what molecules and what is the mechanism of colliding. There are a number of approaches to answering these questions. To determine where, and subsequently, how many collisions occur, the domain of interest must be divided into finite volumes or cells. The reason for this is actually two-fold: discretizing the domain into smaller subsections (cells) is necessary to provide output results in the fluid field and discrete volumes are required to isolate potential collision partners from others that may be in a completely different location in the domain, and thus highly unlikely to collide. Although much effort has been expended by Bird [4, 5, 2] and others [40, 60, 70] in studying collision models and improving practical collision procedures, a useful and relevant model for performing collisions begins with Bird's No-Time-Counter (NTC)

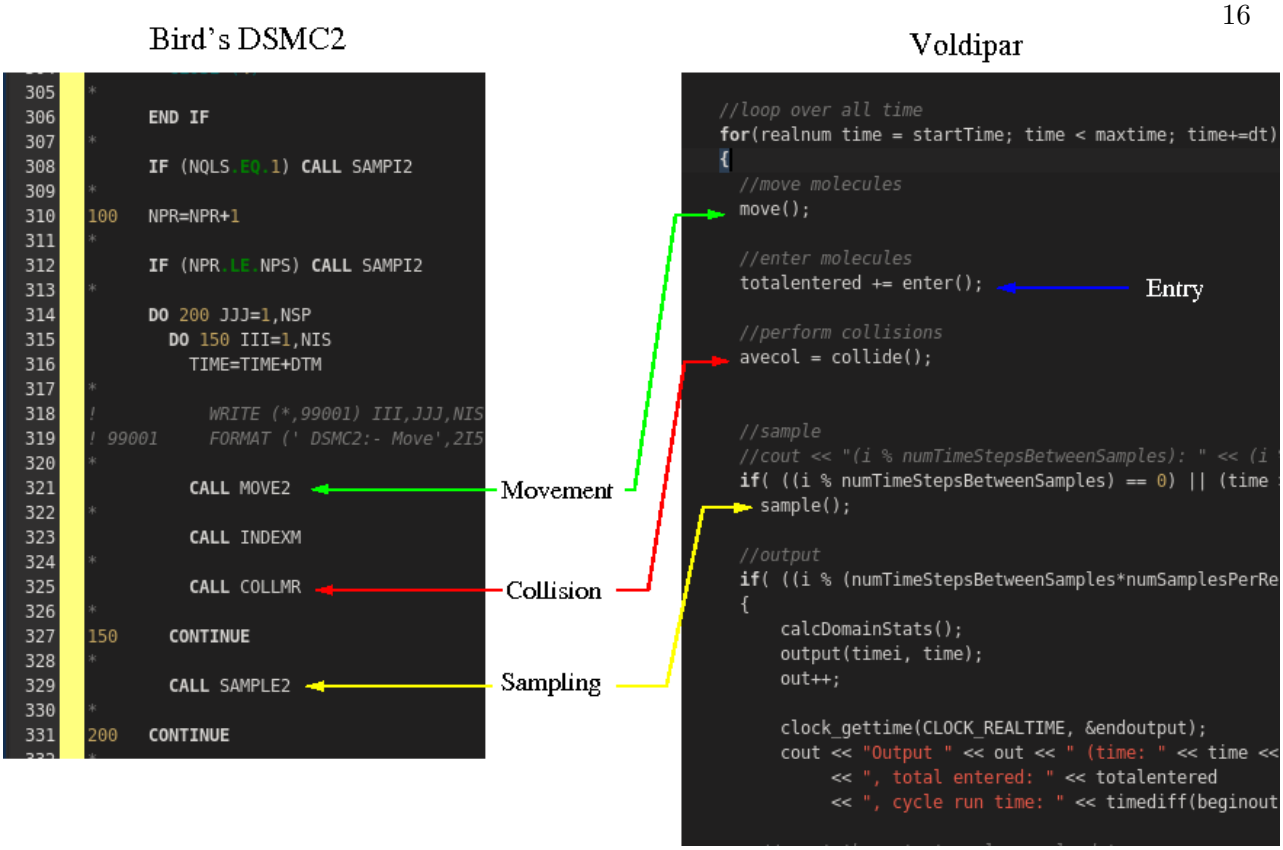


Figure 1.1: A comparison of two DSMC codes' implementations of the core elements of DSMC as they are called from within time loops. Note that Bird's `DSMC2` test code does not explicitly include the entry step in the main loop.

procedure for determining how many collisions occur in a given cell.

$$N_{pq} = \frac{1}{2V_c} N_p \bar{N}_q F_N [(\sigma_T c_r)_{\max}]_{pq} \Delta t \tag{1.5}$$

Equation 1.5 (which is Equation 11.5 in Bird's text [3]) gives the maximum number of collisions likely to occur in a cell of constant volume V_c over the constant time interval Δt between two particle species p and q where N_p is the number of particles of species p in the cell, $\bar{N}_q$ is the mean number of particles of species q in the cell, $[(\sigma_T c_r)_{\max}]_{pq}$ is the maximum value of the collision cross–section product in the cell thus far in the simulation and F_N is the ratio of real molecules to simulated molecules. Using this equation, the number of possible collisions is determined, and then collisions are attempted via one of any number of selection methods. Thus, the NTC method

can be summarized with the following procedure:

(1) For each combination of p and q, calculate N_{pq} using Equation 1.5.

(2) Select a molecule of species p at random from the cell.

(3) Select, via some appropriate method, a "close" collision partner of species q.

(4) Calculate $\sigma_T c_r$ for the *possible* collision $p \leftrightarrow q$, with c_r being the relative speed between the two possible partners.

(5) Compare: $\frac{\sigma_T c_r}{[(\sigma_T c_r)_{\max}]_{pq}} > R_f,\ R_f \in [0..1)$? If so, the collision occurs.

(6) Perform the collision by choosing at least post-collision velocities, and likely post-collision energies.

Step 3, collision partner selection, is one of the most discussed topics amongst practitioners of DSMC. The most obvious way to select a collision partner is to simply select a random particle in the same cell of species q. Since a cell should be considered the smallest unit of domain space in the simulation, this method can be acceptable. However, it has been noticed [48] that choosing in this manor can cause the destruction of intracell gradients that would otherwise have been resolved had closer collision partners been selected. Much of the difficulty in designing a collision selection procedure involves balancing the inherent need in DSMC to minimize the mean collision distance between colliding particles and maximizing performance. This leads to the often–referenced Nearest Neighbor method, which is simply selecting the closest (via euclidean distance) particle of the appropriate species. In order to determine which particle is the exact closest, however, all particles of species q in the cell must be searched, and their distances from the initial particle p must be be calculated, an $\mathcal{O}(N_q^2)$ procedure. This often proves much too computationally expensive for practical implementation; thus linear or better selection methods are preferred.

The method that strikes an appropriate balance between Nearest Neighbor and cell-random selection is subcell selection. The cell is divided into a number of discrete subdivisions or subcells

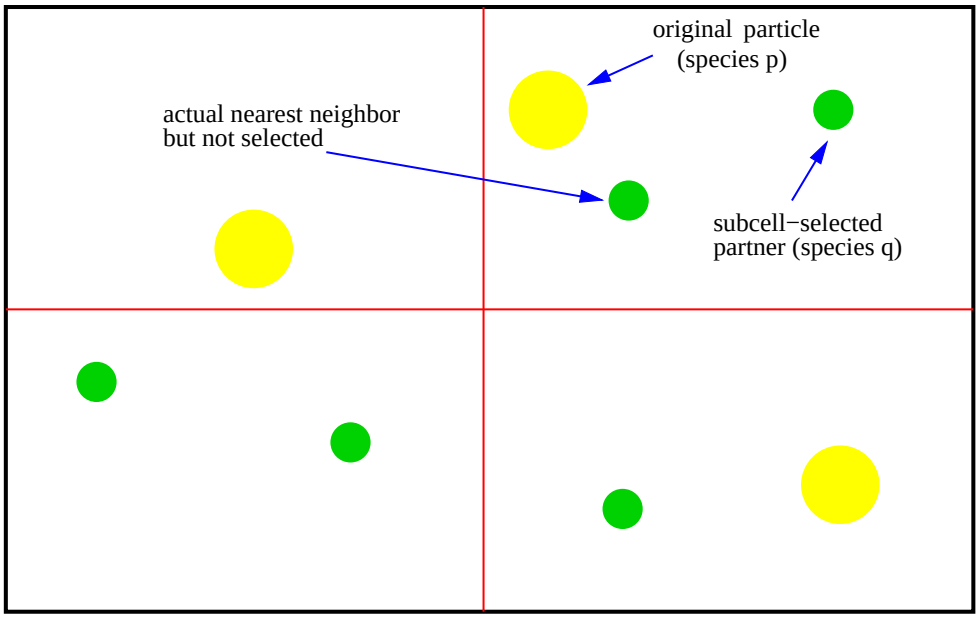


Figure 1.2: A DSMC collision cell of volume V_c with multiple species present and divided into subcells for selection.

(the geometry of which remains yet again another matter of debate) as shown in Figure 1.2. The initial particle's current subcell is checked first to see if any particles of species q exist within it. If so, one of those particles is selected at random. If not, some outward search algorithm selects the "next" subcell (e.g. using an outward spiral search) and that subcell is checked for at least one particle of species q being present, and so on until a suitable partner is found. Also, if the collision is intended to be a $p \leftrightarrow p$ or $q \leftrightarrow q$ collision, care is taken to avoid colliding the initial particle with itself. This selection method is linear in N_q but may not always select the exact closest particle (Figure 1.2 shows this event occurring, for example). As the subcells represent an even finer division of domain space, a good choice of subcell size and geometry serves to preserve the intracell qualities such as gradients (or even vorticity) and keep the mean collision separation distance adequately low for physically realistic collision modeling. Although there have been various attempts at modifying this procedure such as using pseudo-subcells[45], the underlying essence is similar to the method described in general here.

Finally, once the collision partners have been selected, the collision is calculated. This entails, at the very least, generation of post-collisional velocities for each particle. Additional modeling of internal modes may occur, such as rotational and vibrational energy re-distribution or treatment of electrical and quantum modes. Chemical reactions can also occur, possibly creating new species or removing currently active species. All of these considerations fall under the category of modeling binary collisions, the third overall requirement of DSMC. The most important aspect of a collision to remember, however, is that the quantity of each particle that never changes due to colliding is its position. The decoupling of collisions and advective motion prevent this from occurring, thus enforcing a collision to occur instantaneously. And despite the definition of a collision time scale (which is necessary to compute N_{pq}), molecule movement still remains a separate core step of the DSMC algorithm [2] .

1.2 The Voldipar DSMC Code

Voldipar is a new DSMC code has been created that will provide the means of performing simulations to the exact specifications required to investigate new or different physical models of gas-surface interactions, internal modes and other details of gas dynamics, particularly as they pertain to engineering problems. The development of DSMC as a method itself will also be examined. This effort includes implementation of new geometry models and performance alterations stemming from the algorithms that drive the smaller elements of DSMC.

Much of the code remains to be written. As of this time, only 2D geometry modeling is available. The smaller models of DSMC, such as binary collisions, surface interactions, chemical reactions and additional internal mode modeling are limited to a set of basics for each. The Hard Sphere (HS), Variable Hard Sphere (VHS) and Variable Soft Sphere (VSS) binary collision models are included. Rotational energy re-distribution according to the Larsen-Borgnakke model

[2] Variable-adaptive time steps allow collisions to consume the simulation time in an appropriate way, thus moving back to a kind of adapted Time-Counter (TC) method. However, the times of each cell and particle are kept as separately evolving quantities, independent from each other and from a baseline or reference time maintained by the simulation. So collision and motion are still decoupled.

is implemented. No chemical reactions are possible. Surface interactions are limited to specular and standard Maxwellian diffuse, however plans to include more surface models are of priority (see Chapter 2). Finally, one advanced feature (introduced briefly by Bird in Ref. [5]) is the optional use of variable-adaptive time steps, which can increase computation performance by an order of magnitude.

1.2.1 Geometry Model

Similar to how Bird's `DS2V` and `DS3V` codes contain their own geometry models and procedures for generating necessary geometric information from user inputs, Voldipar contains its own geometry model that takes simple coordinate data as an input in 2D and will take standard triangle surface data as input in 3D. From these data, a volumetric representation of internal, closed boundaries is constructed by means of voxelization. A voxel is the generalized analog of a pixel–the smallest geometric element of a 2D image (In 2D, this is termed *rasterization*). Usage of pixel or voxel discretizations of space stems from the field of computer graphics and imaging where approximating a real surface or volume is necessary in scan-conversion [35], such as in medical imaging and in collision detection scenarios very often present in games [29]. The latter case clearly relates to DSMC as particles are required to collide with body geometry. A conservative voxelization [28, 75], which is a discretization of an object in volumetric space such as to allow a topologically closed subset of the space (as voxels), is ideal for implementing a collision detection system via discrete ray-tracing [74]. Volumetric geometry has the advantages of being conceptually simple to implement and fast when collision detection is being performed.

It is worth mentioning that Bird's geometry and collision detection models are somewhat volume-based but are complicated to implement. They require setting of levels or regions extending from a surface in order to facilitate fast collision detection. The extension of Bird's methods from 2D to 3D is also significantly more arduous to implement. In contrast, standard voxel methods and their associated discrete ray-tracing techniques are well-covered in available literature and can scale well from 2D to 3D, both in terms of performance and implementation complexity. The

disadvantage is primarily that non-uniform voxel sizes are generally not permitted as much of the practical performance benefits that arise from volumetric techniques degrades when uniformity is not enforced [3] .

Collisions are determined by performing a discrete ray-trace. At the beginning of its move step, a DSMC particle has a velocity and position. The velocity vector indicates the direction in which the particle will move until it is stopped by a collision with a solid surface or its allowable movement time has elapsed. With the entire domain discretized into voxels, the ray-trace is accomplished by voxelizing the velocity vector or ray one voxel at a time, successively along the direction of the vector, using the conservative algorithm found in Ref. [22]. Each time a new voxel of the ray is created, it is examined and checked for its type (the type of each voxel is set prior to the movement step occurring and is assumed constant over the entire movement step for all particles). If the voxel type is a not "empty" (i.e. not a surface or solid voxel), then the particle is stopped and a collision with the associated surface of the current non-empty voxel is recorded and the appropriate surface interaction procedures are performed. If the voxelization of the ray never encounters a non-empty voxel over the entire prescribed movement time, the particle has not collided with any surface. This procedure is illustrated in Figure 1.3. It is important to notice why the performance of a discrete ray-trace operation is superior to classical ray-tracing (particularly in the context of DSMC), which is often considered an extremely costly operation. Contrary to classical ray-tracing, which must examine all objects in the domain and check for an intersection, a discrete trace is ignorant of the contents of the space. The ray propagates one voxel at a time until it collides with a surface or reaches its maximum length. In DSMC, as in other first-collision-only schemes (e.g. physical movement/projectile mechanics in games), where a collision represents a physical impact of a particle with a solid object, each particle only needs to "look" immediately ahead of its current path as it moves (this type of trace can be termed a ray-*cast* since it is assumed that the ray's propagation can be interrupted, and once so, cannot collide with any other object). Additionally, a

[3] Voxel-based methods typically assume all voxels are of the same small volume. This is one of the key characteristics of "atomic" methods, which construct all larger and more complicated objects from the same set of uniform pieces (voxels).

level system like Bird's can be implemented on top of the base ray-casting procedure to accelerate it further, if desired.

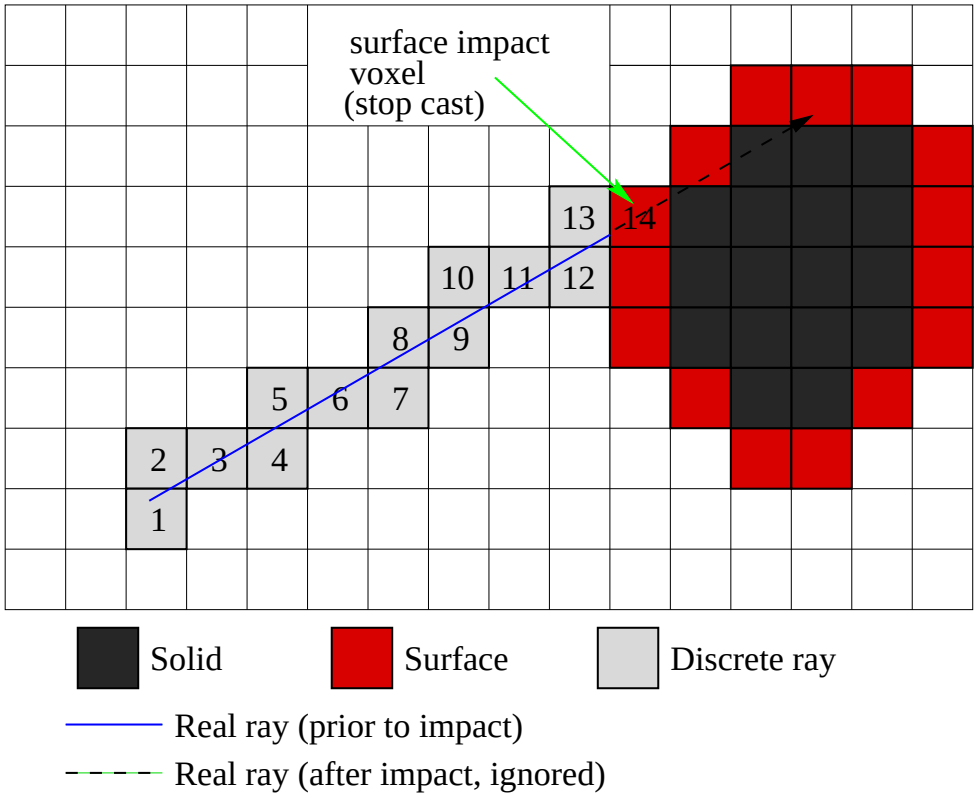


Figure 1.3: Visual illustration of a discrete ray cast in a 2D voxelized domain. The "real" ray is defined by the particle's velocity vector and prescribed movement time. The discrete analog of the ray (in this case, an 8-connected line [30, 22]) is shown as a sequence of voxels that are checked as the ray propagates (shown with sequential integers).

A voxelized domain geometry model forms the core of Voldipar and with it, allows virtually any geometry inputs to be modeled by the code. The remaining DSMC core elements such as binary collision modeling, particle entry and sampling are relatively unaffected by this choice. Surface sampling represents the only problem with a voxelized domain, as it is somewhat unclear how to associate surface voxels of internal boundaries with their original data (e.g. line segments in 2D or triangles in 3D). There are a number of approaches to addressing the problem which is described in Section 1.2.3.

1.2.2 Cell Volume Calculation

DSMC's main procedures are generally unaffected by geometry models. Some changes are important to describe, however. Although particle tracking and surface impacts are the most obviously affected sections of DSMC, collision procedures have some dependence as well. This dependence stems from the requirement in virtually any collision model to know the finite volume within which collisions are to occur (e.g. V_c in Equation 1.5). In a uniform Cartesian cell configuration, a voxelized domain makes determining the available or empty volume in each cell rather straightforward if the assumption can be made that all voxel boundaries lie exactly coincident with cell boundaries. A robust code will unfortunately not be able to enforce this requirement as it is beneficial to provide the user the ability to specify cell resolutions and voxel resolutions separately. Thus, the general case is that not all choices for domain cell resolution may allow for the coincidence requirement to be met. Figure 1.4 shows this general case where a single arbitrarily-placed

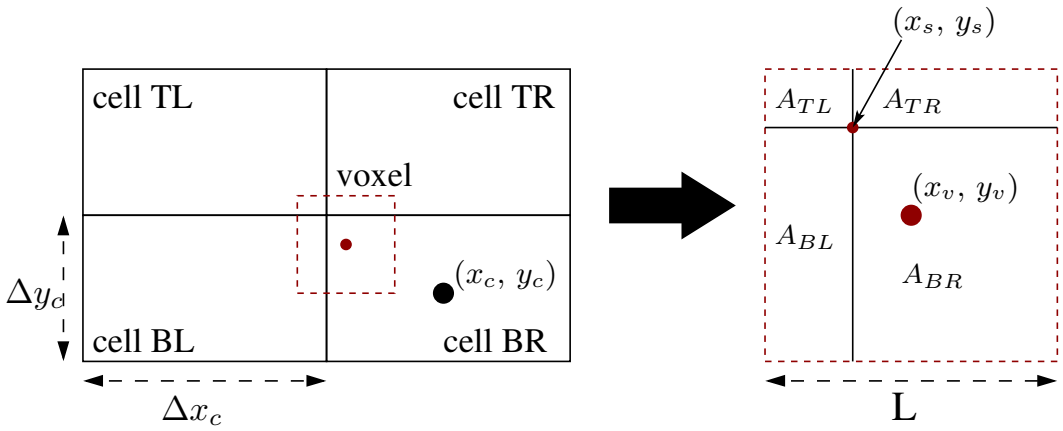


Figure 1.4: Cell volumes in 2D as divided by an arbitrary non-empty voxel that spans a set of four adjacent cells. The voxel's primary cell, BR, is a quadrant four cell in this example.

voxel may span up to four adjacent cells, labeled according to their positions (BL: bottom left, BR: bottom right, TL: top left, TR: top right). The voxel is assumed to be non-empty, and thus should remove from each of the cells a piece of available collision volume. Each cell begins with an available empty volume of $V_c = \Delta x_c \Delta y_c$. Each voxel is exactly square, with side length L. When

a non-empty voxel is created, the volume it consumes must be distributed properly amongst the four cells it can possibly span. To do this, the voxel's primary cell must be identified. The primary cell is the cell in which the voxel's center point (x_v, y_v) resides. The zero-indexed indices of the primary cell are found with simple and fast fixed-point rounding operations such as

$$\begin{aligned} i_c &= \text{fix}\left(\frac{x_v}{\Delta x_c}\right) \\ j_c &= \text{fix}\left(\frac{y_v}{\Delta y_c}\right) \end{aligned} \tag{1.6}$$

Next, the primary cell's relation to the other three must be determined. At this point, the only information available is what the voxel's primary cell is. And since all four cells may require a volume adjustment from this voxel, the indices of the remaining three must be known. The displacement vector between the primary cell's center, (x_c, y_c), and the voxel center is found as

$$\begin{aligned} d_{cx} &= x_c - x_v \\ d_{cy} &= y_c - y_v \end{aligned} \tag{1.7}$$

A quadrant test is performed on the displacement vector. In the example of Figure 1.4, the primary cell is a quadrant four cell. The cell intersection point (x_s, y_s) is then used to determine the volumes (areas in 2D) of each of the sub-regions of the voxel. In the example, $A_{BL} = (L/2 - \Delta x_s)(L/2 - \Delta y_s)$ where, for this particular case of a quadrant four primary cell: $\Delta x_s = x_v - x_s$, $\Delta y_s = y_v - y_s$. For different quadrant primary cells, these definitions require specific sign inversions.

Each sub-area is then subtracted from its respective cell's current area and the process is repeated for the next non-empty voxel that is created. Eventually, cells that intersect internal boundary surface geometry have proportionally less available volume with which to calculate collisions and cells contained entirely within a closed boundary (i.e. completely "solid" cells–those that are completely covered by non-empty voxels) have an available volume of zero. The result of volume-adjustment is shown for an example airfoil body in Figure 1.5. Upon performing collisions, a cell is first checked to see if its available volume is zero and if so, no collisions are considered for that cell. This volume distribution procedure is performed during domain voxelization. Thus,

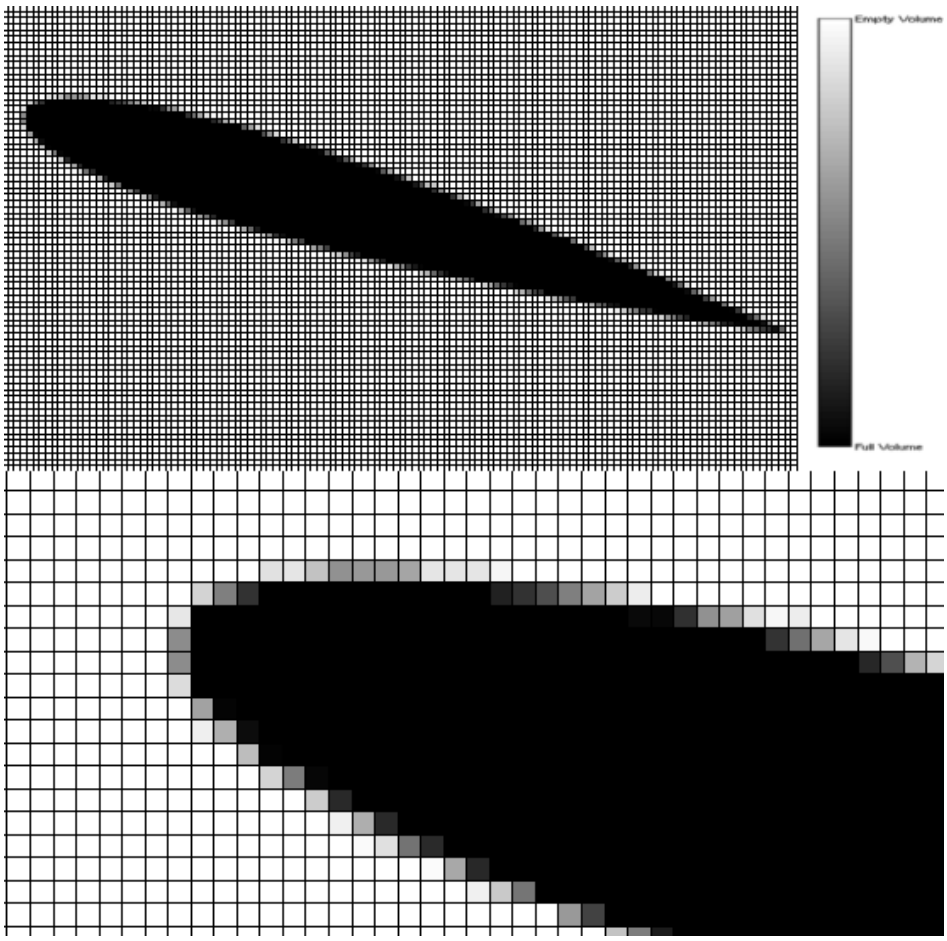


Figure 1.5: Cell volumes after adjustment from non-empty voxel occupation for an airfoil. Darker regions indicate cells with more filled volume. Top: full body. Bottom: leading edge zoom.

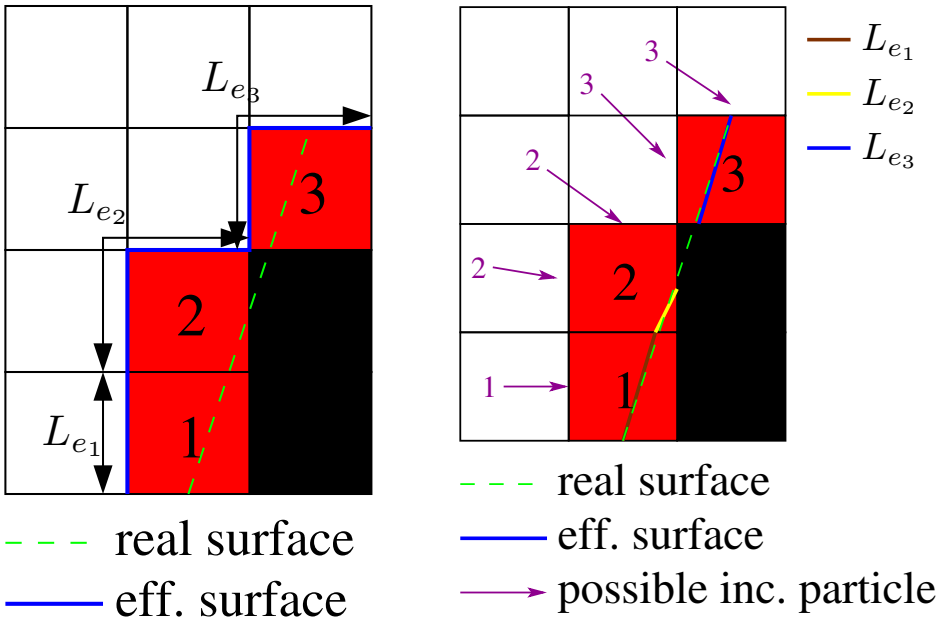
any time the domain is re-voxelized (something that should only occur for moving boundaries), cell volumes must be updated this way as well.

1.2.3 Voxel–Boundary Sampling

A voxelized representation of real source geometry presents a problem when surfaces are considered for sampling. Since surface impacts are determined by a discrete ray propagation into (and actually on top of) a surface voxel, the number of particles that have impacted a given surface can only be tracked per-voxel of that surface. Thus, converting a surface that originally is defined as a sequence of 2D line segments between points or 3D triangles to a set of connected voxels means

the original length or area of the surface elements (be they line segments or triangles) is lost, and thus unavailable when calculating surface properties.

One initial approach to solving this problem is to enforce surface sampling per-voxel. Each surface then has as many sampling locations as their are surface voxels that comprise it. Therefore, as voxel resolution increases, sample resolution does so as well. As for the matter of what surface area to use, a few methods can be applied. The first, as shown in Figure 1.6a, takes each surface



(a) Method 1: Effective areas L_e of each voxel in the surface.

(b) Method 2: Effective areas L_e of each voxel in the surface.

Figure 1.6: Two methods for determining surface sampling area per voxel in 2D.

voxel's exposed sides and sums them to form a total effective area L_e. In this example, the total effective area of the surface–the sum of all voxels' effective areas–is much greater than the actual area. For any non-co-linear boundaries (i.e. general boundaries) such as the curved surface of the cylinder in Section 1.2.6.2, this method is ineffective as it very rarely calculates accurate sampling areas. The second method, shown in Figure 1.6b, attempts to accurately project the real surface segment onto each voxel, cutting it at each's limits. Thus, in this method, each voxel has an effective area such that when summed, the total surface segment's effective area is much closer to the original data. There is a problem with this method as well, however. Although each surface

voxel's effective area may be an acceptable approximation, the number of particles that impact each is still dependent on the exposed area of each. The arrows in Figure 1.6b illustrate how surface voxels 2 and 3 each have two possible sides at which rays may impact, while voxel 1 only has one side. This construction naturally skews the tracking of particles that hit the surface as some voxels will always have more exposed sides than others. Since the goal is to determine when a particle impacts a surface segment, the original surface segment will use the new, better effective area, but will be counting too many particles has having impacted it. In the example of Figure 1.6b, voxel 2 has a very small effective area, but a high probability of receiving incident particles due to having two exposed sides, while voxel 2 has a large effective area and only one side from which to receive particles.

An acceptable solution to these problems involves a smoothing of the surface voxels over their original boundary data and associating each original segment with a set of voxels that represent the discretized version of itself, in surface voxels. In this method, each original boundary segment is given an effective area rather than each surface voxel. This effective area is determined by noting what the first and final surface voxels that have been associated with a given boundary are and then calculating the greatest distance between any of the vertices of both of these voxels. Figure 1.7

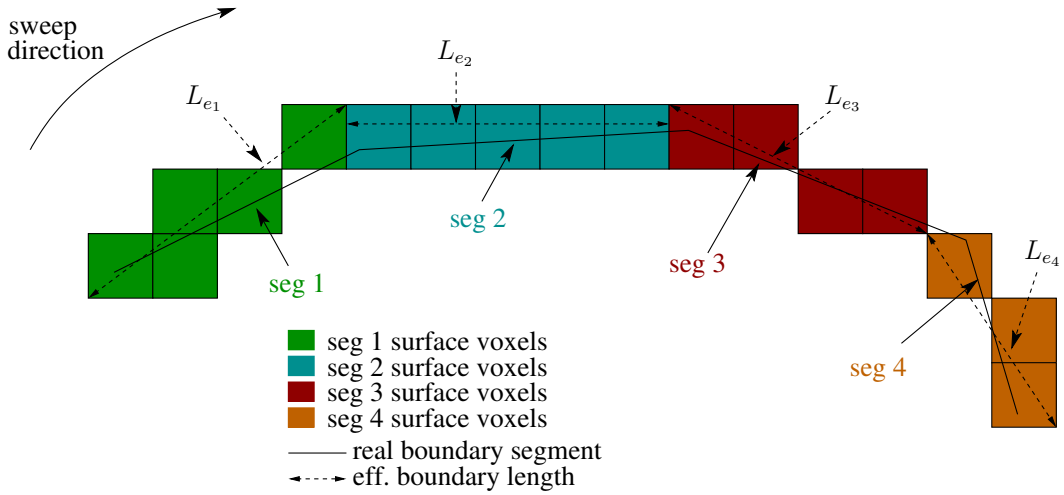


Figure 1.7: Method 3 algorithm results showing four boundary segments, their voxelized representations and sampling effective areas L_e.

illustrates the result of applying this method to a sample set of boundary segments that have been voxelized. The algorithm can be summarized as follows:

(1) Voxelize all boundaries while allowing original boundary segments to posses a unique list of surface voxels that represent them.

(2) Calculate L_e per original boundary segment as the distance between the most extreme vertices of all voxels associated with the segment.

(3) When a particle hits a voxel, add the particle to the voxel's associated original parent boundary segment, forming the total sum of impacted particles to that segment.

(4) Use each boundary segment's own L_e when calculating final sampled flux and variables.

(5) Clear each segment's impacted particle list each time a new sample period is started.

Note that there is a sweep direction indicated. This is required to enforce the necessity of each surface voxel to have exactly one parent boundary segment, thus making each segment's list of children surface voxels unique so as to avoid double-counting particles that impact near interfaces between segments. The endpoints of each segment may lie in different surface voxels because of this, but the smoothing effect of the algorithm's choice of effective lengths means this is not a problem. For example, the endpoint of segment 1, in the sweep direction, lies inside a surface voxel associated with segment 2. Originally, when segment 1 was voxelized, the aforementioned surface voxel belonged to segment 1 as its final surface voxel, but once segment 2 started voxelization, the voxel was claimed by segment 2 and removed from segment 1's list, making the surface voxel shown in the figure the true final voxel of segment 1.

A final refinement in this method is desirable. For boundaries such as segment 2 in Figure 1.7 that contain entirely co-linear voxels, L_e will be calculated as being higher than is necessary as the algorithm will draw a diagonal between the most extreme vertices of the segment's surface voxels. Thus, before calculating L_e, the segment's set of surface voxels should be classified as one of the four types illustrated in Figure 1.8. Each set of surface voxels that has been assigned to a

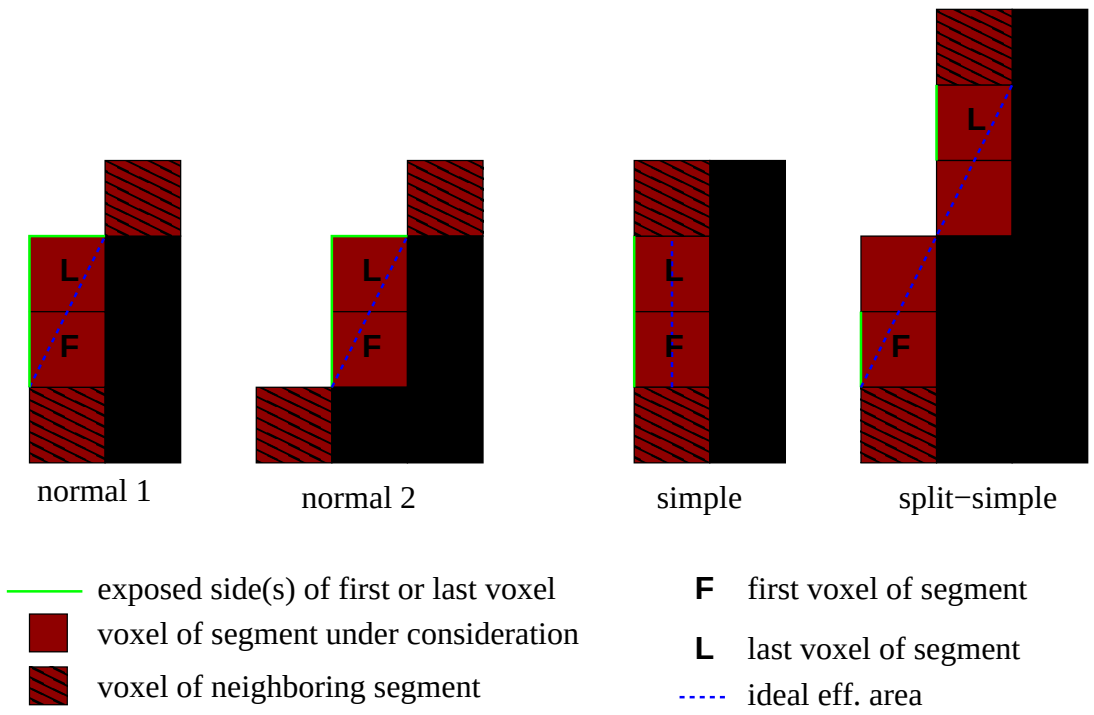


Figure 1.8: Method 3: identifying surface voxel set types per boundary segment so as to allow for refinement of the method.

boundary can be identified as either of these types by noting whether the first and final (or "last", as in the figure) are co-linear or not, and then noting how many exposed sides each has. If they are co-linear and both first and final voxels have different numbers of exposed sides then the set is of type normal 1 or normal 2. If they are co-linear, each with one exposed side, the set is simple. Finally, if they are not co-linear and each has only one exposed side, the set is split-simple. The reason for the definition of split-simple sets is because only checking for number of exposed sides and not for co-linearity would mark a split-simple as just simple. If a set is simple (i.e. *not* normal 1, normal 2 or split-simple), L_e is calculated as shown, rather than the corner-to-corner distance that the algorithm would otherwise use.

This smoothed voxel area method alleviates the problems in the first two methods and is used in the current version of Voldipar. Because real distances are used, the problem of method 1 is avoided. Because surface sampling is tracked per original boundary segment, rather than per voxel,

the problem of uneven particle tracking of method 2 is mostly avoided, though it is not perfect.

One final consideration of the smoothed voxel method is that it is actually possible to return to a perfect boundary sampling system using its natural narrowing of boundary candidates. If all surface voxels are uniquely associated with a boundary segment, then upon impact with any surface voxel, the discrete ray cast can hand over control to a secondary algorithm which can then efficiently check for exact geometric intersections with the voxel's parent boundary. Since a surface voxel can be originally associated with a different boundary segment, as described earlier, impacts near boundary segment interfaces would require both possible segments to be examined for geometric impact. This only adds one additional check to perform, which is minimal overhead to endure. However, in the current 2D-only version of Voldipar it is unclear whether perfect boundary sampling will create any serious improvement in accuracy, particularly since increasing voxel resolution naturally improves the creation of discrete surface voxels as boundaries. Additionally, although there is minimal algorithmic overhead that is required, molecule movement and surface impact calculations represent roughly one third to one half of DSMC computational effort. Adding more complication is undesirable unless obvious benefit is likely. It is for this reason that exact boundary sampling has not yet been implemented in Voldipar.

1.2.4 Variable–Adaptive Time Steps

One element of efficient DSMC that both Voldipar and Bird's series of advanced codes (e.g. `DS2V and DS3V`) share is the inclusion of an automatically adjustable variable time step. This concept is introduced very briefly by Bird in Ref. [5] but is not discussed in detail. A complete explanation of the theory and usage of variable–adaptive time–stepping, as it is implemented in Voldipar, is provided in this section in order to provide some needed clarification.

Not all locations in the domain of a DSMC simulation will (generally) have the same collision rate. This implies that the mean collision time for any given cell is not only not identical to, but may be drastically different from another cell. Consider the simple example given in Figure 1.9. The cells closer to the object will likely have higher mean collision frequencies than cells far from

the object, i.e. $\nu_1 \approx \nu_2 \gg \nu_9$. This, of course, implies that mean collision times for these cells are related as: $\overline{\Delta t}_{col,1} \approx \overline{\Delta t}_{col,2} \ll \overline{\Delta t}_{col,9}$. Another way to describe this occurrence is to say that the cells with lower collision rates will, for the same amount of time, calculate fewer collisions than those with higher rates. Thus, if a constant time step Δt_{flow} is used in Equation 1.5 for Δt, the less dense cells will needlessly be computing collisions. To avoid the unnecessary computation of

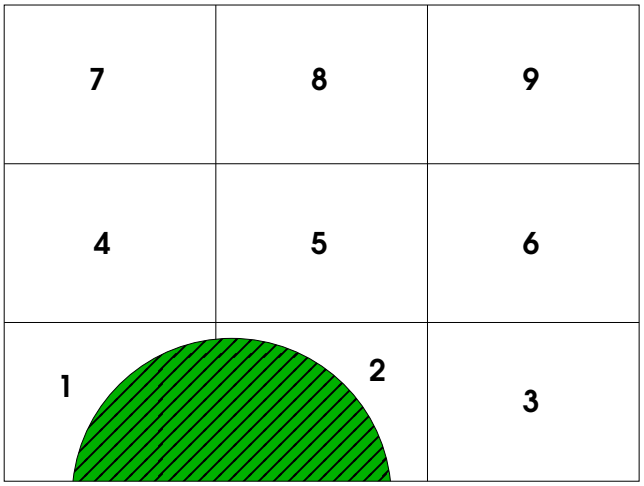


Figure 1.9: A simple example DSMC domain with nine collision cells and a solid object with which particles collide.

collisions in cells where said collisions do not occur as often as other, more dense cells, the time variable Δt_{flow}, as used in Equation 1.5, can be replaced with a characteristic time step unique to each cell that is a function of the mean collision time in that cell, or, in the case of free-molecular flow, some estimate for mean transit time of particle across the cell. Equation 1.5 becomes unique to each cell as

$$N'_{pq} = \frac{1}{2} N_p \overline{N_q} \left(\sigma_T c_r\right)_{pq,max} \frac{\Delta t_c}{V_c} \tag{1.8}$$

where the cell's characteristic time is

$$\Delta t_c = \min\left(\overline{\Delta t}_{col}, \overline{\Delta t}_{tr}\right) \tag{1.9}$$

The mean cell transit time, $\overline{\Delta t}_{tr}$ can be determined a number of ways but is primarily just an approximation based on characteristic cell geometry and mean macroscopic velocity in the cell.

In the case of zero macroscopic velocity in all directions (i.e. a stationary gas) that is not free-molecular, the minimum thermal velocity component can be used. The cell's mean collision time is more specifically obtained. At any given time in the simulation after initialization, a cell's mean collision time can be determined by using any of three methods.

Method 1: Scale the current sample interval.

Scale the most recent sample interval Δt_s by the ratio of the mean number of particles in the cell over the sample ($\overline{N}$) to the number of particles of the sample that have experienced at collision ($2N_{col}$).

$$\overline{\Delta t}_{col} = \frac{\overline{N}\Delta t_s}{2N_{col}} \tag{1.10}$$

One caveat to using Equation 1.10 to determine mean collision times is that it is inaccurate when $\overline{N}$ and N_{col} are sampled with low fidelity. Sampling over more main loop iterations will reduce the statistical scatter in these variables, causing more realistic "measurements" of them, rather than highly varying instantaneous values. However it may still be undesirable to require that $\overline{\Delta t}_{col}$ rely on the sample interval directly as it should remain as separate from the physics of the problem as possible.

Method 2: Use the most obvious definition of mean collision time.

Mean collision time in a cell is the reciprocal of the mean collision frequency $\overline{\nu}$, so a similar equation to Method 1 would be

$$\overline{\Delta t}_{col} = \frac{1}{\overline{\nu}} = \frac{\Delta t_s}{N_{col}} \tag{1.11}$$

This definition gives mean collision times that are close to but still slightly smaller on average than Equation 1.10.

Method 3: Use the general form of $\overline{\nu}$ to augment Method 2's definition.

The collision frequency can be written as

$$\overline{\nu} = n(\overline{\sigma_T c_r}) \tag{1.12}$$

meaning the mean collision time is

$$\overline{\Delta t}_{col} = \frac{1}{(\overline{\sigma_T c_r})} \tag{1.13}$$

In a DSMC algorithm, the collision cross-section $\sigma_T c_r$ of each possible collision is necessarily computed during the collision step. Since it is already available, it is a simple matter to track its mean value $\overline{\sigma_T c_r}$ per cell over the sample interval. The number density n is determined using information that is already readily available as

$$n = \overline{N}/V \tag{1.14}$$

where V is the cell volume. Note that the macroscopic number density may be zero in the event that $\overline{N}$ is zero. This can occur if particles leave the cell over the sample without new particles entering before the calculation of Δt_c takes place. If $\overline{N}$ is zero, the cell should retain its current mean collision time rather than attempting to re-calculate it, as the cell has not gained enough new information to warrant a change.

At the initial stage of a simulation (i.e. at $t_f = 0$, where t_f is the absolute overall simulation flow time), $\overline{\Delta t_{col}}$ can be determined as a function of the domain's initial characteristics, vis

$$\overline{\Delta t}_{col}^{(0)} = \left[\sum_p f_{r_p} \nu_p^{(0)} \right]^{-1} \tag{1.15}$$

$$\nu_p^{(0)} = \sum_q 2\sqrt{\pi} d_{ref,pq}^2 n f_{r_q} \left(\frac{T_i}{T_{ref,pq}} \right)^{1-\omega_{pq}} \left(\frac{2kT_{ref,pq}}{\bar{m}_{pq}} \right)^{\frac{1}{2}} \tag{1.16}$$

where $f_{r_{p|q}}$ is the fraction of species p or q, T and n are the cell's macroscopic overall temperature and number density, respectively, k is Boltzmann's constant, and the rest are species $p{-}q$ reference data. Equation 1.16 is a form of Equation 4.64 from Ref. [3].

The overall flow time step now must be updated at every sampling operation. Each time the domain is sampled, Equation 1.8 is evaluated and stored for each cell. The overall flow time step would ideally be calculated as

$$\Delta t_f = \min \Delta t_c \tag{1.17}$$

where min is taken over all cells. However in practical application, taking the actual minimum over all cells may result in excessively small time steps due to fluctuations in cells' information. Some cells may possess poor information (e.g. cells immediately behind a body, in the highly rarefied wake where few particles exist at any given instant) which may yield a mean characteristic time

that is "artificially" very small or large. As a concession of pragmatism, the flow time may be instead calculated as

$$\Delta t_f = \frac{\overline{\Delta t_c}}{\gamma_t} = \frac{\sum_i (\Delta t_{c,i})}{N_v \gamma_t} \tag{1.18}$$

where $\overline{\Delta t_c}$ is the mean characteristic time of the entire domain calculated by only including cells whose information was valid over the sample interval (i.e. whose δt_c was not zero or infinity). The right-hand-side of Equation 1.18 sums times over all valid cells i and divides by the total number of valid cells N_v to obtain this specific mean value of characteristic time in the domain. Naturally, this time would be too large for some cells. A scale factor γ_t is included to reduce the mean time. This factor becomes a tuning parameter of the algorithm that must be specified by the user. A smaller value of γ_t gives larger flow time steps, but also higher collision and movement fractions (the fraction of cells that actually perform collisions and particles that actually are moved per sample interval).

To examine the effect of changing γ_t, computations of steady flow over the NACA-0012 airfoil in Kn =0.01 Ma =28.40 in molecular Nitrogen (see Section 1.2.7) were performed at various values of γ_t. These results, which compare overall drag force, are given in Table 1.1. Using Equation 1.18

Table 1.1: Effect of scale factor γ_t on flow time step calculation and overall drag of an airfoil section in DSMC (% collide/move are rough averages occurring at steady-state).

γ_t	**Drag** [N]	$\Delta t_{f,min}$ [s]	$\Delta t_{f,max}$ [s]	**% collide**	**% move**	**run time** [s]
3.0	38.30	3.87×10^{-7}	4.36×10^{-7}	30.1	35.1	1432
5.0	38.14	2.33×10^{-7}	2.65×10^{-7}	18.6	22.0	1787
7.0	38.06	1.66×10^{-7}	1.92×10^{-7}	13.4	16.1	2098
9.0	38.08	1.29×10^{-7}	1.50×10^{-7}	10.4	12.7	2360
11.0	38.06	1.05×10^{-7}	1.23×10^{-7}	8.6	10.5	2634
13.0	38.10	8.91×10^{-8}	1.05×10^{-7}	7.3	9.0	2881
15.0	38.04	7.73×10^{-8}	9.08×10^{-8}	6.3	7.8	3127
17.0	38.10	6.81×10^{-8}	8.04×10^{-8}	5.6	6.9	3390

with an adjustable γ_t does not ensure that the overall flow time is advanced at a resolution higher than the smallest characteristic time; which is a requirement for proper DSMC collision-advection process decoupling. However, choosing a low enough γ_t will often yield acceptable results for many

common problems. The airfoil example in Table 1.1 is representative of many similar external flow problems. Acceptable results are often obtained with $\gamma_t \geq 5$, with diminishing returns in accuracy past $\gamma_t = 15$; however, the user's specific requirements for a balance of accuracy and performance will ultimately decide the value of γ_t.

The flow time step will be used until the end of the next sample interval when it is re-calculated. Each iteration of the simulation's evolution contains collision and movement procedures. Thus, each cell and each molecule must now store its own absolute flow time. At each iteration of collision, each cell compares its flow time to the overall flow time. If the cell's flow time is less than the current overall flow time, collisions for that cell are calculated using the characteristic time for that cell, given by Equation 1.9. The flow time for the cell is then advanced by its characteristic time Δt_c.

Similarly to how collision cells are asynchronously advanced through time only when necessary to be physically consistent, individual molecules should be advanced based on where they are in the domain. Each time a molecule is considered for movement, its flow time is compared to the overall flow time. If the molecule's flow time has lagged behind the overall flow time, the molecule is moved a distance that is determined from its current velocity and the characteristic time of the cell in which it currently resides at the beginning of the movement step. This requires the simulation to be capable of quickly and efficiently determining a given molecule's current cell. There may be problems in performance with this requirement, however they are mitigated by various programmatic design methods that won't be discussed here.

As a means of providing additional clarification of the variable-adaptive time advancement method, a detailed example is given that is based on the domain of Figure 1.9. The safety factor α_t is included to slow time advancement by a slight amount in order to ensure that cell and molecule time advancement don't occur faster than they should. Artificially (somewhat arbitrary but mildly consistent with the domain and its inclusion of a solid object) selected cell characteristic times are given in Table 1.2 for a set of consecutive samples, starting at $t_f = 0$. If the number of overall flow time steps per sample $N_{\text{sam}} = 5$, then Figure 1.10 gives a visual representation of the various

Table 1.2: Example domain cell characteristic times over four consecutive samples starting at $t_f = 0$, with $\alpha_t = 0.5$. The first column, $\Delta t_c^{(0)}$, would be the times calculated from the initial domain state, using Equation 1.15. Superscripts denote the sample number. Times are dimensionless.

Cell	$\mathbf{\Delta t_c^{(0)}}$	$\mathbf{\Delta t_c^{(1)}}$	$\mathbf{\Delta t_c^{(2)}}$	$\mathbf{\Delta t_c^{(3)}}$
1	0.20	0.10	0.30	0.20
2	0.40	0.30	0.30	0.20
3	1.00	0.90	0.80	0.80
4	0.50	0.70	0.70	0.70
5	0.70	0.70	0.80	0.70
6	1.00	1.10	1.00	1.10
7	1.40	1.50	1.60	1.50
8	1.80	1.90	1.80	1.80
9	2.00	2.00	2.10	2.10
	$\mathbf{\Delta t_f^{(0)}}$	$\mathbf{\Delta t_f^{(1)}}$	$\mathbf{\Delta t_f^{(2)}}$	$\mathbf{\Delta t_f^{(3)}}$
	0.10	0.05	0.15	0.10

timelines of the simulation as it evolves, starting at $t_f = 0$ (the far-left points of all lines) to infinity, over four samples. The vertical broken lines indicate a point in the simulation at which collisions and movement are considered for all cells (i.e. each iteration of the simulation's main loop). The small notches along each cell's timeline are locations of the cell's current flow time. All cells begin with a flow time of zero. They each have their characteristic times calculated using Equation 1.9, which uses Equation 1.15. The initial overall flow time step $\Delta t_f^{(0)} = 0.10$ is determined and used over the first sample interval, which extends for $N_{\text{sam}} = 5$ overall flow time steps (until the $S^{(1)}$ location). The first time collisions and movement are considered is at $t_f = \Delta t_f^{(0)}$, the first vertical line. Cell 1's current flow time at this point is zero. As zero is less than the current overall flow time, collisions for cell 1 are performed using $\Delta t_{c,1}^{(0)} = 0.20$ and cell 1's current flow time is incremented by this amount, which is where the first notch (with a numeral 1 adjacent to it) is on its line. This procedure is repeated for the remaining cells. As they all are synchronized at zero initially, they all have collisions performed and their flow times advanced to the next notch. The cells with longer characteristic times have their times advanced much more than the those with smaller characteristic times, as can be seen by noting how far down their respective timelines their first notches are. As

the next iteration comes, all cells are once again considered for collision. Cell 1's current flow time is still at notch 1 (0.20), and as the overall flow time of this, the second, iteration is 0.20, the cell's time is equal (i.e. not strictly greater, but this is a detail of implementation) to the overall flow time. Thus, collisions are *not* performed and cell 1's time is not advanced. Cell 2's current flow time is at its notch 1, which is at 0.40, which is well beyond the overall flow time, so it is also skipped; as are the remaining cells at this iteration. At the third iteration, cell 1's current time (0.20, notch 1) is now lagging behind the flow time (0.30), so collisions are performed and its time is advanced to notch 2. At the fourth iteration, cell 1 and cell 2 have times equal to the flow time and are therefore skipped, as are the remaining cells in the example. At the fifth and final iteration of this sample interval, $t_f = 0.50$ and cells 1 and 2 are updated, with others skipped. Immediately after collisions and movement are performed for this iteration, the first domain sampling occurs ($S^{(1)}$). It is important to note that this event occurs after the collision and movement steps as new characteristic times and thus, overall flow time steps are calculated when sampling occurs. This also implies that any collision cell updates at the final iteration of the sample interval should still be using their "current" time information. Recall, as well, that from now onwards, the new cell characteristic time steps are calculated using Equation 1.9 with Equation 1.13. As soon as this sampling occurs, all cells will be using the second column of times in Table 1.2 and the overall flow time step is now smaller: $\Delta t_f^{(1)} = 0.05$. This procedure continues until the end of the simulation.

Table 1.3 provides a reduction of the process a bit further by showing which cells receive updates at which overall flow times. The cell with the smallest characteristic time (cell 1) can be seen to have collisions calculated 10 times, while the cells with very large mean collision times (e.g. cells 7,8,9 etc.) only have collisions calculated once or not at all after the initial calculation at $t_f = 0$, and thus skip most time steps. The integers increase along each cell's row, indicating another update for that cell, and that the integer is where the cell's flow time number will be at the end of that update (e.g. cell 1 will have moved to its third flow time after the completion of the update that occurs for it at $t_f = 0.50$, or the fifth overall iteration).

It can be seen how much of an advantage this advancement method gives if one considers

Table 1.3: A visual account of updates (collisions calculated and cell time advancement) in the example problem. A filled entry indicates an update was performed on that cell at the corresponding overall flow time. The number in the entry indicates the notch in Figure 1.10 to which the cell was advanced at the conclusion of the collision calculation. Cells 7 and 8 can be seen to have only two each, with cell 9 having only 1 by the end of the 20th iteration at $t_f = 2.0$.

Cell	$\mathbf{t_f^{(1)}}$	$\mathbf{t_f^{(2)}}$	$\mathbf{t_f^{(3)}}$	$\mathbf{t_f^{(4)}}$	$\mathbf{t_f^{(5)}}$	$\mathbf{t_f^{(6)}}$	$\mathbf{t_f^{(7)}}$	$\mathbf{t_f^{(8)}}$	$\mathbf{t_f^{(9)}}$	$\mathbf{t_f^{(10)}}$
	0.10	0.20	0.30	0.40	0.50	0.55	0.60	0.65	0.70	0.75
1	1		2		3			4		5
2	1				2					
3	1									
4	1					2				
5	1									2
6	1									
7	1									
8	1									
9	1									

Cell	$\mathbf{t_f^{(11)}}$	$\mathbf{t_f^{(12)}}$	$\mathbf{t_f^{(13)}}$	$\mathbf{t_f^{(14)}}$	$\mathbf{t_f^{(15)}}$	$\mathbf{t_f^{(16)}}$	$\mathbf{t_f^{(17)}}$	$\mathbf{t_f^{(18)}}$	$\mathbf{t_f^{(19)}}$	$\mathbf{t_f^{(20)}}$
	0.90	1.05	1.20	1.35	1.50	1.60	1.70	1.80	1.90	2.00
1	6		7		8			9		10
2	3		4		5			6		7
3		2							3	
4		3						4		
5					3					
6		2								
7					2					
8									2	
9										

that the total number of updates over the entire simulation (as given in this example) would be based off of the smallest (estimated) characteristic time, which (when multiplied by $\alpha_t = 0.5$) is 0.05, and assumed constant. This would mean that there would be 40 updates for every cell, totaling 360 updates for the 2.0 time units of evolution. With variable-adaptive time advancement, the total number of updates is only 34, a 90% improvement. Of course, the exact performance benefit for this method is dependent on the physical nature of the problem being solved. Domains with uniform density everywhere would not benefit much. However, most engineering problems of interest contain sufficient disparities within their domains that implementing variable-adaptive time steps is likely worth the effort. For example, Bird remarks that "The average time step in

the benchmark case is seven times the size of the minimum time step and the ratio of the largest to the smallest step would be more than 100" [5], in reference to the cylinder benchmark case of Section 1.2.6.2.

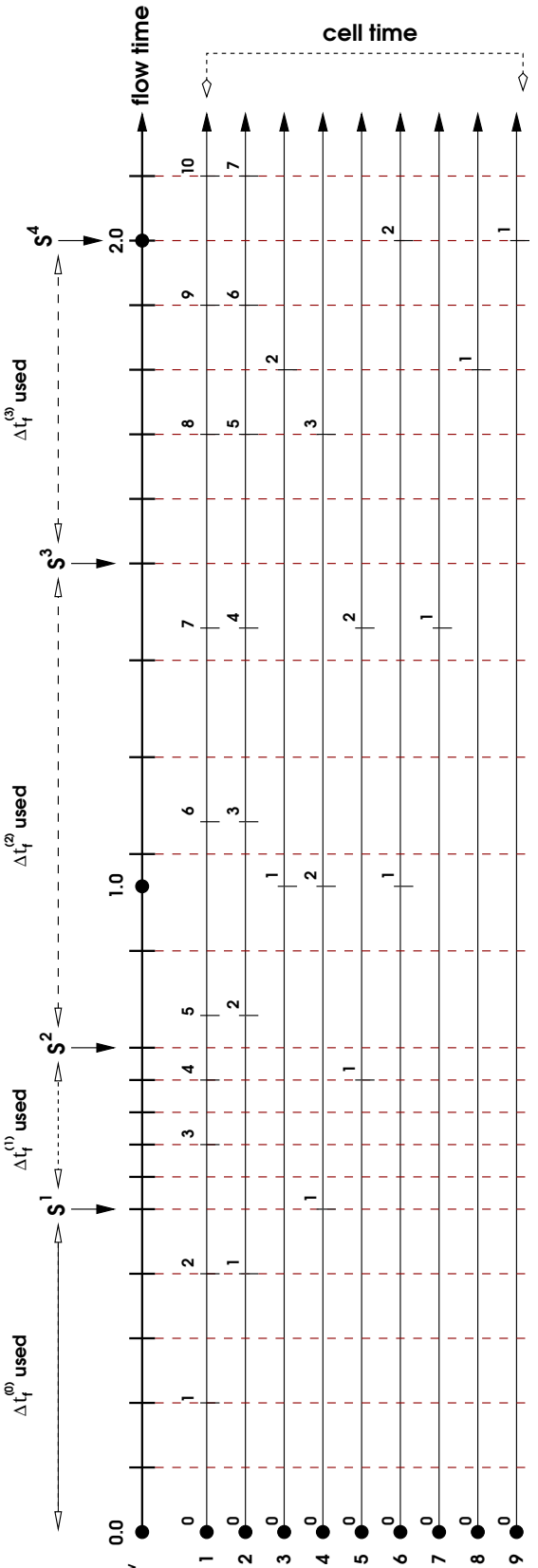


Figure 1.10: The timelines of the overall simulation flow and individual cells in the example are given, with locations of interest denoted by small vertical notches along each cell's timeline. Each notch indicates a time to which cells are advanced at the conclusion of their most recent collision or movement computation (the occurrence of which appears at the vertical line immediately to the right of the previous numbered notch on any given timeline, e.g. the computations that advance cell 1's time to notch 2 occur at the third vertical line, or $t_f = 0.30$).

1.2.5 Stream Boundary Considerations

Bird's molecule entry model is based on usage of Equation 1.19 (Equation 4.22, from Ref. [3]), which is the analytical result for particle flux through a surface as a function of surface temperature and entry velocity.

$$\frac{\beta \dot{N}_{\text{inc}}}{n} = \frac{1}{2\sqrt{\pi}} \left[e^{-s^2 \cos^2(\theta)} + \sqrt{\pi} s \cos(\theta)(1 + \text{erf}(s \cos(\theta))) \right] \tag{1.19}$$

The angle θ is the angle between the incident velocity vector and the surface normal and $s = \beta V_{in}$ is the molecular speed ratio, where V_{in} is the incident bulk speed. In the `DSMC2` code (which is used for verification of the first problem in Section 1.2.6), the stream boundary entry fluxes are determined by imposing a kind of throughput or advective velocity on the stream boundaries. This method may result in physically realistic results for the particular problem of a supersonic flat plate, but some doubts may be raised about its application to general cases.

Problems arise from the imposition of an exit velocity at certain locations where flow is assumed to exit. Similar to how outflow boundary conditions often present an ambiguity in continuum CFD methods, there is not always an appropriate choice to make for an exit velocity (or even an exit boundary at all) in DSMC. In the case of the supersonic flat plate problem shown in Figure 1.14, the right boundary is the assumed exit. When the entry stream boundary on the left is considered, the angle θ in Equation 1.19 is zero (assuming the inflow velocity–a quantity that is specified–has no vertical component) and $s = \beta_L V_{in,L}$, where β_L is the reciprocal of the most probable molecular speed using prescribed external gas conditions at the inlet (i.e. the chosen free stream conditions). This causes the maximum possible flux through the left boundary. When the right boundary is considered, `DSMC2` calculates an *inflow* flux (that is, flow from the right external region, through the right boundary and into the domain, or molecules moving to the left) across this boundary by defining $s = -\beta_R V_{in,L}$ and assuming θ is still zero. That is, either setting s to be negative or choosing $\theta = \pi$ would give the same thermal backflow result in Equation 1.19. The result of this assumption creates almost no flux into the domain from the right boundary since $V_{in,L}$ is very large in this problem.

Ideally, however, a proper definition of a DSMC stream boundary will not be dependent on knowledge of any prescribed or assumed flow conditions. Calculating inflow flux from an "exit" boundary using the the inflow speed from an actual entry boundary does not model a true stream boundary unless the external stream, beyond that boundary (in this case, outside the domain to the right) is desired to be moving at the inlet velocity as well. If the right stream boundary is supposed to be a static gas, then the inflow flux calculation should take into account the orientation of the normal vector of the boundary, which defines "in" as to the left. This implies that the parameters for the right boundary should be such that $\theta = 0$ (always, if the velocity is entirely normal to the boundary, and directed in with respect to that boundary) and $V_{in,R} = 0$. The latter equality is important to enforce, as it causes s to be zero, which results in a proper static gas diffusion flux–a quantity much higher than that which results from $s < 0$, as Bird imposes in the flat plate problem.

The imposition of a throughput velocity by Bird in `DSMC2` is not incorrect, but simply non-generalized. When handling a DSMC stream boundary, a general code requires the following quantities to be defined on the boundary (or just-externally): number density, temperature and bulk velocity[4] . Any given stream boundary's inflow flux is then determined by calculating β, then s for the boundary, and then the entry flux from Equation 1.19, noting that θ should always be between zero and $\pi/2$.

To simplify the prescription of stream boundary conditions, Voldipar allows stream boundaries to be specified in the most natural way. Each stream boundary represents an interface to an external stream which has a bulk velocity, density and temperature. These three parameters are given by the user and the code determines by means of an inner product rather than a direct angle θ what the inflow normal component of velocity is. This defines the flow as simply existing outside the domain with some naturally entering the domain through the boundary.

When considering outflow streams, a prescribed external stream velocity is applied but is not necessarily physically accurate. Information about external stream conditions is not possessed by

[4] Additionally, it is possible that the external regions may have different species composition than inside the domain. In this case, the molecular mass and gas constant, used to calculate β must be prescribed as well.

the simulation at the beginning of its evolution. For the flat plate problem, the imposition of an outflow speed (and subsequent minimal backflow) is acceptable for that particular problem, but not in general. The resulting influx across the right boundary is effectively zero, which implies a vacuum condition would be equally acceptable at that location. In the proceeding verification of this problem, Voldipar uses a vacuum condition when comparing with Bird's `DSMC2` code. The top boundary's entry flux is small, but non-trivial, and not zero. To impose the existence of a flowing stream at the same velocity as the inlet, entry flux at the top boundary is equivalent to using $V_{in,T} = 0$, as no bulk flow is present that crosses the boundary. This results in a flux identical to a static gas diffusing across any boundary. Once molecules are placed into the domain just inside of the top boundary, the free steam velocity (which is actually the user-specified top-external velocity) is added to their thermal velocity. Future improvements in this way of specifying stream conditions are discussed in the following section.

1.2.5.1 Improvements in Stream Conditions

The problem of lack of information about the external domain when imposing stream conditions may seem similarly difficult to resolve as it is in continuum CFD. Consider the example of the flat plate problem where the right boundary is chosen to be a static gas. This implies that the entering flow crosses the domain and exits into an external region where there is no bulk flow. To model this, $V_{in,R}$ would be set to zero, as mentioned earlier, effectively modeling a static gas. However, this assumption fails to account for the inevitable change in realistic boundary conditions as a function of the evolution of the external environment. That is to imply that the static gas to the right of the right boundary would no longer remain static once the bulk flow from the domain has begun to push it rightward. This occurrence is illustrated in Figure 1.11. The static gas stream condition at the right boundary is no longer valid once the simulation has affected the external flow. In fact, the imposition of a negative normal inflow speed (i.e. an outflow speed) becomes appropriate. In the initial phase of the simulation (perhaps only the first time step), shown on the left of Figure 1.11, there is a non-zero number of molecules from the external stream

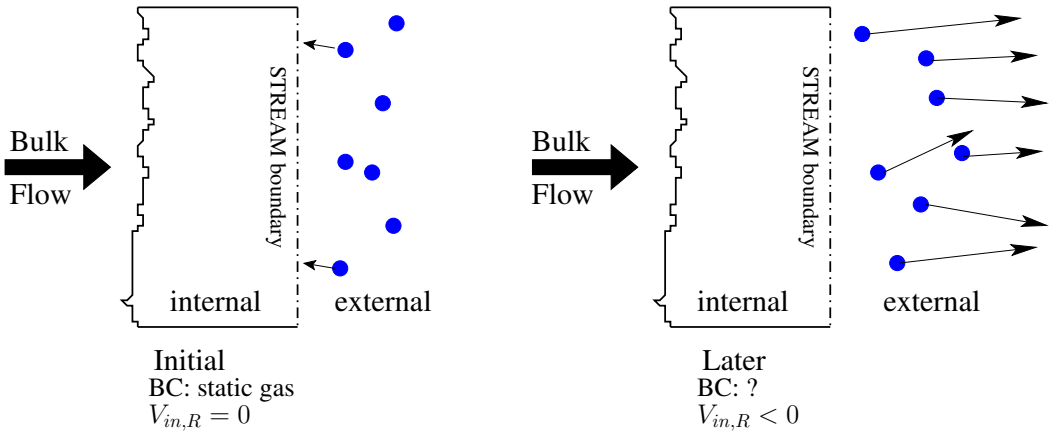


Figure 1.11: The stream boundary condition as imposed on an assumed exit boundary will affect the external molecules and their velocity distributions after some arbitrary (possibly steady-state) time in the evolution of the simulation.

which pass into the domain from the external region due to thermal motion. In the later stages of the simulation, shown on the right of Figure 1.11, very few, if any, molecules move back into the domain. This is the thermal backflow limit phenomenon that occurs as the bulk flow increases in magnitude. Thus, the entry fluxes for each phase of the simulation will be different. Furthermore, DSMC's standard procedures introduce new molecules in the domain, said to have recently crossed the boundary, at each iteration of the simulation's main time loop. The number of molecules is determined from the number flux. If the static gas stream condition is maintained throughout the simulation, a buildup of molecules just inside of the boundary occurs. Although molecules that enter are mostly immediately knocked out again by high-Ma flows (as in the flat plate example), more are reintroduced the very next time step. This causes the cells immediately inside a stream boundary of a static gas to always contain, somewhat artificially, a higher density of particles than their immediate neighbors. This increase in density can be seen in Figure 1.12. The imposition of a static gas for all time at the exit compresses the oncoming flow, causing molecules incident on the right boundary to be unable to easily leave the domain, as they collide with the external region's virtual static particles as if they were actually a "soft" or semi-permeable wall.

Consider yet another example in which this outflow uncertainty is more prevalent. In DSMC,

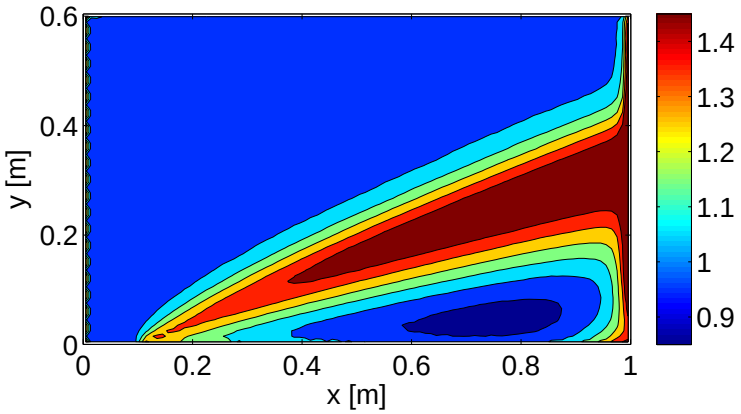


Figure 1.12: Nondimensional density n/n_{in} for imposing static gas stream conditions with Voldipar, for all time, at the right boundary in the supersonic flat plate problem. The entering molecules block the oncoming flow from exiting smoothly just inside the boundary.

if an internal boundary is given as a solid object, defined by a 2D surface comprised of N_S points, it is possible to define a problem in which this surface is used as a source for entering molecules. A set of external[5] boundary properties such as stream velocity, density and temperature are prescribed on the boundary at each point (or segment) and molecules enter the domain via these locations. Figure 1.13 illustrates this concept.

The domain boundaries are also stream conditions, meaning that they are interfaces with a different external gas. Both the internal gas and external gas are assumed to be static at the start of the simulation. Initially, the effusing boundary will cause the static gas in the domain to move according to the algorithms of DSMC. However, what occurs at the domain boundaries is unclear, and it is important to known how these boundaries' properties should be specified. If the typical constant stream conditions are imposed, the resulting overall result inside the domain, and particularly close to the boundaries, will be incorrect for the aforementioned reasons relating to

[5] In this case, "external" would refer to the region inside the solid boundary, as it acts as a source, which models a hole in the domain.

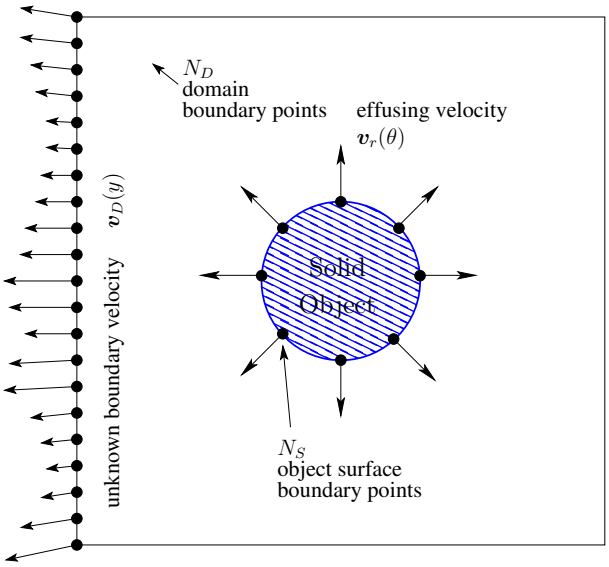


Figure 1.13: DSMC domain effusion problem. The outer domain boundaries' properties (e.g. velocity is shown) are unknown at the beginning of the simulation. Only the left domain boundary is shown in detail.

Figure 1.11. Thus, the domain boundaries need to be updated as the simulation evolves in order to reflect this effusion. If they are not, and the boundaries remain unchanged as a static gas, the artificial "soft" boundary will again be the result.

Since a user may not wish to specify an outflowing stream condition or a constant static gas stream condition (for the preceding reasons), some balance between the two is required. Fortunately, DSMC allows an easy formulation of a dynamic stream boundary condition. Each time step, the macroscopic flow (bulk) velocity just inside the boundary can be used as a distributed velocity on the boundary. Thus, in the example of Figure 1.11, the right boundary would eventually evolve into an outflow stream boundary. This method would also work in reverse, allowing a static gas external to a boundary to be sucked into the domain, thus becoming a bulk-inflow or entry stream condition. Related approaches which use information from outside the computational region have been proposed. Lilley and Macrossan [43] described an acceptance-rejection formulation of generating entry particles as if they were to start from external locations and diffuse into the domain, thus providing a varying number flux of particles per time step. This method also avoids

the need for velocity distribution cutoffs, allowing all possible velocities to be selected rather than a finite region that is defined by the user.

This interpolation scheme may prove too costly to perform very often. Therefore it may be desirable use a reduced-order model to provide some means of determining the external domain boundary properties as a function of the prescribed inner, effusing boundary's properties. Consider this problem as the relation of $3N_S$ parameters to $3N_D$ parameters, where the factor of three comes from the requirement of knowing velocity, temperature and number density at each point. If both components of velocity are considered separate parameters, this factor is increased to four in 2D, and 5 in 3D. This relation would be required to update at each time step of the simulation for transient cases (the most general case) or could be given as a steady-state relation of the domain (a common, but less general case). Various uncertainty quantification methods (e.g. Separated Representation [1]) that relate a number of inputs to a different number of outputs may be useful in this arrangement of the problem. It is worth mentioning that the "black box" that connects the inputs (inner boundary effusion stream properties) and outputs (domain boundary properties) is DSMC itself. Thus it involves much averaging and uncertainty in its operation anyway. But for the purposes of such an analysis, it should still be considered deterministic.

1.2.6 Verification

Prior to using Voldipar to provide information on rarefied gas flow, a verification was performed of the code in its current state. Two canonical problems were chosen, both being heavily examined by Bird and others. The supersonic leading-edge or flat plate problem is a simple 2D problem that requires no geometry considerations. The hypersonic round cylinder problem is an example of non-trivial geometry in rarefied flow that verifies the code's geometry mechanisms as well as the overall DSMC method's implementation. Both of these problems were computed using Voldipar and results compared to those obtained with Bird's DSMC2 code, and others. The `DSMC2` code is not a production code, and may be considered to be obsolete. However it is still validated as giving acceptable results by Bird (Ref. [3], Chapter 14). As its source is readily available, a proper

direct comparison can be made. Therefore, this code was used to validate the flat plate problem. The cylinder problem contains geometry that `DSMC2` cannot handle, as that code has no internal geometry model. That problem was compared using available computational results from Ref. [5] and [62].

1.2.6.1 Hypersonic Leading-Edge

The supersonic flat plate problem is without any need for complicated geometry as it simply uses a distribution of boundary conditions over one of its natural boundaries. It is constructed by setting the domain geometry and boundary conditions as shown in Figure 1.14. The boundary conditions of type "Stream" indicate an interface with the undisturbed free stream of gas. The

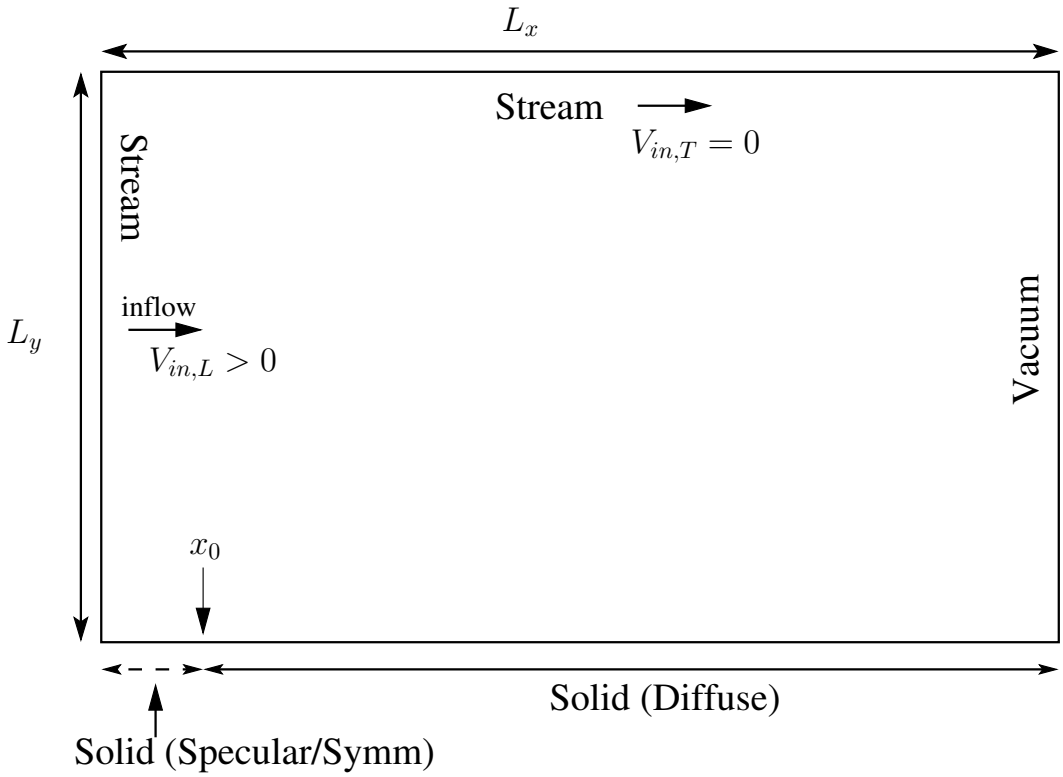
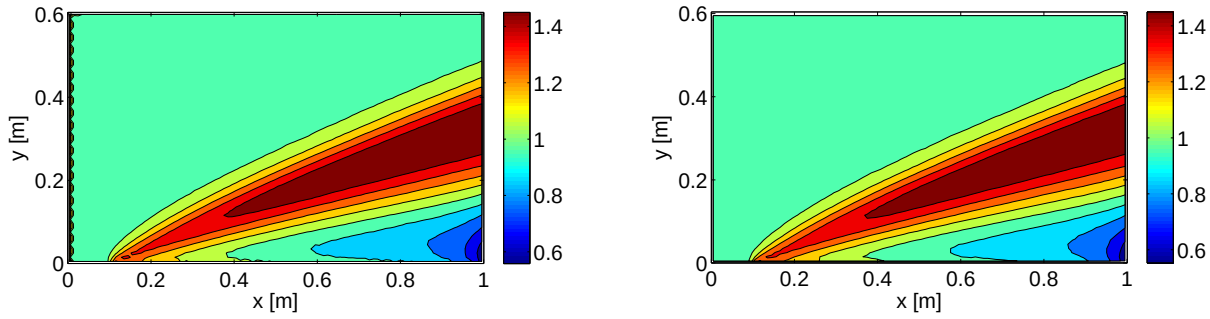


Figure 1.14: Geometry and boundary conditions of the supersonic leading edge or flat plate example problem.

specific parameters of the problem are given in Table 1.4, where the gas in use is molecular nitrogen

and the collision model is Variable Hard Sphere (VHS). Rotational energy re-distribution is the only internal mode that is modeled. The lower boundary of the rectangular domain is split into two regions. The first 10% is a solid surface using specular reflection (which is identical to a symmetry condition). The remaining 90% is a solid surface that uses standard diffuse reflection. Thus, a transition occurs that effectively models a flat plate with a leading edge at x_0. The initial particle distribution was determined by randomly positioning particles in cells according to the constant number density n_{in}, constant temperature T_{in} and constant initial velocity $[V_{in}\ \ 0]$. Each simulation was run from $t = 0$ to $t = t_{\max}$ using a constant time step of Δt, at which point steady-state had been long-achieved, and approximately 100,000 samples had been taken. The cell sizes and spacing in `DSMC2` were set to be uniform and constant.

The first data to compare are the domain statistics. These are quantities that are tracked in both codes as various measures of accuracy and performance, which include mean collision separation distance and collision acceptance ratio.

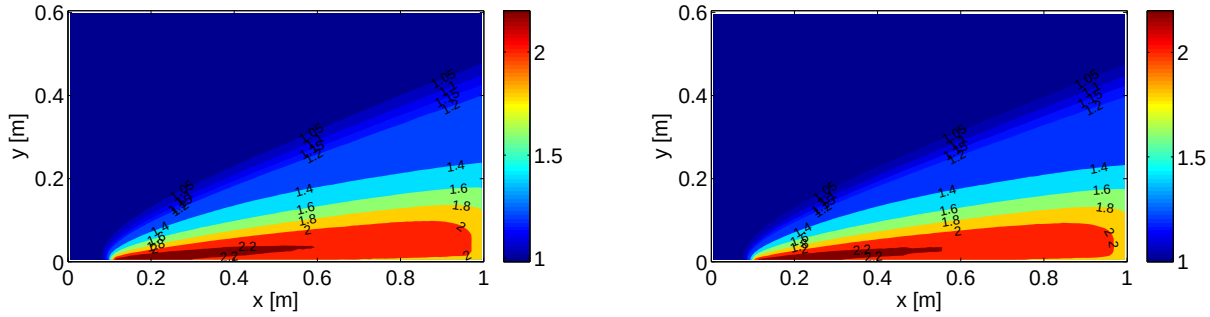


(a) Voldipar n/n_{in}

(b) DSMC2 n/n_{in}

Figure 1.15: Nondimensional number density contours for the supersonic flat plate problem at $t_{\max}$, or 0.7976 s after steady state was achieved: $\mathrm{Kn} = 0.01$, $\mathrm{Ma}_{in} = 4.0$

Field variables for each code are compared in Figures 1.15 to 1.17. Field comparisons indicate Voldipar gives close results to `DSMC2`, with most of the disparity between the two being at the edges



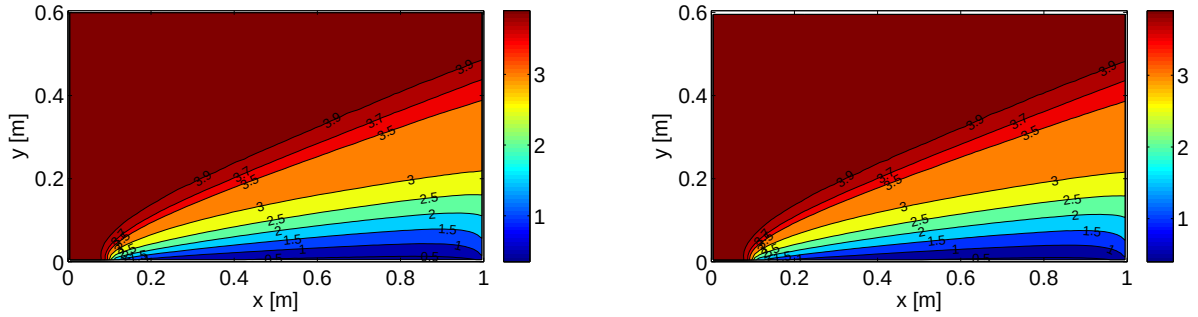
(a) Voldipar T_{ov}/T_{in}

(b) DSMC2 T_{ov}/T_{in}

Figure 1.16: Nondimensional overall temperature contours for the supersonic flat plate problem at $t_{\max}$, or 0.7976 s after steady state was achieved: Kn = 0.01, Ma$_{in}$ = 4.0

of the domain. The apparent jagged edges of Figure 1.15a near the entry boundary are likely due to the necessary oversampling that occurs due to the use of voxels as data containers, as well as due to a per-voxel distribution of entry molecules. The latter consideration is different from `DSMC2` as in that code, the entire left boundary is considered a single surface over which entry molecules will enter (i.e. there can be no distribution of surface properties over the boundary that would create different entry fluxes at different locations) where as in Voldipar, the option to specify boundary surface properties as functions of the coordinate along the surface necessitates these properties being unique down to the level of a voxel–the smallest geometric unit allowed in the code. To validate this assumption further, the plate surface properties were examined. Figures 1.18 to 1.21 show comparisons of various surface quantities of interest, non-dimensionalized and scaled according to Equations 1.20 to 1.23, where e is the natural exponential base, m_g is the molecular mass and k is Boltzmann's constant (Coordinates are shifted such that $x/\lambda_\infty = 0$ is the leading edge of the plate.).

$$N_{fm} = \frac{n_{in} e}{2\sqrt{\pi}\beta_{in}}, \ \beta_{in} = \sqrt{\frac{m_g}{2kT_{in}}} \tag{1.20}$$



(a) Voldipar Ma

(b) DSMC2 Ma

Figure 1.17: Mach number contours for the supersonic flat plate problem at $t_{\max}$, or 0.7976 s after steady state was achieved: Kn = 0.01, $\mathrm{Ma}_{in} = 4.0$

$$C_p = \frac{2(p - p_{in})}{\rho_{in} V_{in}^2}, \; p_{in} = \frac{\rho_{in}}{4\beta_{in}^2} \tag{1.21}$$

$$C_f = \frac{2\tau_{xy}}{\rho_{in} V_{in}^2} \tag{1.22}$$

$$C_h = \frac{2\epsilon}{\rho_{in} V_{in}^3} \tag{1.23}$$

Equation 1.20 is the free-molecular non-dimensional number flux based on the free stream values of temperature and density. The remaining equations are definitions of typical aerodynamic coefficients, with p_{in} being static thermal pressure and ϵ in Equation 1.23 being a general energy quantity that represents either the total net energy transfer to the plate, incident translational energy, reflected translational, incident rotational or reflected translational, as shown in Figure 1.21.

There appears to be sufficient agreement between Voldipar and `DSMC2` for all plate surface variables. The only noticeable difference is at approximately seven to ten mean free paths from the plate leading edge where C_p and number flux are slightly lower for Voldipar. Energy coefficients (Figure 1.21) are unaffected, being the most agreeable of the present surface variables in the comparison.

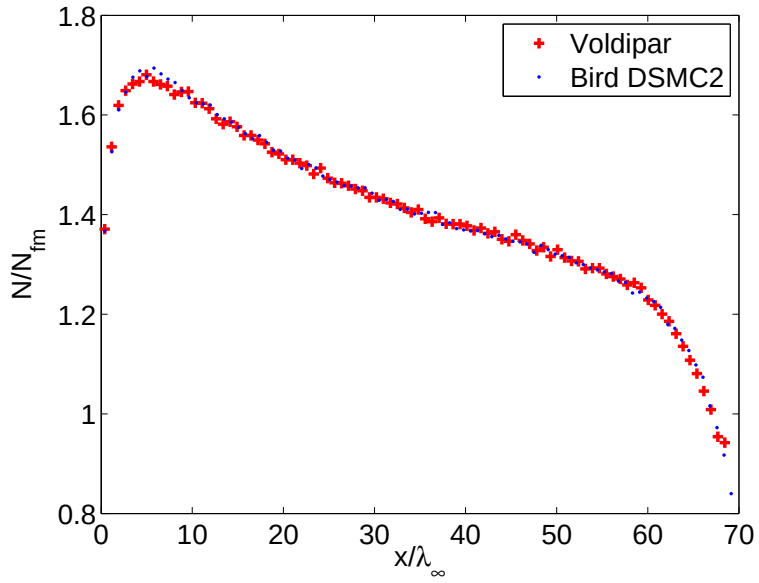


Figure 1.18: Normalized number flux to the plate surface over the steady-state sample interval.

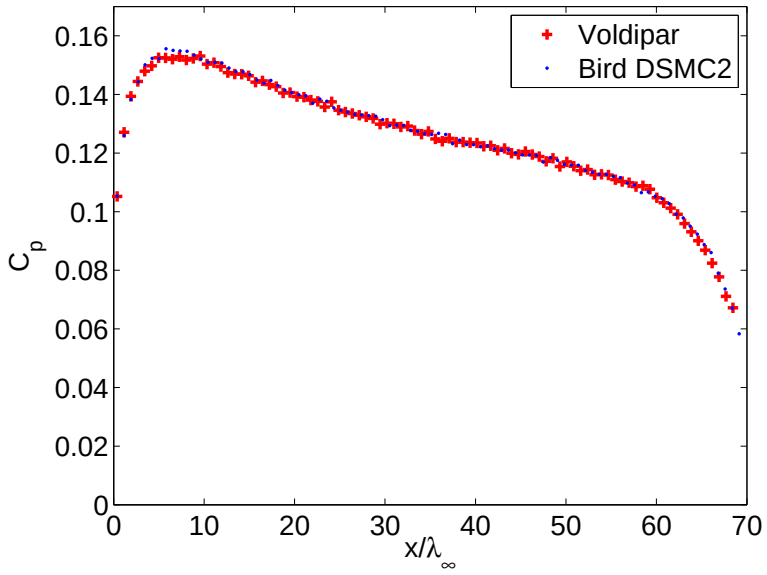


Figure 1.19: Pressure coefficient over the plate surface over the steady-state sample interval.

Table 1.4: Supersonic flat plate problem variable values and parameters.

Description	Symbol	Value	Unit
Mach number	Ma	4.0	
Knudsen number	Kn	0.01	
Ratio of real-to-simulated particles	F_N	1.75×10^{15}	
Leading edge location	x_0	0.10	m
Domain length	L_x	1.00	m
Domain height	L_y	0.60	m
Num. cells in x	N_{cx}	100	
Num. of cells in y	N_{cy}	100	
Num. of subcells per cell (each dir.)	N_{sc}	2	
Constant time step	Δt	4.0×10^{-6}	s
Time to steady state (imposed)	t_{ss}	0.0024	s
Time to end	$t_{\max}$	0.80	s
Num. time steps per sample	N_{sam}	2	
Temp. of inflow stream	T_{in}	300	K
Temp. of solid bottom wall	T_w	500	K
Inflow number density	n_{in}	1.0×10^{20}	m^{-3}
Inflow speed (aligned)	V_{in}	1412.5	m/s
Gas molecular mass	m_g	4.65×10^{-26}	kg
Gas viscosity power	ω	0.74	
Gas reference temp.	T_{ref,N_2}	273	K
Gas reference diameter	d_g	4.17×10^{-10}	m
Gas rotational deg. of freedom	ζ	2	
Rotational relaxation constant	Z_{rot}	1/5	
Voxel resolution (Voldipar only)	n_{vox}	160	

Table 1.5: Steady-state DSMC statistics comparison (at $t = t_{\max}$).

	Voldipar	Bird's `DSMC2`
Molecules Present	37,717	37,441
Molecular Moves	7,532,560,434	7,548,037,584
Num. Pair Selections	1,226,612,705	1,223,014,573
Num. Collisions	642,814,332	641,883,494
Coll. Accept. Ratio	0.524056	0.524837
Mean Coll. Sep.[m]	0.003121	0.003267
Num. Entered Total	44,199,827	-
Num. Removed Total	44,196,424	-

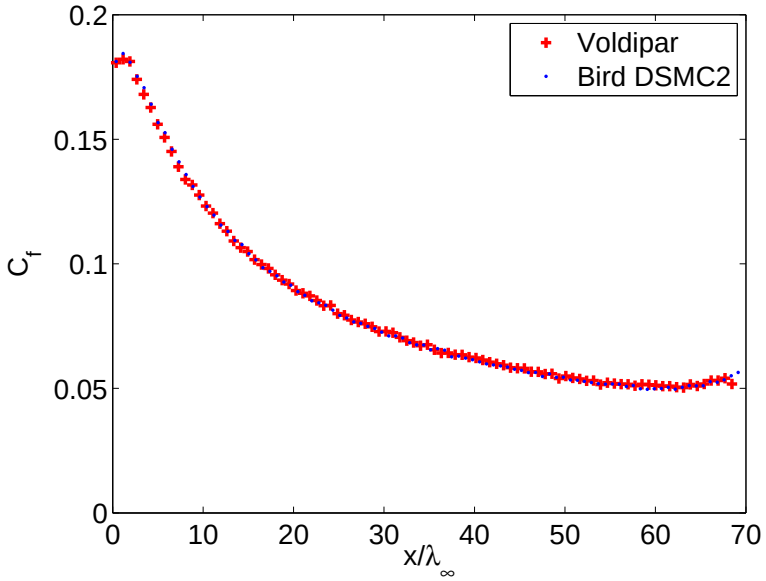


Figure 1.20: Skin friction (shear) coefficient over the plate surface over the steady-state sample interval.

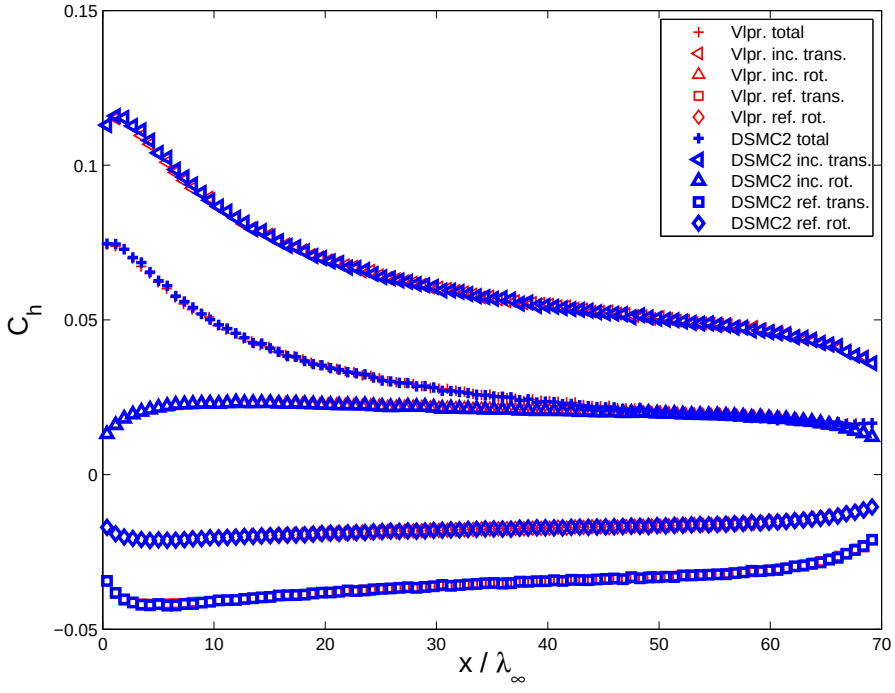


Figure 1.21: Heat transfer (partial energy) coefficients over the plate surface over the steady-state sample interval.

1.2.6.2 Hypersonic Cylinder

A far more interesting problem than flow over a plate, is hypersonic flow over a solid object. In a classic benchmark case, a round half-cylinder is placed in the domain for a high–Mach number flow to intercept. The degree of rarefaction of the gas in this case is set rather low at Kn = 0.01 so as to model what should be close to continuum flow. Thus, the expectations of this problem involve the appearance of a shock region ahead of the cylinder and a recirculated zone just aft of it, the size of which being a function of Ma [6] . Figure 1.22 gives the geometry and boundary conditions of the problem. Specific parameters are identified in Table 1.6. A vacuum condition is used at the right boundary once again, for similar reasoning as was presented for the flat plate problem. The cylinder is placed with its center point at x_0 downstream from the inlet where it is modeled as a solid, diffuse wall. The remaining sections of the bottom boundary are set to be specular solid or symmetry conditions, as the problem is geometrically symmetric about the bottom boundary. The gas in use is atomic Argon and the collision model is Variable Hard Sphere (VHS). Rotational energy internal modes are irrelevant for this gas as it has zero degrees of rotational freedom. The variable-adaptive time step scheme described in Section 1.2.4 is used in Voldipar, while `DS2V` also uses a type of variable time stepping that should be almost identical, with the exception of the parameter α_t which tunes Voldipar's version.

The code used for comparison in this problem was Bird's `DS2V` code. Some of the parameters in Table 1.6 do not correspond perfectly to those in `DS2V`. The number of cells in each direction is constant in Voldipar and cells are rectangular and uniformly spaced [7] . In `DS2V`, the collision and sample cells are separated, and each set is non-uniform in size and spacing. Thus, the total number of collision cells should be compared. Both are set at approximately 40,000 collision cells. Additionally, the time to steady state and F_N are quantities that are imposed in Voldipar, but obscured in `DS2V` where they are implicitly determined (F_N is calculated as a function of a

[6] Typically this region's qualities are a strong function of Reynolds number (Re). However, in this highly-compressible regime, Mach number dominates as the flow is already so strongly inertial

[7] Cells in Voldipar are basically rectangular, but are adjusted to fit the geometry down to voxel resolution for the purposes of calculating effective volume and collisions.

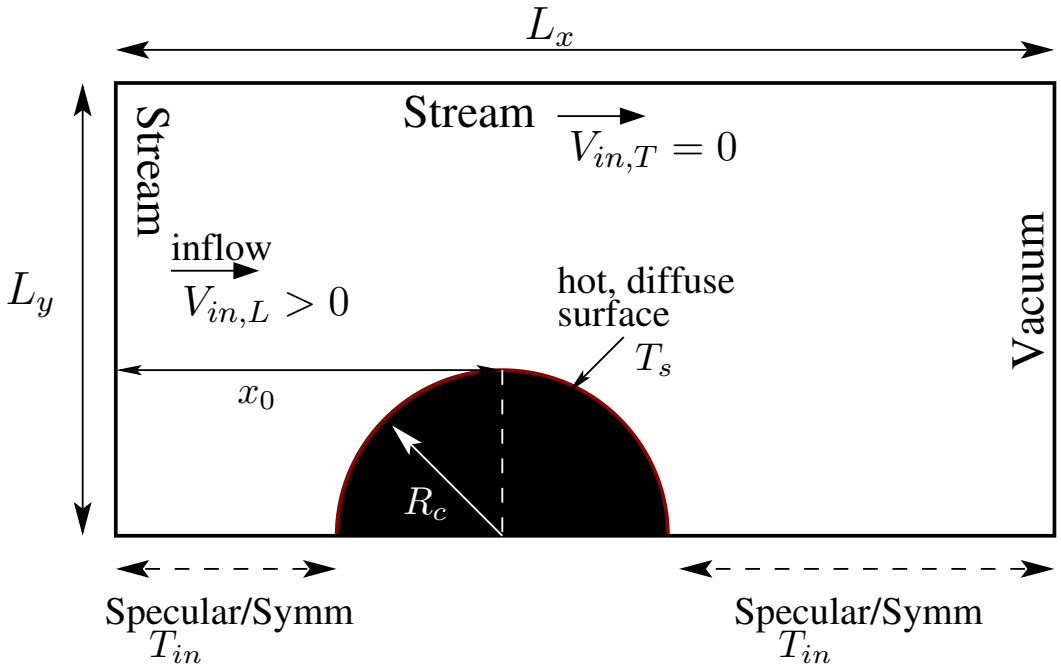
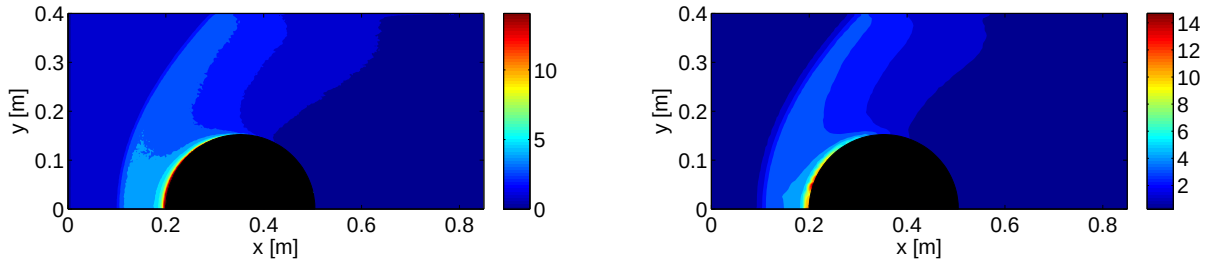


Figure 1.22: Geometry and boundary conditions of the hypersonic round cylinder example problem.

user-defined amount of initial memory with arbitrary scaling and steady state is determined from analyzing the statistical fluctuations of the total number of molecules).

Macroscopic field variables for each code are compared in Figures 1.23 to 1.25. The contours of `DS2V` were created via interpolation to a uniform query grid as the code only provides field data per cell where each cell is non-uniformly spaced and sized. The query grid's resolution was also of coarser quality than the cell grid for Voldipar. This is because `DS2V` separates sampling and collision cells and only provided roughly 6000 sampling cells worth of field data for this example and Voldipar's sampling was performed at each of the 40,500 collision cells present in the domain. Thus, `DS2V`'s contours appear somewhat artificially smoother. The overall quality is similar to Voldipar's results, however. The location, thickness and temperature profile of the the shock is captured by both codes equally well. The recirculated region aft of the cylinder is also present where bulk velocities are nearly zero. Density shows the greatest difference, but this is primarily due to choice of contour scaling. Voldipar had a few cells for which number density was unnaturally



(a) Voldipar n/n_{in}

(b) DS2V n/n_{in}

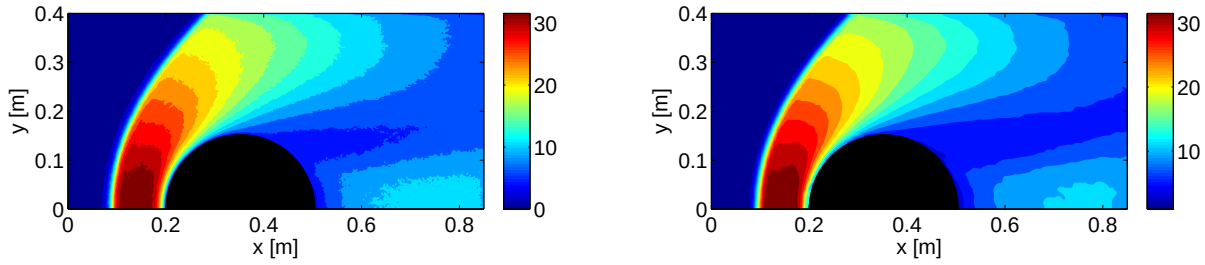
Figure 1.23: Nondimensional number density contours for the hypersonic cylinder problem at $t_{\max}$, or 0.0015 s after steady state was achieved: $\mathrm{Kn} = 0.01$, $\mathrm{Ma}_{in} = 10.0$ (cylinder body artificially superimposed)

high, which is likely a slight defect due to particles getting "stuck" at a few key surface voxels. These cells were artificially set to the highest natural nondimensional density of approximately fifteen to allow for presentable plotting. This problem does not seem to affect the overall field or surface results but will be examined in the future. `DS2V` also had some strange behavior in sampling in that it presented very high densities inside the cylinder–a region where no cells should exist and no particles should be present (the solid black patch that represents the cylinder body obscures this).

Selected surface quantities are compared in Figures 1.26 and 1.27, with C_p defined as

$$C_p = \frac{2p}{\rho_{in} V_{in}^2} \tag{1.24}$$

Voldipar's voxel-boundary sampling mechanism (see Section 1.2.3) naturally introduces some ambiguity in surface sampling operations. The scatter that is seen in both plots near the leading side of the cylinder is likely partially due to this issue. However, this scatter is also mentioned by Bird in Ref. [5] where it is remedied by introducing separate collision and sample cells and adapting and body-fitting both kinds of cells. Voldipar does not currently offer non-uniform cells

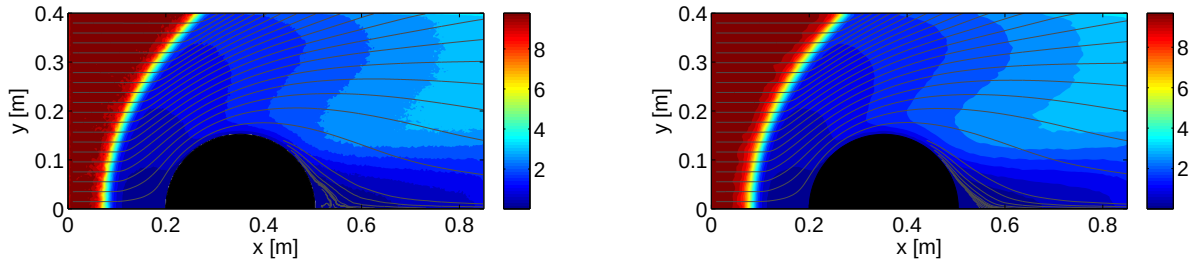


(a) Voldipar T_{ov}/T_{in}

(b) DS2V T_{ov}/T_{in}

Figure 1.24: Nondimensional overall temperature contours for the supersonic flat plate problem at $t_{\max}$, or 0.0015 s after steady state was achieved: Kn = 0.01, Ma$_{in}$ = 10.0 (cylinder body artificially superimposed)

or separated cell types, so this scatter may be unavoidable until these features are implemented. However, despite the scatter present, general trends show that Voldipar is acceptably close to `DS2V`. Additionally, a simpler, single value to compare is the full-cylinder drag force. The sectional drag force results were 39.90 N/m for `DS2V` and 40.25 N/m for Voldipar. When the same problem was run in Voldipar using a constant time step of $\Delta t = 1.20 \times 10^{-7}$ s, this value was slightly smaller at 40.10 N/m. Further still, running the computation for longer time after steady state tends to reduce the drag force slightly. The value for a total run time of $t_{\max} = 0.01$ s with variable time steps was 40.17 N/m. The accepted value for this problem is typically given at approximately 40 N/m[5].



(a) Voldipar Ma

(b) DS2V Ma

Figure 1.25: Mach number contours and selected streamlines for the supersonic flat plate problem at $t_{\max}$, or 0.0015 s after steady state was achieved: $\mathrm{Kn} = 0.01$, $\mathrm{Ma}_{in} = 10.0$ (cylinder body artificially superimposed)

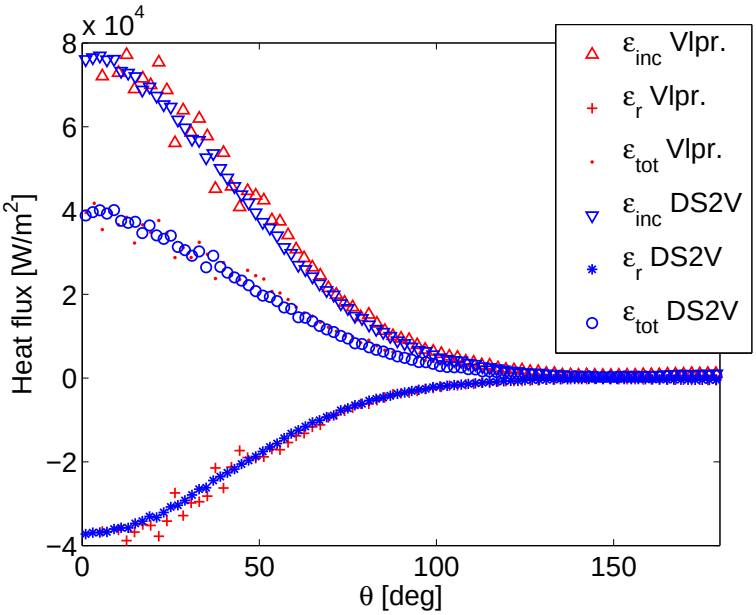


Figure 1.26: Dimensional partial heat fluxes to the cylinder surface over the steady-state sample interval.

Table 1.6: Hypersonic cylinder flow problem variable values and parameters.

Description	**Symbol**	**Value**	**Unit**
Mach number	Ma	10.0	
Knudsen number (based on $2R_c$)	Kn	0.01	
Ratio of real-to-simulated particles	F_N	5.7244×10^{14}	
Cylinder center location	x_0	0.3524	m
Cylinder radius	R_c	0.1524	m
Domain length	L_x	0.85	m
Domain height	L_y	0.40	m
Num. cells in x	N_{cx}	300	
Num. of cells in y	N_{cy}	135	
Num. of subcells per cell (each dir.)	N_{sc}	2	
Variable time step scaling (Vlpr. only)	α_t	0.5	
Time to steady state (imposed)	t_{ss}	0.0015	s
Time to end	$t_{\max}$	0.003	s
Num. time steps per sample	N_{sam}	5	
Temp. of inflow stream	T_{in}	200	K
Temp. of cylinder surface	T_s	500	K
Inflow number density	n_{in}	4.247×10^{20}	m^{-3}
Inflow speed (aligned)	V_{in}	2634.1	m/s
Gas molecular mass	m_g	6.630×10^{-26}	kg
Gas viscosity power	ω	0.74	
Gas reference temp.	T_{ref,N_2}	1000	K
Gas reference diameter	d_g	3.595×10^{-10}	m
Gas rotational deg. of freedom	ζ	0	
Rotational relaxation constant	Z_{rot}	1/5	
Voxel resolution (Voldipar only)	n_{vox}	800	

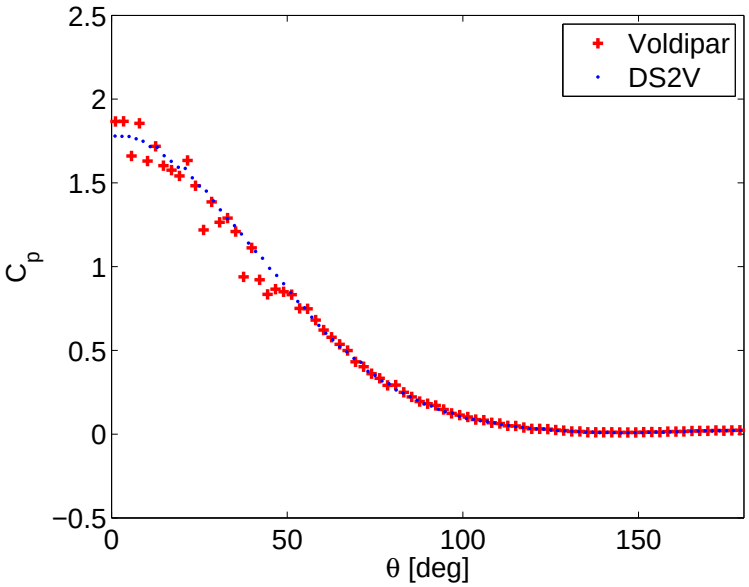


Figure 1.27: Pressure coefficient over cylinder surface over the steady-state sample interval. $\theta = 0$ corresponds to the leading side of the cylinder.

1.2.7 Airfoil Example

Some additional verification was performed using a NACA0012 symmetric airfoil section in 2D, with a chord length of 1 m. Although there are no results with which to directly compare, this problem was selected to test Voldipar's capability to quickly introduce new geometry and fluid conditions. A number of cases were run for the airfoil under the general conditions of Table 1.7 for molecular Nitrogen. Boundary conditions were set to be a stream inflow at the left, vacuum on the right, and stream at the top and bottom. The airfoil's centroid, situated at its quarter-chord, and along its chord line is positioned at (x_0, y_0). A range of angles of attack $0 \leq \alpha \leq \pi/2$ was performed for four magnitudes of Knudsen numbers (shown in Table 1.7) that span the transition regime region. A steady-state solution was assumed at the time of $t_{ss} = 0.002\,\mathrm{s}$, which may not be the most optimal choice. However, the purpose of this study is to choose a consistent set of parameters and perform a large batch of computations using DSMC rather than attempt to verify the physical result of this particular problem. Figures 1.28 to 1.31 show contours for the chosen case of Kn = 0.1 at $\alpha = 46°$.

Of particular interest in this example is the ill-defined shock that lies ahead of the body in Figure 1.31 and the attached flow that leaves the trailing edge of the foil in a thin layer. The expected low-pressure region behind the airfoil is also present. At this angle of attack, it experiences an upward force (lift) and rightward force (drag) both on the order of 1 N. As forces and moments as functions of dynamic state are of interest for characterizing rigid-body dynamics in rarefied gas flows, the full set of data for all angles and all Kn were aggregated so as to compare lift, drag and pitching moment as the airfoil underwent these changes in state. Figures 1.32 and 1.34 depict each change in magnitude of Kn corresponding to an equivalent scaling of forces and moments exerted on the body as would be expected. The airfoil also experiences much more drag than lift, with a maximum Lift/Drag ratio occurring at approximately 30° for all Kn but Kn = 10, where it is maximum at 24°. Pitching moment is signed such that negative moment implies a rotation towards $\alpha = 0$, what is considered a stabilizing moment for an initial deflection of $\alpha > 0$. This moment

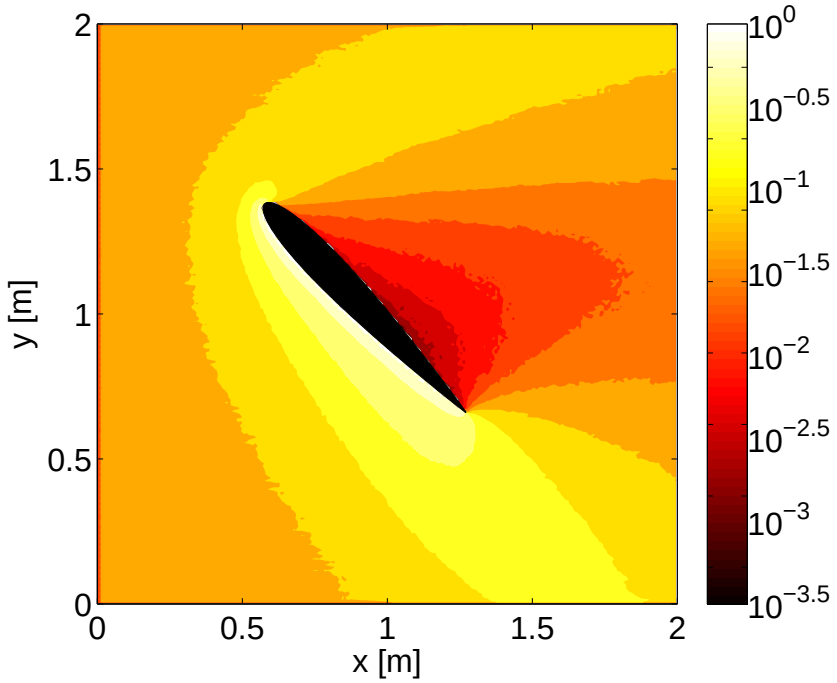


Figure 1.28: Nondimensional number density contours for a NACA0012 airfoil in hypersonic flow of Nitrogen at $\alpha = 46°$.

is taken about the airfoil's imposed centroid, or quarter–chord. When evaluating moment about a location farther aft of this location along the chord line, pitching moment begins to inflect, switching sign and creating an unstable system as would be expected as the center of pressure remains within a region centered around roughly 50% from the leading edge. The center of pressure itself is a function of α, which implies the location of the centroid must remain fore of the entire region within which the center of pressure moves during a change in α. This dynamic margin is plotted for a similar case with this airfoil in the free-molecular limit as Figure 3.5.

While this problem is of minor interest practically, it is a demonstration of the maturity of Voldipar. This capability is required to begin investigating rigid-body dynamics problems and providing the source functions described in Chapters 3 and 4.

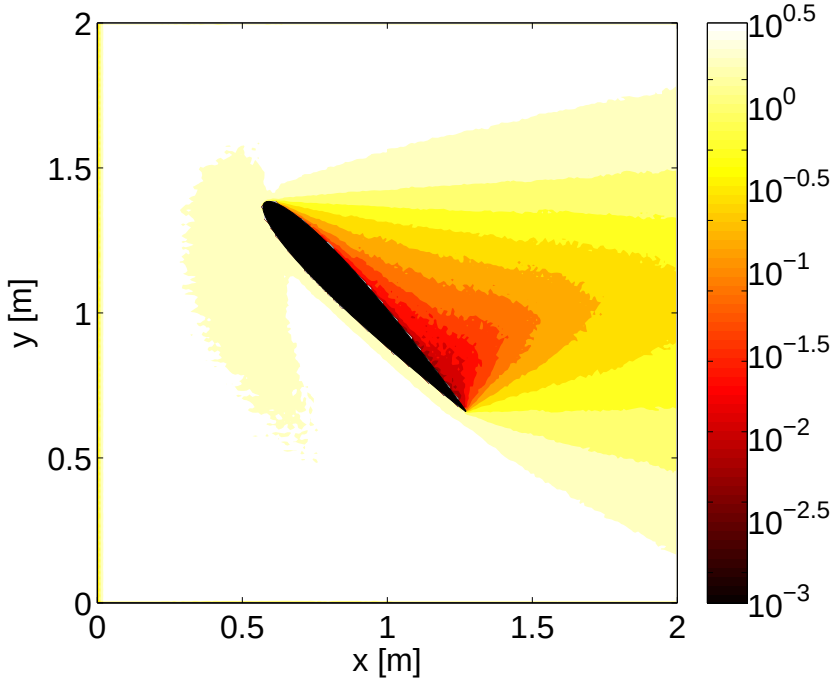


Figure 1.29: Stagnation pressure ratio contours for a NACA0012 airfoil in hypersonic flow of Nitrogen at $\alpha = 46°$.

Table 1.7: Hypersonic NACA0012 airfoil section flow problem variable values and parameters for four different values of Kn .

Description	**Symbol**	**Value**	**Unit**
Mach number	Ma	10.0	
Knudsen numbers (based on chord)	Kn	0.01, 0.10, 1.00, 10.00	
Ratio of real-to-simulated particles	F_N	2.75×10^{15} 2.00×10^{14} 2.50×10^{13} 2.30×10^{12}	
Centroid location	(x_0, y_0)	(0.5, 1.2)	m
Domain length	L_x	2.00	m
Domain height	L_y	2.00	m
Num. cells in x	N_{cx}	200	
Num. of cells in y	N_{cy}	200	
Num. of subcells per cell (each dir.)	N_{sc}	2	
Variable time step scaling (Vlpr. only)	α_t	0.5	
Time to steady state (imposed)	t_{ss}	0.002	s
Time to end	$t_{\max}$	0.005	s
Num. time steps per sample	N_{sam}	4	
Temp. of inflow stream	T_{in}	200	K
Temp. of airfoil surface	T_s	300	K
Inflow number density	n_{in}	1.294×10^{20} 1.294×10^{19} 1.294×10^{18} 1.294×10^{17}	m^{-3}
Inflow speed (aligned)	V_{in}	2882.78	m/s
Gas molecular mass	m_g	4.65×10^{-26}	kg
Gas viscosity power	ω	0.74	
Gas reference temp.	T_{ref,N_2}	273	K
Gas reference diameter	d_g	4.17×10^{-10}	m
Gas rotational deg. of freedom	ζ	2	
Rotational relaxation constant	Z_{rot}	1/5	
Voxel resolution (Voldipar only)	n_{vox}	500	

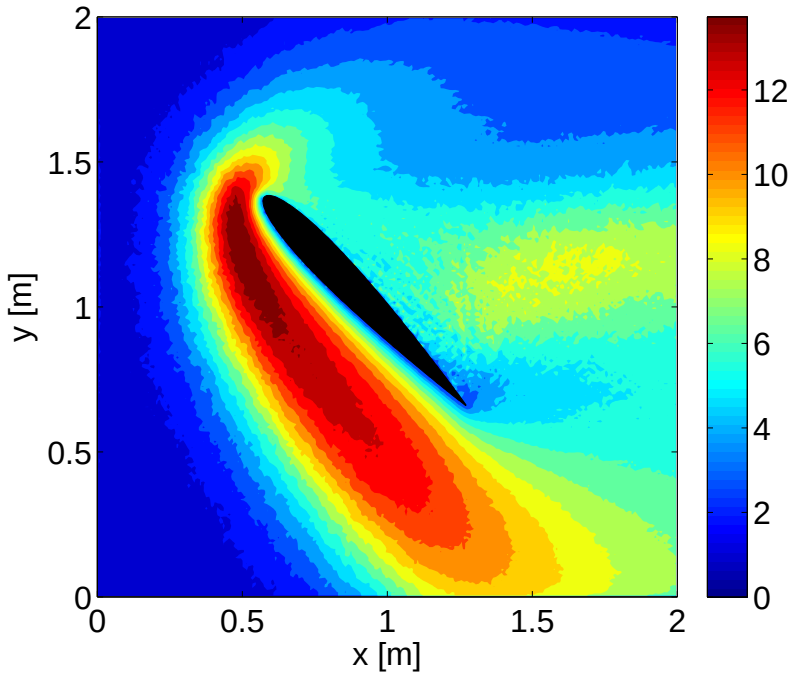


Figure 1.30: Nondimensional overall temperature contours for a NACA0012 airfoil in hypersonic flow of Nitrogen at $\alpha = 46°$.

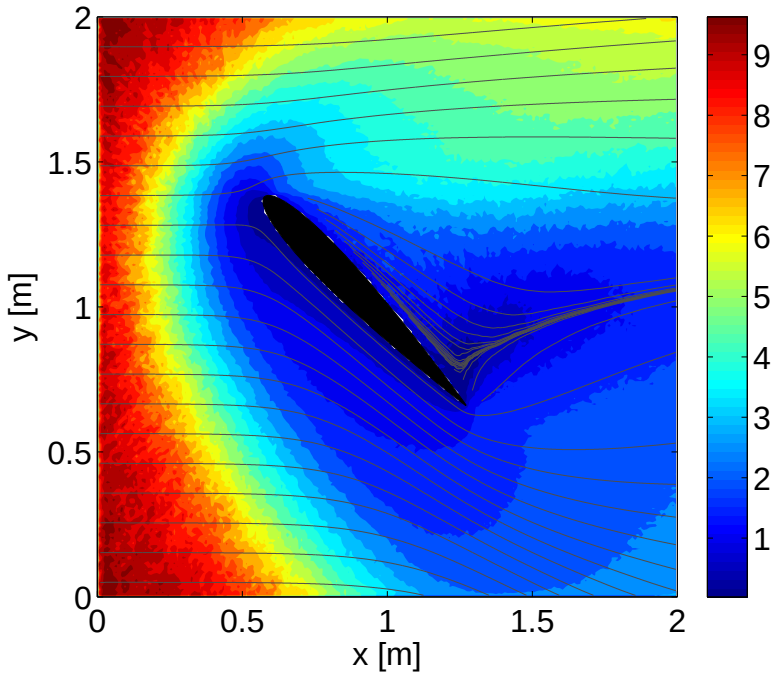


Figure 1.31: Nondimensional Mach number contours and selected streamlines for a NACA0012 airfoil in hypersonic flow of Nitrogen at $\alpha = 46°$.

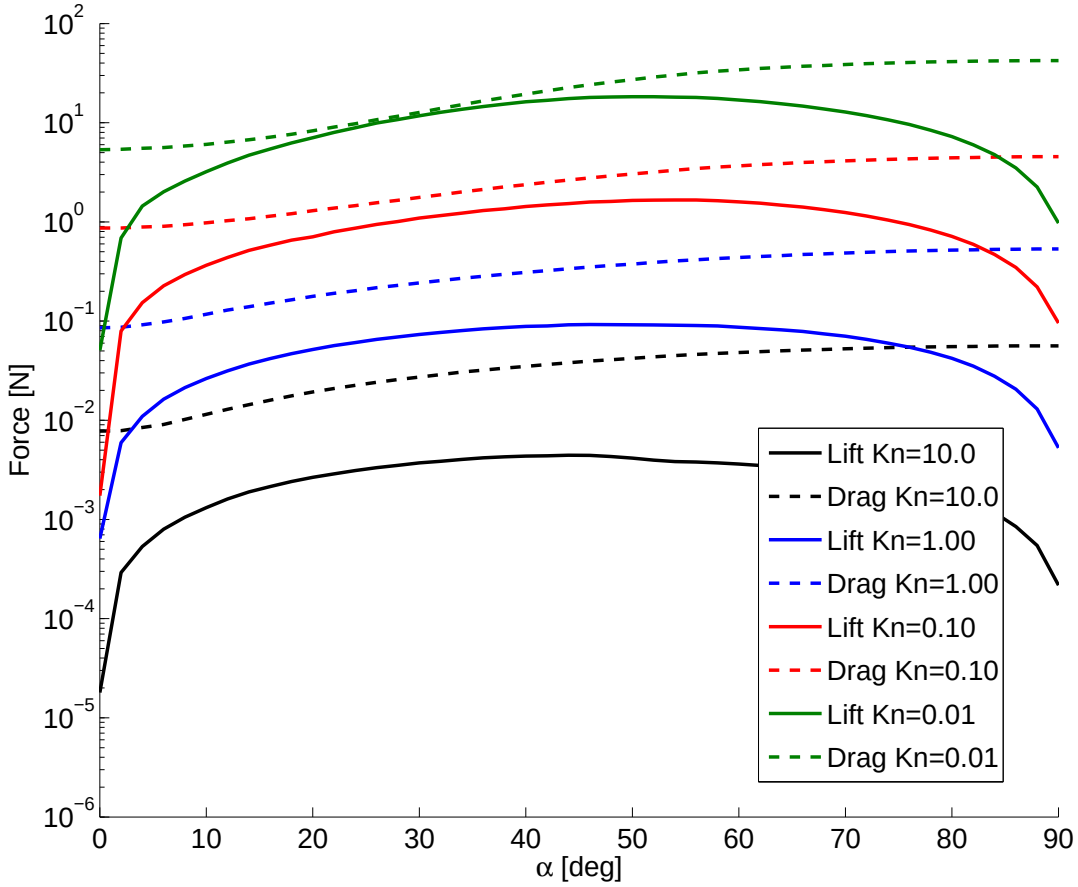


Figure 1.32: Lift and drag of a NACA0012 airfoil section in hypersonic flow of Nitrogen over the entire range of $0 \leq \alpha \leq \pi/2$ for each Knudsen number.

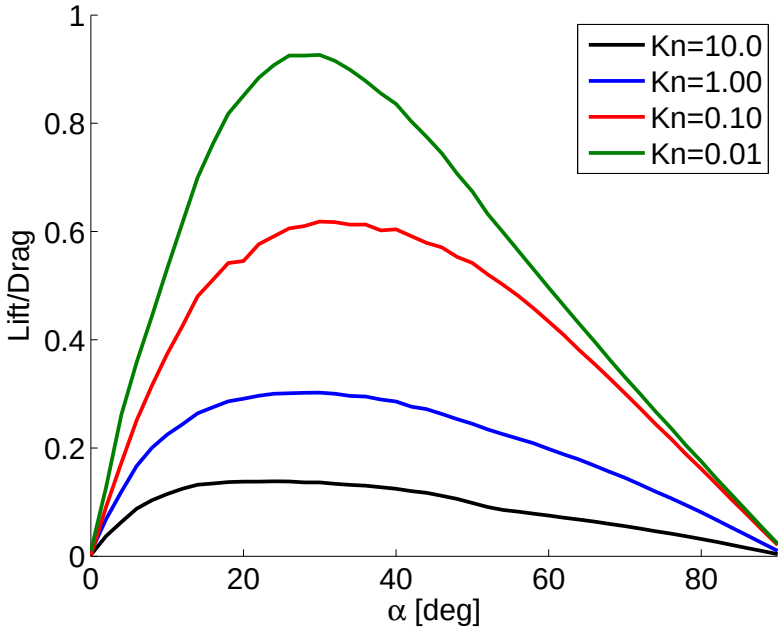


Figure 1.33: Lift/Drag ratio of a NACA0012 airfoil section in hypersonic flow of Nitrogen over the entire range of $0 \leq \alpha \leq \pi/2$ for each Knudsen number.

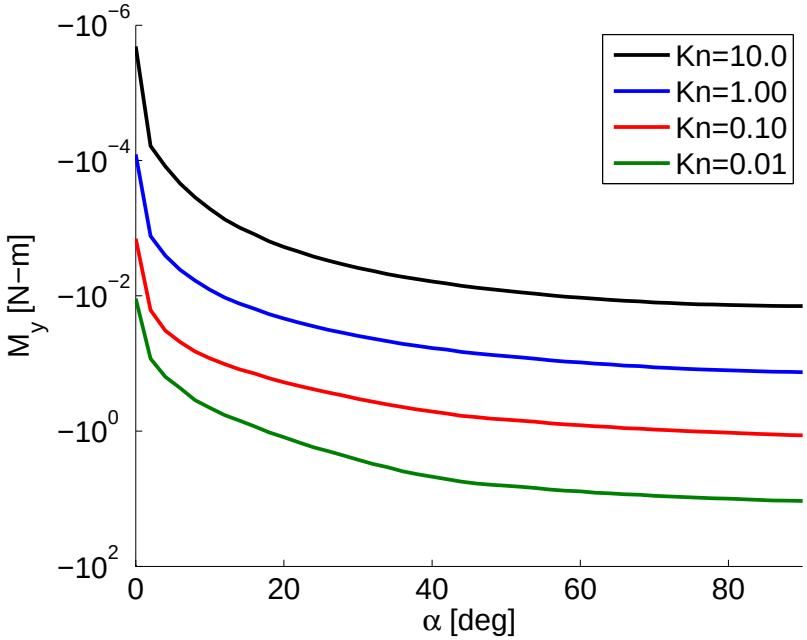


Figure 1.34: Pitching moment about the quarter–chord of a NACA0012 airfoil section in hypersonic flow of Nitrogen over the entire range of $0 \leq \alpha \leq \pi/2$ for each Knudsen number. The moment is signed such that negative implies stabilizing.

Chapter 2

Implementing Gas-Surface Interaction Models in DSMC

Accurate analysis and simulation of rarefied gas dynamics problems require the knowledge of appropriate boundary conditions of the Boltzmann equation. These conditions take the form of Gas-Surface Interaction (GSI) models which describe how energy and momentum is transferred to/from a surface and how reflecting molecules are scattered following a surface impact. Much of the fidelity of rarefied gas dynamics simulations depends on the accuracy of the submodels employed therein. In the case of DSMC, submodels of molecular collisions, movement and surface interactions comprise the code. Collision modeling must be further divided into models of internal modes, some of which were mentioned briefly in Section 1.1. This aspect of discrete particle interaction can become quite complicated and can include chemical reactions that would occur at or near collision events. Molecular movement is primarily a function of the geometry of the code. For Voldipar, this is discussed in Section 1.2.1 with additional relevant information discussed in subsequent subsections. Remaining, then, is the modeling of surface interactions.

The reason for focus on gas-surface interactions lies primarily in the desire to solve engineering problems of the "satellite drag" type. Much of the remaining unknown contributions to spacecraft drag may be from poor knowledge of how molecules are adsorbed and re-emitted from real surfaces. This uncertainty presents a problem when attempting to measure atmospheric density by observing the motion of real spacecraft, or from instrumentation on-board the spacecraft. Attempts at measuring densities by authors such as Cook [16, 17], Jacchia [33, 34] and King-Hele [37, 39, 38] starting in the 1960s brought about the conclusion that error in drag coefficients led to errors

as high as 50% in the resulting densities. Poor atmospheric density models feed back to create poor spacecraft dynamics models, (as density is an essential parameter in numerical rarefied gas dynamics simulations) and hinder a number of space and earth atmospheric scientific pursuits.

This chapter identifies some of the most common GSI models and describes how they may be implemented in a DSMC code. Extended focus is given to the Cercingnani-Lampis (CL) model, as it has the most complete incorporation of accommodations and kernels of all the GSI models discussed here. The CL model's algorithmic implementation by R.G. Lord, known as CLL, is examined for correctness and a slight modification is applied to aid in its accuracy. Additionally, the Schamberg reflection kernel's implementation is shown to extend to general cases in which 2D analytical or numerical (i.e. obtained from experiments) kernels are given. Altogether, the clarifications and methods provided in this chapter intend to strengthen DSMC as a tool used for rarefied gas dynamics.

2.1 Gas-Surface Interaction Models

Surface interaction is defined as the way in which a particle impacts a solid boundary, is adsorbed, and is reflected. There are various differing models for determining reflected or post-impact quantities such as velocity distribution and temperature. Furthermore, the specific chemistry of the particle's interaction with the surface can greatly affect the prediction of post-impact behavior. Surface interaction can be separated into surface physics and reflection dynamics. Surface physics is primarily concerned with understanding and modeling if and how a molecule is adsorbed and re-emitted. Typically, energy and momentum accommodation modeling represent the bulk of theory in this half of surface interaction. The other half consists of velocity distribution or scattering and the methods of producing the proper results in the form of scattering kernels. It is often that a scattering kernel will include inputs from a surface physics model in the form of a thermal or momentum accommodation. The following sections summarize the practical forms of surface interaction as they may appear in numerical methods. An additional review by Moe and Moe [49] also presents some useful information, including experimental comparisons of drag

coefficients calculated using assumptions of different GSI models that supplements the following sections well.

2.1.1 Maxwellian Standard Reflection

Most DSMC methods employ at least a standard Maxwellian diffuse reflection model. In this model, first described by Maxwell[52, 46, 47] in the 1860s, molecules are assumed to be completely adsorbed and re-emitted at the surface temperature–a process referred to as full thermal accommodation. Reflection velocity distributions are therefore unaffected by incident distributions, but are affected by the necessity of the reflected particle to leave the surface in the half-plane defined by the surface outward normal. Modifications to this model include the addition of full or impartial thermal accommodation and/or partial (i.e. fractional) specularity. The Maxwellian equilibrium distribution function is given as

$$f_0 = \frac{\beta^3}{\pi^{3/2}} e^{-\beta^2 c'^2} \tag{2.1}$$

where c' is the molecular thermal speed and the reciprocal of the most probable thermal speed is

$$\beta = \frac{m_g}{2kT} \tag{2.2}$$

with k as Boltzmann's constant and m_g as the molecular mass of a molecule of the gas under consideration. The distribution of molecular speed in spherical polar coordinates, irrespective of direction is

$$f_{c'} = \frac{4\beta^3}{\pi^{1/2}} e^{-\beta^2 c'^2} \tag{2.3}$$

This is the distribution function of molecular thermal speed in a static gas at a temperature T, and thus does not provide the information required to select reflected velocity components. Some manipulation (as shown by Vincenti and Kruger [73] and Bird [3]) can reduce Equation 2.1 to the single component distribution or scattering kernel of

$$f_{u'} = \frac{\beta}{\pi^{1/2}} e^{-\beta^2 u'^2} \tag{2.4}$$

from which velocity components can be sampled if there are no restricting boundary conditions. In the case of a reflection from a surface, such a boundary condition does exist, however. Sampling

from Equation 2.4 will produce both negative and positive velocity component values. When considering the variable u' to be the velocity component normal to the surface and facing outward from it, using this equation as the distribution function would be unacceptable as it would allow reflecting molecules to penetrate the surface. Sampling from only the positive half of the distribution using a rejection technique is also incorrect, as the presence of the wall induces a preferred direction to the distribution that is ignored.

Equation 2.4 is a scaled symmetric Gaussian normal distribution function of the type

$$f_{u'_n} \propto u' e^{-\beta^2 u'^2} \tag{2.5}$$

where the scaling by u' necessarily enforces a singled-sided distribution. Bird states in Chapter 12 of Ref. [3] that to incorporate the effects of this preferred direction, a normalization over the range $u' \in [0..\infty]$, where u' is the normal to the surface, must be performed, thus obtaining the surface-outward-normal distribution

$$f_{u'_n} = 2\beta^2 u' e^{-\beta^2 u'^2} \tag{2.6}$$

Sampling from Equation 2.6 can be performed using a rejection method, but a faster, direct alternative is available in this particular case. The Box-Muller method [7] is a method of choosing two independent random variables, each of which are distributed according to a symmetric Gaussian distribution, from a set of independent uniform random variables. The two sets are related by the Box-Muller transformation

$$\begin{aligned} z_1 &= \sqrt{-2\ln(R_{f_2})}\cos(\theta) = \sqrt{2}\beta R\cos(\theta) \\ z_2 &= \sqrt{-2\ln(R_{f_2})}\sin(\theta) = \sqrt{2}\beta R\sin(\theta) \end{aligned} \tag{2.7}$$

where the definition $R^2 = -(1/\beta^2)\ln(R_{f_2})$ is made and $\theta = 2\pi R_{f_3}$ and R_{f_2} and R_{f_3} are random, independent, uniformly-distributed variables in the range (0..1]. The reason for this definition is that this is a transformation of the 2D uniformly-distributed random variable vector $\boldsymbol{R}_f = [R_{f_2}\, R_{f_3}]^{\mathrm{T}}$, which is interpreted as being polar, to the Cartesian Gaussian 2D vector or joint-distribution $\boldsymbol{z} = [z_1\, z_2]^{\mathrm{T}}$ where $\sqrt{2}\beta R$ is the radial coordinate. In 2D, the variable $s = z_1^2 + z_2^2$,

which is the sum of squares of two standard normal variables, will be distributed according to the χ^2–distribution, which is

$$f_{\chi^2}(s:k) = \frac{1}{2^{k/2}\Gamma(k/2)} s^{k/2-1} e^{-s/2} \tag{2.8}$$

In 2D, $k = 2$, and the equation simplifies to

$$f_{\chi^2}(s:2) = f_S(s) = \frac{1}{2} e^{-\frac{s^2}{2}} \tag{2.9}$$

which corresponds to the exponential distribution, scaled by 1/2. Thus, since the range of $2f_{\chi^2}$ is [0..1], the following definition can be made:

$$2f_{\chi^2} = R_{f_1} = e^{-\frac{s^2}{2}} \tag{2.10}$$

where R_{f_1} is another uniform random variable in the range (0..1]. Furthermore, is clear that, according to the transformation of Equation 2.7 that

$$s = z_1^2 + z_2^2 = (\sqrt{2}\beta R)^2 \cos^2(\theta) + (\sqrt{2}\beta R)^2 \sin^2(\theta) = 2\beta^2 R^2 \tag{2.11}$$

Substituting this definition of s into Equation 2.10 and inverting (which can only be done now that the distribution is in the form of a simple exponential function) yields the expression for R previously mentioned as a definition, but now as a function of a new uniformly-distributed random variable:

$$R = \pm \frac{1}{\beta} \sqrt{-\ln(R_{f_1})} \tag{2.12}$$

If the positive solution is taken, then $u'_n = R$ is a velocity component parallel to the outward normal of a surface which is only a function of a single, uniformly-distributed random variable R_{f_1}, which is easy to generate using most any random number generator. The two parallel components u'_{p1} and u'_{p2} should follow the symmetric normal distribution of Equation 2.4. Thus, they can be generated by using Equations 2.7 directly. The final, reflected velocity vector, in surface coordinates is then

$$\boldsymbol{V}_{ref,diff} = \begin{bmatrix} u'_n \\ u'_{p1} \\ u'_{p2} \end{bmatrix} = \begin{bmatrix} R \\ \frac{\sqrt{-\ln(R_{f_2})}}{\beta} \cos(\theta) \\ \frac{\sqrt{-\ln(R_{f_2})}}{\beta} \sin(\theta) \end{bmatrix} \tag{2.13}$$

This velocity can then be rotated into any other coordinate system in order to facilitate the requirement of determining inertial or computational velocity from a surface of any orientation. In 2D, this can mean assuming u'_n is already aligned with one of the inertial coordinate directions, such as the positive x-direction, determining the rotation angle from that direction to the target surface's normal direction, and performing the rotation on the reflected velocity as it is given in Equation 2.13.

All that remains is to prove that the distribution of R, $f_R(r)$, is identical to Equation 2.6. This is done by noting that $f_R(r)$ is the distribution of the sum of the squares of two normal distributions, s, the same as the distribution of $2\beta^2 R^2$, from the definition of s in Equation 2.11. Therefore, the distribution of the square root of s, scaled by $1/(2\beta^2)$, should give $f_R(r)$. The variable r is defined through the transformation

$$r = g(s) = \sqrt{\frac{s}{2\beta^2}} \tag{2.14}$$

which derives from the definition of s in Equation 2.11. The inverse transformation is

$$g^{-1}(r) = s = 2\beta^2 r^2 \tag{2.15}$$

or simply just a slight re-statement of Equation 2.11 itself. The first derivative is

$$g'(s) = \frac{dg}{ds} = \frac{\sqrt{2}}{4\beta\sqrt{s}} \quad \Rightarrow \quad g'(r) = \frac{1}{4\beta^2 r} \tag{2.16}$$

Transforming from the $f_S(s)$ distribution to the $f_R(r)$ distribution is then performed as

$$f_R(r) = \left|\frac{1}{g'(g^{-1}(r))}\right| f_s(g^{-1}(r)) = \left|\frac{1}{g'(s)}\right| f_S(s) = 2\beta^2 r e^{-\beta^2 r^2} \tag{2.17}$$

which is the same as Equation 2.6.

The distributions used for determining the diffuse reflected velocity are illustrated in Figure 2.1. The theoretical normalized functions are compared to a numerical generation of the components of velocity using Equation 2.13 for a large number of samples. The generated function $f_{c'}$ is the magnitude of the entire vector, while the theoretical $f_{c'}$ is the free-space (i.e. surface-independent) speed as given by Equation 2.3. This comparison makes it clear that the imposition

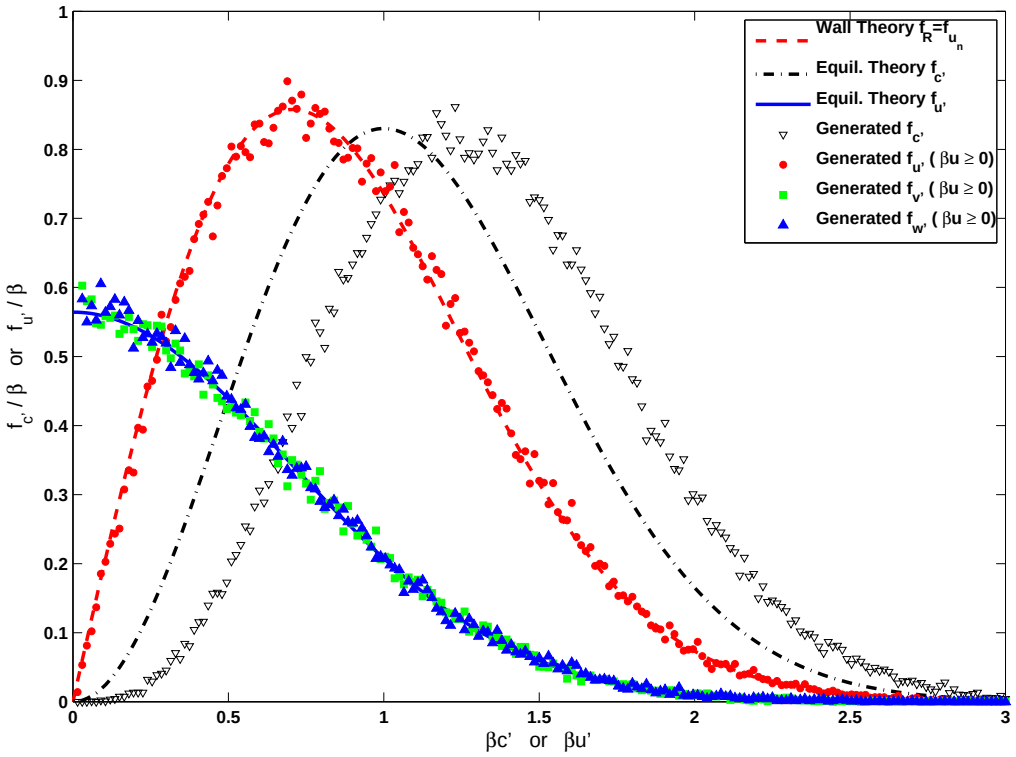


Figure 2.1: Comparison of the positive half of the domain for normalized theoretical diffuse wall velocity distribution functions with numerical generation of 100,000 samples. $f_{v'}$ and $f_{w'}$ are the two parallel components, corresponding to u_{p1} and u_{p2} from Equation 2.13.

of the surface's preferred direction not only pushes the entire distribution of the normal component ($f_{u'}$ in the figure) rightward, but skews the resulting generated speed by pushing it higher as well (i.e. the curve and the data should be different, as the figure indicates).

Although this procedure is included in Appendix C of Ref. [3], its relation to the Box-Muller transformation and other details and proofs are omitted. These details have been discussed here for the purpose of providing a more complete explanation of how to practically implement generation of a standard Maxwellian diffuse reflected velocity, which is a key element of basic DSMC. Furthermore, Maxwellian inflow velocity generation for stream boundaries is closely related to diffuse reflected velocity generation as both enforce the same condition of a modified equilibrium distribution that is affected by the presence of a preferred direction of effusion. The relation to

stream boundary conditions is also apparent, as discussed in Section 1.2.5. A more thorough and mathematically rigorous presentation of this family of subjects is presented by Garcia and Wagner [25].

2.1.2 Sentman and Thermal/Energy Accommodation

Perhaps the most useful extension to the standard diffuse model is the inclusion of influence from thermal accommodation. In a surface reflection, reflected molecules may be modeled as having different temperatures than the surface itself. It is then assumed that incident molecules "accommodate" to the surface to some degree and are re-emitted at a temperature that is not necessarily the surface temperature. The amount of this thermal adjustment is measured by the accommodation coefficient:

$$\alpha_n = \frac{T_{tr,i} - T_{tr,r}}{T_{tr,i} - T_s} \tag{2.18}$$

where T_s is the actual surface temperature and $T_{tr,i}$ is the translational or kinetic temperature of a molecule incident upon the surface, defined as

$$T_{tr,i} = \frac{m_g}{3k} \|\boldsymbol{V}_i\|^2 \tag{2.19}$$

where $\boldsymbol{V}_i$ is the molecular incident velocity. In this context, α_n is the thermal or *energy* accommodation, which assumes isotropic energy balance between the surface and the gas. Other accommodations may be defined, such as momentum accommodations, that are directional (e.g. the CL model, Section 2.1.4). The accommodation coefficient is often a given or known constant, thus $T_{tr,r}$, the reflected molecule temperature, can be defined as

$$T_{tr,r} = \frac{m_g}{3k} \|\boldsymbol{V}_i\|^2 (1 - \alpha_n) + \alpha_n T_s \tag{2.20}$$

Sentman [67] first proposed using α_n to determine the reflected velocity distribution, using $T_{tr,r}$ as the input, through β, to the standard Maxwellian reflection kernels of Equations 2.4 and 2.6. Of note is the use of only translational energy in the determination of α_n in Equation 2.18. Without consideration of additional internal modes (e.g. rotational) that would change the total molecular

energy, and thus temperature, of incident particles, the definition of Equation 2.19 remains unchanged. However, with gases comprised of multiple species and with species that are polyatomic, these definitions become more complicated. These complications are reserved in this discussion as it intends only to be a summary. Further details on this matter can be found in Chapters 1 and 5 of Ref. [3]. Determination of an appropriate value for α_n is the subject of separate theory. Thus, the only input to the Sentman model is α_n, with possible extensions of a specular/diffuse weighting.

2.1.3 Nocilla

A modification of the standard Maxwellian diffuse reflection model, which is applicable for free-molecular flow only, is the Nocilla model [53, 31], which creates a so-called "drifting" Maxwellian distribution function. In the standard diffuse model, the surface normal vector represents the direction upon which the distribution of velocities is centered when projected into the 2D incidence plane. This function is often termed a raised cosine distribution. The rotation of the centering vector from the surface normal direction to another direction dependent on the gas properties causes a preferred reflection direction, with a lobe centered about it. Nocilla's re-emission reflection distribution kernel is given as the distribution function

$$f_{\text{noc}}(\boldsymbol{c}') = \frac{\beta_r^3}{\pi^{3/2}} e^{-\beta_r^2(\boldsymbol{c}'-\boldsymbol{V}_r)^2} \tag{2.21}$$

where $\boldsymbol{c}'$ is the molecular thermal velocity and $\beta_r = \sqrt{m_g/(2kT_r)}$ is the reflected speed reciprocal–a function of T_r, which is the imposed reflection temperature. Therefore, the additional parameters are introduced are the drift velocity, $\boldsymbol{V}_r$ and T_r, which is considered to be distinct from the actual surface temperature T_s. Notable is the difficulty in determining the values of these parameters. It is mostly unclear what choice should be made for the drift velocity and much of the information that provides this, or its more easily-measured alternative, the speed ratio $s_r = \|\boldsymbol{V}_r\|/\sqrt{2kT_r/m_g}$, has come from experimental data. A review of relations between reflected and incident angles as well as values for speed ratio was performed by Collins and Knox [15] and will not be discussed here. The reflection temperature, however, can be determined from an imposition of a thermal

accommodation model as in all kernels where this parameter (or a similar variable) is present.

2.1.4 Cercignani-Lampis

It is often in the context of numerical simulation via the DSMC method that specific implementations of GSI models are of the most interest. Although the fully diffuse Maxwellian model has been most often used, the Cercignani-Lampis (CL) model [10, 9, 11] may represent the most rigorously-developed of alternatives to fully diffuse reflection, It is a scattering kernel with implicit inclusion of normal and tangential energy and momentum accommodation, and meets the requirements of reciprocity, positive definiteness and normality necessary to satisfy the Boltzmann equation.

Those seeking to validate the CL model and indeed, values of its parameters, with respect to real problems such as those arising from the desire to understand spacecraft flight aerodynamics (e.g. satellite drag) often turn to DSMC. A means of generating the necessary reflected velocities in code, in the form of an algorithm, such that methods such as DSMC may implement the CL model in simulation was developed by Lord[44]. This became known as the Cercignani-Lampis-Lord (CLL) GSI model. This algorithm is very quick and does not pose much implementation overhead beyond a fully diffuse GSI model. However, an oversight has existed thus far in the form of a lack of proper satisfaction of the necessary condition of surface isotropy. This section explains the CL kernel, the CLL algorithm's mistake, and proposes a simple adjustment that properly satisfies surface isotropy, as well as allows the CL kernel to function as was originally intended for cases of "back–scattering" tangential accommodation ($\sigma_t > 1$).

The CL scattering kernel can be expressed in a few ways. It is often given in what is perhaps its most physically meaningful form,

$$f_{CL}(u_i, u_r, \mathbf{V}_i, \mathbf{V}_r) = \frac{2\beta^4 u}{\pi \alpha_n \alpha_t} I_0(q) e^{-D_n \beta^2} e^{-D_t \beta^2} \tag{2.22}$$

where $\beta = \sqrt{m_g/(2kT_s)}$ is the reciprocal of the most probable molecular speed at a surface temperature of T_s. The other parameters are explained in Table 2.1. For brevity and notational clarity,

Table 2.1: Parameter names and descriptions for the CL kernel.

Symbol	Description
u_i	Component of incident velocity normal to the surface
u_r	Component of reflected velocity normal to the surface
v_i	Component of incident velocity along surface-tangential direction 1
v_r	Component of reflected velocity along surface-tangential direction 1
w_i	Component of incident velocity along surface-tangential direction 2
w_r	Component of reflected velocity along surface-tangential direction 2
$\mathbf{V}_i$	Vector sum of incident velocity in the surface plane, $\mathbf{V}_i = [v_i\ w_i]^T$
$\mathbf{V}_r$	Vector sum of incident velocity in the surface plane, $\mathbf{V}_r = [v_r\ w_r]^T$
α_n	Normal energy accommodation coefficient
α_t	Tangential energy accommodation coefficient, defined as $\alpha_t = \sigma_t(2 - \sigma_t)$

the variables q, D_n and D_t are defined as

$$q = \frac{2\beta^2\sqrt{1-\alpha_n}}{\alpha_n} u_i u_r \tag{2.23}$$

$$D_n = \frac{u_r^2 + (1-\alpha_n)u_i^2}{\alpha_n} \tag{2.24}$$

$$D_t = \frac{\|\mathbf{V}_r - (1-\sigma_t)\mathbf{V}_i\|^2}{\alpha_t} \tag{2.25}$$

The magnitude of vectors $\mathbf{V}_i$ and $\mathbf{V}_r$ are also the tangential components of the total incident and reflected velocity vectors, respectively, only in the 2D planes of incidence and reflection. This is to make the important clarification that the CL kernel describes a molecule's reflection from a surface entirely within the incidence and reflection planes. Each plane contains only a surface-normal and surface-tangential component of velocity and itself can be rotated azimuthally (about the surface normal). Equation 2.22 is a statement of this formulation. For an isotropic surface, this further implies that neither tangential direction is more significant than the other, meaning that component distributions for the surface-tangential directions $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$ must be equivalent. In fact, in Lord's original discussion, the CL kernel is also introduced this way, with the statement:

> *"... the scattering kernel therefore consists of a product of three terms, one for each component. For isotropic surfaces, however, the two tangential velocity components v and w must behave identically, so that two of the terms are exactly similar in form."* Ref. [44], Section 2.

Equation 2.22 can then be written as a decomposition of the three directions in the surface coordinate system as

$$f_{CL,n}(u_i, u_r) = \frac{2\beta^2 u}{\alpha_n} I_0(q) e^{-D_n \beta^2} \tag{2.26}$$

$$f_{CL,t_1}(v_i, v_r) = \frac{\beta}{\sqrt{\pi \alpha_t}} e^{-D_{t1} \beta^2} \tag{2.27}$$

$$f_{CL,t_2}(w_i, w_r) = \frac{\beta}{\sqrt{\pi \alpha_t}} e^{-D_{t2} \beta^2} \tag{2.28}$$

where I_0 is the modified Bessel function of the first kind. The definitions of the exponential arguments for the tangential components are

$$D_{t1} = \frac{[v_r - (1 - \sigma_t) v_i]^2}{\alpha_t} \tag{2.29}$$

$$D_{t2} = \frac{[w_r - (1 - \sigma_t) w_i]^2}{\alpha_t} \tag{2.30}$$

which naturally satisfy the joint-probability density distribution that is Equation 2.22 as the product:

$$f_{CL} = f_{CL,n} f_{CL,t_1} f_{CL,t_2} \tag{2.31}$$

Any algorithm that aims to generate the three reflected velocity components (u_r v_r w_r) should then, over a large number of samples for each component, generate distributions that match Equations 2.26 to 2.28. The CLL algorithm supposedly provides the method of sampling reflected components of velocity in three dimensions.

2.1.4.1 Corrections to the CLL Algorithm

Lord's original presentation of a generation algorithm for the CL kernel presents a velocity space diagram that interprets a reflection using the CL kernel[44]. This diagram is reproduced in Figure 2.2. Lord describes this geometry as being applicable to both the normal kernel and the tangential (that is, product of Equations 2.27 and 2.28) kernels. When it describes the normal kernel, Lord's definitions for the various points and distances allow the correct derivation of a generating algorithm for u_r. However, when the diagram is re-interpreted as describing the tangential

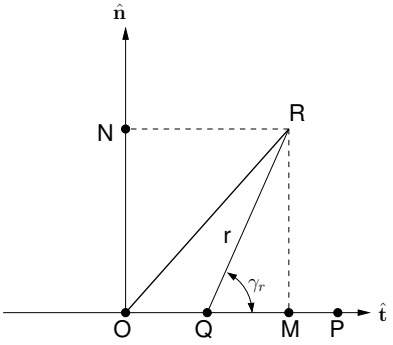


Figure 2.2: Lord's original geometric interpretation of the CL kernel.

reflection, Lord re-defines the horizontal axes as in-surface-plane direction $\hat{\mathbf{t}}_1$ and the vertical axis as in-surface-plane direction $\hat{\mathbf{t}}_2$, with the line segment $\overline{OR}$ as being equivalent to $V_r = \sqrt{v_r^2 + w_r^2}$ (the magnitude of the tangential velocity). Using the same definitions (with a change of variable from α_n to α_t) of

$$\overline{OQ} = \sqrt{1-\alpha_t}\ \ \overline{OP} \tag{2.32}$$

where $\overline{OP} = v_i$, Lord's algorithm follows:

$$\begin{aligned} r_1 &= \sqrt{-\alpha_n \ln(R_1)} \\ \gamma_{r_1} &= 2\pi R_2 \\ u_r &= \frac{1}{\beta}\left[r_1^2 + u_i^2\beta^2(1-\alpha_n) + 2r_1\left|u_i\beta\right|\sqrt{1-\alpha_n}\cos(\gamma_{r_1})\right]^{\frac{1}{2}} \end{aligned} \tag{2.33}$$

$$\begin{aligned} r_2 &= \sqrt{-\alpha_t \ln(R_3)} \\ \gamma_{r_2} &= 2\pi R_4 \\ v_r &= \frac{1}{\beta}\left[v_i\beta\sqrt{1-\alpha_t} + r_2\cos(\gamma_{r_2})\right] \end{aligned} \tag{2.34}$$

$$w_r = \frac{r_2\sin(\gamma_{r_2})}{\beta} \tag{2.35}$$

where R_i are random fractions taken from a uniform distribution in the range $[0, 1]$. What is obvious about this algorithm is that the two tangential components v_r and w_r have different definitions,

and thus, generate improperly restricted distributions. The aforementioned surface isotropy does not permit this asymmetry. What is meant by "symmetry" in this sense is that both v_r and w_r should be distributed similarly in the surface plane, where no azimuthal angle is more inherently probable than another, such that the definitions should not offset one more than the other. The generating equation for v_r clearly possesses the offset of $v_i\beta\sqrt{1-\alpha_t}$, where the equation for w_r does not. This implies that reflections always posses a bias towards direction $\hat{\mathbf{t}}_1$, or in other words, a reflection uses one in-surface-plane tangential incident component "more" than the other.

The reason for this discrepancy is the misinterpretation of the diagram in Figure 2.2. In short, the same diagram cannot be used to describe the tangential reflection. If the diagram is considered an illustration of velocity space for a reflection in the reflection plane, then the definitions given must appear as illustrated in Figure 2.3. In this diagram, the incidence and reflection planes sit

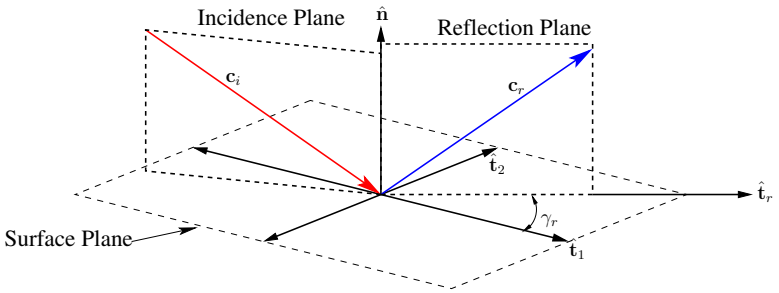


Figure 2.3: Locations and definitions of a reflection. Relevant points are reproduced along the line segment that is collinear with the reflection plane in-surface direction $\hat{\mathbf{t}}_r$.

"above" the surface plane as illustrated. The reflection plane is oriented at an arbitrary azimuth angle γ_r, which must be a random variable uniformly distributed between 0 and 2π if the surface is isotropic. The two tangential directions in the surface plane are shown as well as the directions $\hat{\mathbf{t}}_i$ and $\hat{\mathbf{t}}_r$, which are the directions parallel with the incident and reflection planes, respectively. The reflected velocity vector $\mathbf{c}_r$ is projected into the surface plane, creating $\mathbf{V}_r$, which lies along direction $\hat{\mathbf{t}}_r$, and being further decomposed into components v_r and w_r by projections along directions $\hat{\mathbf{t}}_1$ and $\hat{\mathbf{t}}_2$, respectively. A similar description can be made for incident velocity vector $\mathbf{c}_i$.

What is important to note is that the CL model decomposes the reflected velocity into a

contribution from a diffuse scattering model (which shall be termed the *independent* component) and a contribution from the incident total velocity (termed the *dependent* component). Thus, the statement that γ_r must be uniformly distributed amongst all possible directions in the surface plane for an isotropic surface specifically applies to how the independent component is defined. The CL kernels naturally introduce (by superposition) the dependent components which create a preferred azimuthal direction that is a function of the incident azimuth (or in this case, directly from the individual components v_i and w_i).

From these considerations and this geometry, a new tangential reflection diagram can be made by examining the surface plane. Figure 2.4 shows one quadrant of the surface plane where the distance $\overline{QP} = V_{r1}$, which is the dependent component of the in-plane tangential speed. The offset distance $\overline{OQ}$ is shown as being the magnitude of the variable $r = \overline{OQ}$, which, as before, is distributed according to the half-plane Maxwellian diffuse kernel

$$f(r) = \frac{2\beta^2 r}{\alpha_t} e^{\frac{r^2}{\alpha_t}} \tag{2.36}$$

This is the typical kernel used to describe scattering of the normal component of velocity off of a standard fully diffuse surface with full accommodation. Sampling two orthogonal velocity components from this distribution can be performed with the Box-Muller [7] transform as usual. In this case, this leads to the definitions

$$\begin{aligned} V_r &= \frac{r}{\beta} + V_{r1} \\ v_{r0} &= \frac{r}{\beta}\cos(\gamma_r) \\ w_{r0} &= \frac{r}{\beta}\sin(\gamma_r) \\ v_{r1} &= V_{r1}\cos(\gamma_r) = \sqrt{1-\alpha_t}v_i \\ w_{r1} &= V_{r1}\sin(\gamma_r) = \sqrt{1-\alpha_t}w_i \end{aligned} \tag{2.37}$$

The CL kernel creates a type of shifted Maxwellian where the offset or shift velocity is given by a scaling of the incident velocity by the factor $\sqrt{1-\alpha}$, where α is either α_n for the normal component or $\alpha_t = \sigma_t(2-\sigma_t)$ for the reflection plane tangential component. This means that the line segment

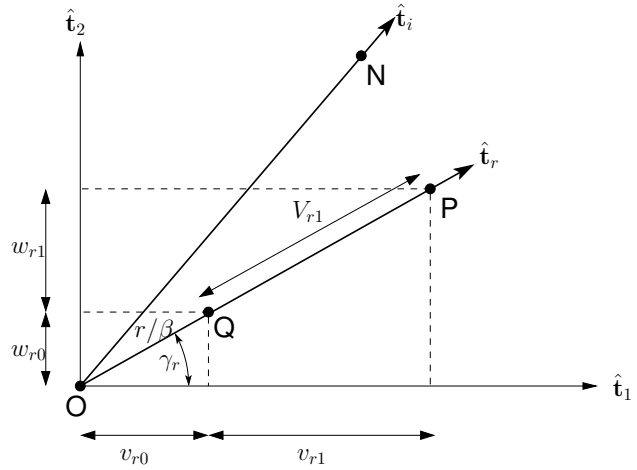


Figure 2.4: Correction to Lord's tangential (in-plane) reflection diagram (velocity space).

$\overline{ON} = V_i$, where V_i is the in-surface-plane incident tangential speed, and thus $V_{r1} = \sqrt{1-\alpha_t}V_i$ and similarly, component-wise: $v_{r1} = \sqrt{1-\alpha_t}v_i$, $w_{r1} = \sqrt{1-\alpha_t}w_i$. This formulation leaves the CL kernel to be in-plane direction-independent, as it must be for a tangentially isotropic surface, preferring only to specify how the scattering occurs of the summed component V_r along the direction $\hat{\mathbf{t}}_r$[1] . From this definition of V_{r1}, the visible geometry, and Equations 2.37, the reflected velocity components in the surface plane are given as

$$v_r = v_0 + v_1 = \frac{r}{\beta}\cos(\gamma_r) + v_i\sqrt{1-\alpha_t} \tag{2.38}$$

$$w_r = w_0 + w_1 = \frac{r}{\beta}\sin(\gamma_r) + w_i\sqrt{1-\alpha_t} \tag{2.39}$$

where once again, because of the uniform distribution of these components angularly in the surface plane, variables r and γ_r are random variables that are generated separately from those given in Equation 2.33, renamed r_2 and γ_{r_2}, as shown in Equations 2.40.

$$\begin{aligned} r_2 &= \sqrt{-\alpha_t \ln(R_3)} \\ \gamma_{r_2} &= 2\pi R_4 \end{aligned} \tag{2.40}$$

There is still yet a problem with this formulation. The tangential momentum accommodation

[1] An anisotropic model would require tangential momentum accommodation coefficients to be defined for each surface direction and thus, would not allow a simple scaling relationship between speeds V_{r1} and V_i.

coefficient σ_t is one of the CL kernel's tunable parameters, which gives a measure of how tangential momentum is imparted to reflecting particles. It is permitted to be in the range $0 < \sigma_t < 2$. However, for values of $\sigma_t > 1$, a reflection should have a reflected in-plane velocity $\mathbf{V}_r$ that is opposite in sign to $\mathbf{V}_i$. Figure 2.5 illustrates this problem, where it can be seen that the magnitude of the vector $\mathbf{V}_r$ is V_r and is unchanged regardless of σ_t, with only a sign change occurring. Equations 2.38 and 2.39 will fail to properly capture this phenomenon. For the case of $\sigma_t = 1$, the distributions f_{CL,t_1} and f_{CL,t_2} are equivalent for all $\mathbf{c}_i$. It is perhaps for this reason that the appropriate handling of tangential momentum has been overlooked thus far.

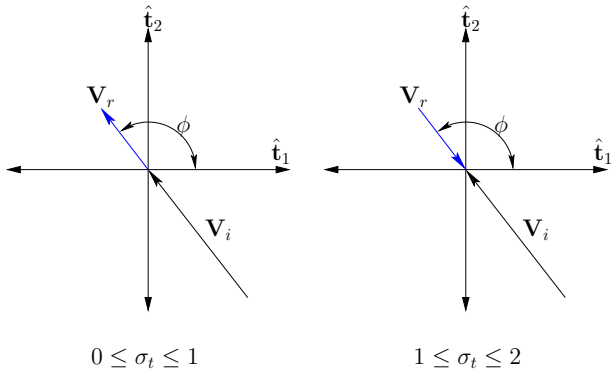


Figure 2.5: An illustration of the action of the value of σ_t on the reflected in-surface-plane tangential velocity.

This problem is easily remedied by realizing that, according to the geometry of Figures 2.3 and 2.4, the scaling of V_i should have two solutions, i.e. $V_{r1} = \pm\sqrt{1-\alpha_t}V_i$, as these components are not restricted to the positive half-space of the surface system (as is the case for the normal component[2]). The final versions of the generating equations for components v_r and w_r are then

$$v_r = \begin{cases} \frac{r_2}{\beta}\cos(\gamma_{r_2}) + v_i\sqrt{1-\alpha_t} & \text{for } 0 < \sigma_t \leq 1 \\ \frac{r_2}{\beta}\cos(\gamma_{r_2}) - v_i\sqrt{1-\alpha_t} & \text{for } 1 \leq \sigma_t < 2 \end{cases} \tag{2.41}$$

[2] Naturally, the absolute value seen in the formulations of Equation 2.34 as given by Padilla and Boyd[55] cannot be correct either.

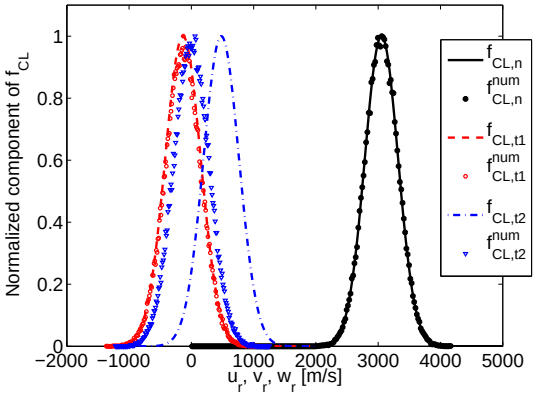
$$w_r = \begin{cases} \frac{r_2}{\beta}\sin(\gamma_{r_2}) + w_i\sqrt{1-\alpha_t} & \text{for } 0 \leq \sigma_t \leq 1 \\ \frac{r_2}{\beta}\sin(\gamma_{r_2}) - w_i\sqrt{1-\alpha_t} & \text{for } 1 \leq \sigma_t \leq 2 \end{cases} \tag{2.42}$$

The normal component equation is unchanged from Equation 2.33.

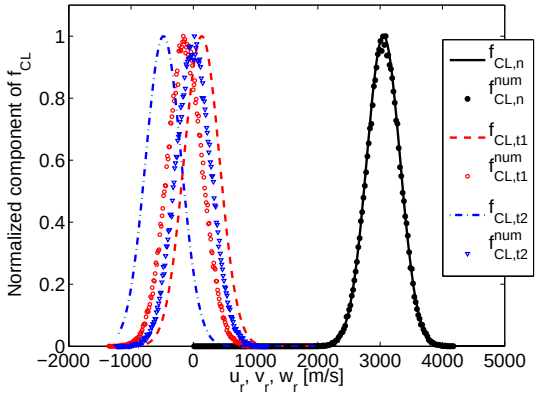
2.1.4.2 Numerical Results

To visualize the difference the corrected tangential component generation equations make, a direct evaluation and plotting of the individual components of the CL kernel (Equations 2.26 to 2.28) over a suitable range of possible values of reflected components of velocity was first made. A gas of mass $m_g = 46.5 \times 10^{-27}$ kg with a surface temperature of $T_s = 300$K was chosen. The generating equations for each component for the original formulation (Equations 2.33 to 2.35) were then used to generate 100,000 samples of each component in 200 bins. The same procedure was then performed for the corrected version of the algorithm. For all functions, a constant value of $\mathbf{c}_i$ was chosen as $\mathbf{c}_i = [u_i \ v_i \ w_i]^T = [-8600 \ -450 \ 1600]^T$ m/s. This comparison is shown in Figure 2.6.

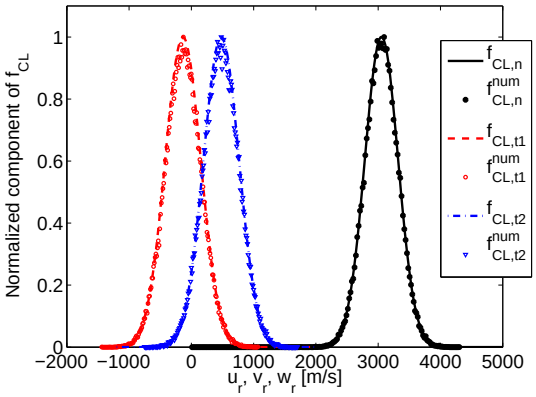
The tangential components' numerical distributions (i.e. those that were generated) can be seen to fail to correspond to their target distributions for the original algorithm (Figures 2.6a and 2.6b). For the case of $\sigma_t < 1$ (Figure 2.6a), the distributions of v_r appear to match, yet those of w_r do not, thus elucidating the aforementioned lack of appropriate isotropy. Tangential generated distributions for the corrected algorithm, however, appear to follow their evaluated distributions for both cases of σ_t (Figures 2.6c and 2.6d).



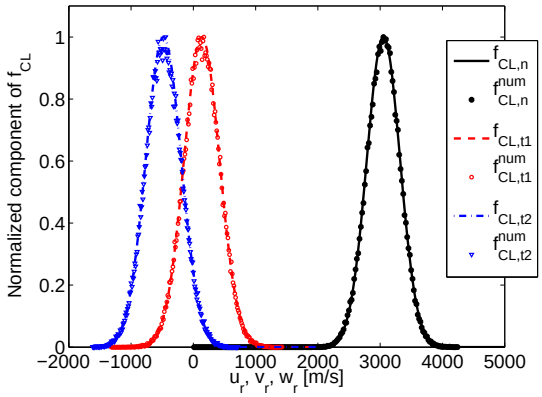
(a) Lord's original algorithm with $\sigma_t < 1$ ($\sigma_t = 0.7$)



(b) Lord's original algorithm with $\sigma_t > 1$ ($\sigma_t = 1.3$)



(c) Corrected algorithm with $\sigma_t < 1$ ($\sigma_t = 0.7$)



(d) Corrected algorithm with $\sigma_t > 1$ ($\sigma_t = 1.3$)

Figure 2.6: Comparison of normalized distributions of a direct evaluation of the CL kernel with Lord's original generation algorithm and the corrected version for $\alpha_n = 0.8$, $\mathbf{c}_i = [u_i \; v_i \; w_i]^T = [-8600 \; -450 \; 1600]^T$ m/s. The $^{\text{num}}$ superscript indicates numerically generated distributions, which are plotted with line markers only.

2.1.4.3 DSMC Comparisons

There are few examples of the CLL algorithm being implemented in numerical simulations. This analysis is performed with the understanding that GSI models such as CL are to be implemented in the DSMC method. A comparison with an investigation by Padilla & Boyd [55] and experimental results (also used by Padilla & Boyd) by Cecil & McDaniel [8] was made in order to place the improvements in the CL algorithm in the appropriate context. The problem of interest is a flat plate in a hypersonic flow (created by placing a small fused quartz plate model in the zone of a silence of an expanding free jet in Ref. [8]) with geometry and conditions given by Figure 2.7 and Table 2.2 (flow enters from the left). The center-line quantities in Table 2.2 refer to the fact that the experiment was constructed such that the plate's surface was collinear with the center-line of an axisymmetric free jet. The Knudsen number is based on the plate length of $\ell = 20$mm. The plate's exposed surfaces were modeled with CL (either the original or newly corrected versions). A small region of specular surface was specified along the plate in the lead-in region, from $x = 0$ to $x = x_0$. The inflow profiles $\boldsymbol{V}(y)$, $T(y)$, $n(y)$ were obtained as data from the experiment of Ref. [8] and extend from $y = 2$mm (the plate's surface) to $y = y_{in}$[3] . Inflow properties above y_{in} were assumed constant and equal to the values of the given data at y_{in}. The top stream boundary condition was also assumed to be flow under these conditions. Conditions were chosen to best model the experiment of Cecil & McDaniel as well as the computation by Padilla & Boyd such as to make an appropriate comparison.

Voldipar was used to perform a simulation time from zero to 7.5×10^{-4}s with an assumed steady state time of 3.75×10^{-4}s was used. The particle weighting was 4×10^{12} real particles per simulation particle. The domain was discretized into 140 uniformly spaced rectangular collision/sample cells in each direction. Rotational modes were modeled with two degrees of rotational freedom with a rotational relaxation of $Z_{rot} = 18.1$ and characteristic temperature of $T^* = 91.5$K.

Padilla & Boyd determined that, of the four values used, a tangential accommodation of

[3] The velocity profile is given as a plot by Padilla & Boyd in Ref. [55]. Temperature and number density profiles are not listed, however. All three profiles were obtained via private communication with the authors of Ref. [55]

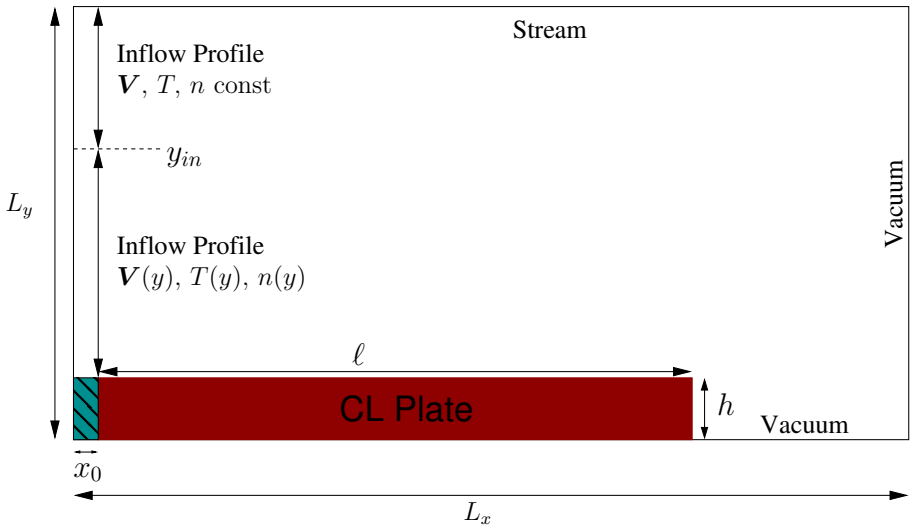


Figure 2.7: DSMC domain geometry and boundary conditions for hypersonic flow over a flat plate. Flow enters from the left.

$\sigma_t = 1.00$ fit the data of Cecil & McDaniel best for $V_x(y)$ and $\sigma_t = 0.75$ fit best for $V_y(y)$, under the same conditions, at the same location along the plate. The objective of this comparison was not to determine what the accommodation coefficients are to properly model hypersonic flow over a flat plate under these conditions, but rather to compare the effects of the change in implementation of the CL model as an algorithm. Thus, the aforementioned values of α_n and σ_t were chosen for the current comparison. Figures 2.8 and 2.9 give these results[4] .

The implementation of CL by Padilla & Boyd is the original (uncorrected) version. There appear to be no significant differences between the corrected and uncorrected versions in Voldipar, or between either of these results and Padilla & Boyd's computation (which used the MONACO [19] code). The most likely reason for this lack of difference between corrected and uncorrected algorithms is the fact that, in 2D, the reflected velocity component along surface tangential direction two, w_r, does not contribute to a change in the planar components u_r and v_r. It is the generation of w_r that has received the most obvious correction, with the form of v_r being essentially unchanged. In 2D, surface direction $\hat{\boldsymbol{t}}_2$ is aligned with computational direction $\hat{\boldsymbol{z}}$, velocities of which are not

[4] Data were obtained via digitization of plots in Ref. [55] and in Ref [8], the latter of which did not include error bars.

Table 2.2: DSMC simulation parameter values for hypersonic flow of molecular nitrogen (N_2) over a flat plate. Mach and Knudsen numbers are center-line ($y = h$) values at the inlet.

Variable	**Symbol**	**Value**	**Unit**
Mach number	Ma	11.9	
Knudsen number	Kn	9.5128×10^{-3}	
Domain width	L_x	27.0	mm
Domain height	L_y	14.0	mm
Plate surface temperature	T_S	300	K
Plate length	ℓ	20	mm
Plate height	h	2.0	mm
Inflow region limit	y_{in}	9.5	mm
Lead-in distance	x_0	2.0	mm
Gas molecular mass	m_g	46.5×10^{-27}	kg
Gas diameter	d_g	4.17×10^{-10}	m
Gas ratio of specific heats	γ	7/5	
Gas constant	R	296.8	kJ/(kg–K)

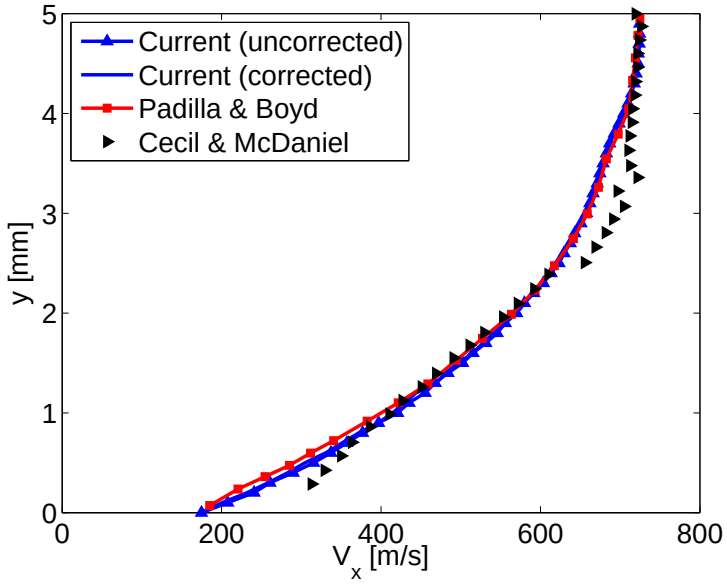


Figure 2.8: Comparison of boundary layer velocity profiles for component V_x at a distance along the plate of 10mm with $\alpha_n = 1.00$, $\sigma_t = 1.00$.

considered in 2D studies such as those of Ref.s[55] and [8]. Note that there is no difference between the performance of the uncorrected and corrected algorithm in the distribution of v_r between Figure 2.6a and Figure 2.6c. Although this is only true when $\sigma_t = 1$. When $\sigma_t = 1$ and $\alpha_t = 1$, no

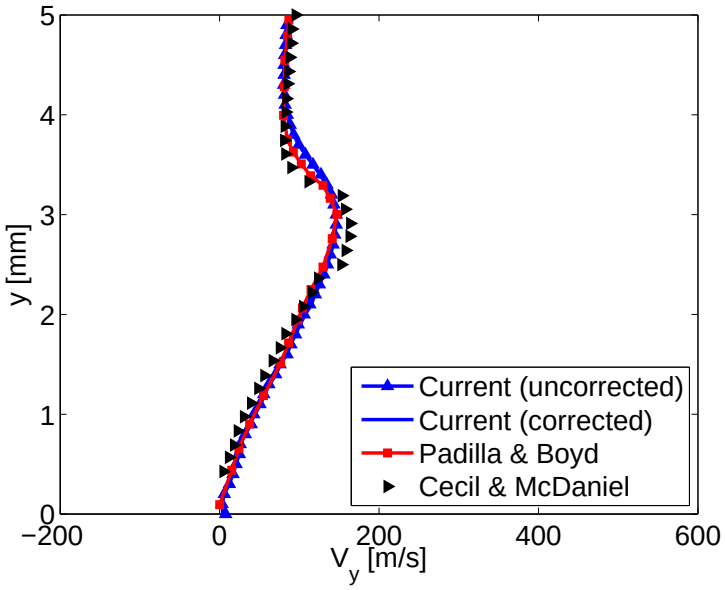


Figure 2.9: Comparison of boundary layer velocity profiles for component V_y at a distance along the plate of 10mm with $\alpha_n = 1.00$, $\sigma_t = 0.75$.

contribution from incident velocity is made to reflecting molecules. This is the fully-diffuse limit of the CL model. If more forward tangential momentum is allowed to pass through the surface reflection process (such as when $\sigma_t < 1$, meaning more specularity), the terms containing v_i and w_i in Equations 2.41 and 2.42 will affect the total reflecting velocity. If the computation was fully 3D, or a different definition of surface coordinate systems were employed, this problem would be assumed to exhibit a difference in boundary layer profiles for $\sigma_t \leq 1$. In 2D, a difference in the corrected generating equations and the original algorithm may be seen by performing the plate simulation with the two surface plane tangential directions swapped. Figure 2.10 shows the result of this computation for $\alpha_n = 1.00$, $\sigma_t = 0.25$. The corrected profile is skewed forward as a low tangential momentum accommodation allows more of the incident tangential momentum to push the profile forward. The uncorrected profile appears indifferent to this change in σ_t.

Additionally, an immediate difference can be observed in profiles when $\sigma_t > 1$, as both tangential velocity distributions are different between uncorrected and corrected versions even without regard to tangent direction definition. Figures 2.11 and 2.12 give these comparisons for a value of

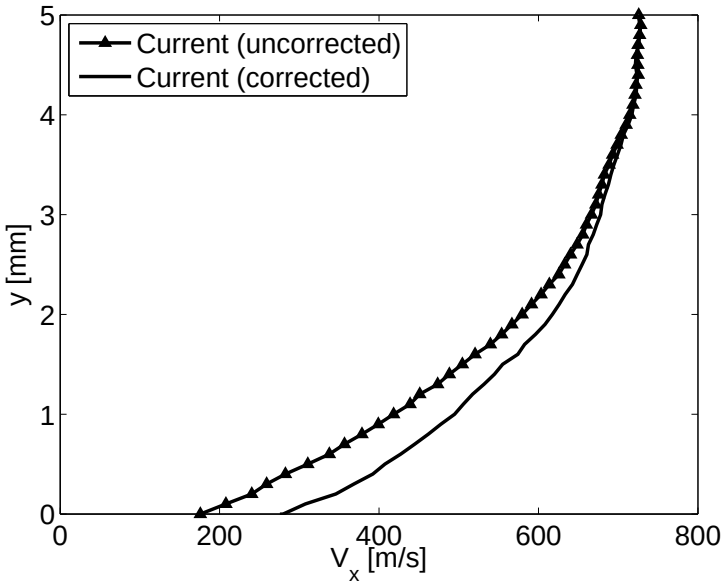


Figure 2.10: Comparison of boundary layer velocity profiles for component V_x at a distance along the plate of 10mm with $\alpha_n = 1.00$, $\sigma_t = 0.25$.

$\sigma_t = 1.50$. No relevant experimental data or DSMC computations were available in this case, and thus the comparison is between Voldipar DSMC implementations only.

This back-scattering value of tangential momentum accommodation contributes to much lower wall slip velocity than in the forward-scattering cases. Both profiles in the uncorrected results appear closer to those at much lower values of σ_t (such as $\sigma_t = 0.5$ in Ref.s[55] and [8]). The corrected results exhibit a quality closer to their counterparts for $\sigma_t = 1$ and $\sigma_t = 0.75$ with adjustments one might expect for a surface that is reluctant to permit as much stream-wise scattering of reflecting molecules.

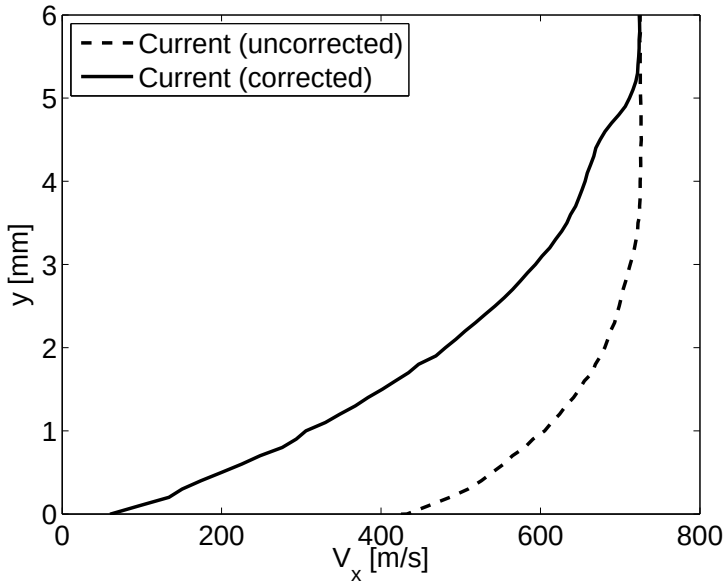


Figure 2.11: Comparison of boundary layer velocity profiles for component V_x at a distance along the plate of 10mm with $\alpha_n = 1.00$, $\sigma_t = 1.50$.

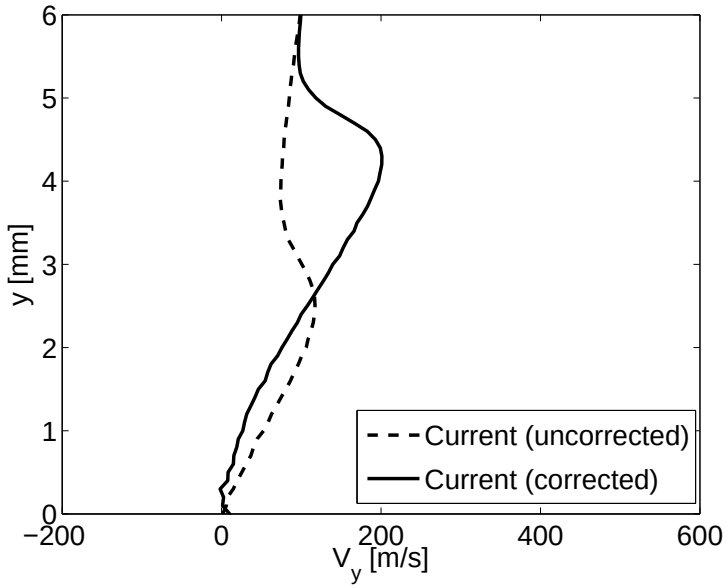


Figure 2.12: Comparison of boundary layer velocity profiles for component V_y at a distance along the plate of 10mm with $\alpha_n = 1.00$, $\sigma_t = 1.50$.

2.1.4.4 Conclusions of CL Adjustment

Having determined that the generation of reflected thermal velocity components does not satisfy the necessary condition of isotropy with respect to the surface tangential directions, a minor

re-derivation of the algorithm was made in order to ensure said isotropy is maintained. Additionally, a slight oversight that caused the original algorithm to function incorrectly for tangential momentum accommodation coefficients greater than unity was addressed.

These improvements were shown to work for all conditions by comparing analytically–evaluated CL distributions with numerically generated distributions. A model problem involving hypersonic flow of a flat plate was then used to test the implementation of the corrected CL algorithm in DSMC. The results of these computations, when compared with computations of the same problem available in the literature as well as with a published experiment showed that there was little change in boundary layer profiles when $\sigma_t < 1$, but significant change for $\sigma_t > 1$. It is likely that the nature of the computations, being 2D, prevented the former cases from exhibiting a difference between original and corrected CL. A 3D computation in which all three components of velocity within the boundary layer are compared would likely show these expected differences.

This analysis does not attempt to validate the CL model itself, nor choose appropriate accommodation coefficients for it for any problem. It instead is intended to raise the question of correctness in the application of gas-surface interaction models when applied in a simulation context such as DSMC.

2.1.5 Schamberg

A typical extension of the standard diffuse scattering kernel includes a specular component in addition to the standard diffuse component. Although this model is often termed the Maxwellian model, other models incorporate some degree of specularity which still depend partially on Maxwellian diffuse scattering or something similar. The CL model can recover a fully diffuse reflection when $\sigma_t = \alpha_n = 1$, and naturally introduces some specularity when those parameters are less than unity. A kernel introduced by Schamberg [63, 64] attempts to take this concept further by not only being quasi-specular but also defining a region of probable reflection centered on an incidence plane lobe angle θ_c as shown in Figure 2.13. The kernel assumes that all incident molecules enter at the same speed and all re-emitted or reflected molecules leave the surface at

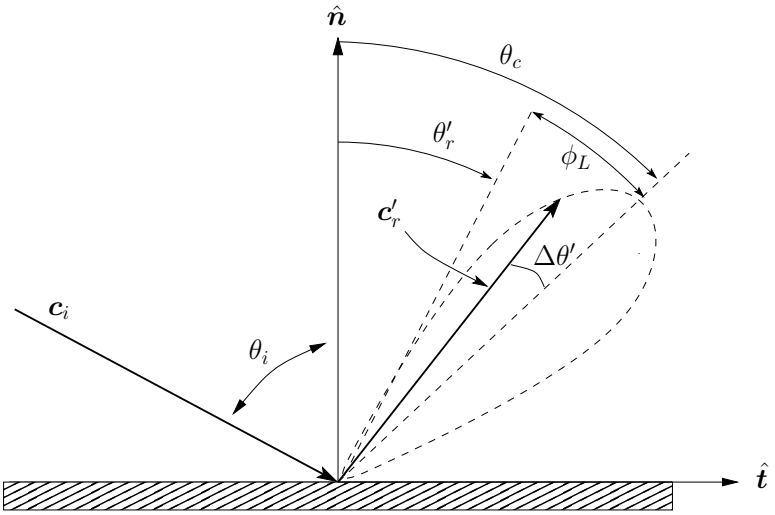


Figure 2.13: Schamberg wedge angular scattering kernel shown in the incidence plane. The reflection lobe represents the probability of a molecule leaving the surface at the angle θ'_r.

another constant speed.

There are some ambiguities regarding this model that stem from Schamberg's original presentation in Ref. [63]. The first is that the model is often said to be limited to free-molecular flow as it depends on individual molecules' interactions being independent from others, however there is no argument that properly supports this claim. Schamberg makes the remark:

> *"... it seems reasonable to require that $\theta_r \geq \theta_i$ since there is a general tendency for the surface interaction towards obliterating the effect of past history of the molecules, i.e., a trend towards diffuse reflection."* Ref. [63].

which may have been interpreted as implying the model is only free-molecular[5] when its implication seems to be that reflections appear to have a degree of independence from the local gas environment. The next ambiguity is in the interpretation of Schamberg's original description of the geometry of reflection. The direction of reflection is selected from an angular kernel given by a cosine distribution. However, the parameters of the kernel may have different meanings depending on whether the reflection is assumed to be truly 3D (cone) or if it is a 2D incidence plane-only reflection that must be given an azimuth angle (wedge). Schamberg explains that molecules are *"reflected in*

[5] Note that Schamberg's notation uses θ_r as the central beam angle, and ϕ_0 as the half-width. These are denoted θ_c and ϕ_L in this treatment, respectively.

a conical or wedge-shaped beam having a half-angular width denoted by ϕ_0". Much of the Schamberg model seems to have been formed by choosing some mathematical treatment to fit observations and thus suffers from a lack of rigor (Imbro et. al refer to it as "artificial"[32]). Nevertheless, it is a well-known model that serves to elucidate some of the challenge of creating and implementing GSI models at the engineering level. It seems reasonable to provide implementations for the Schamberg model in both its possible interpretations: Cone and Wedge.

2.1.5.1 Schamberg Wedge

In the wedge formulation, the reflection kernel should be considered to describe directions within the incidence plane only. The reflection plane may naturally be different from the incidence plane. The wedge version assumes that the angular kernel is the result of a projection of the true reflection direction into the incidence plane, illustrated in Figure 2.13 [6] . The Schamberg kernel is

$$f(\theta'_r) = \begin{cases} 1 + \cos\left(\frac{\pi}{\phi_L}\Delta\theta'\right) & \text{if } -\phi_L \leq \Delta\theta' \leq \phi_L \\ 0 & \text{otherwise} \end{cases} \tag{2.43}$$

When this kernel is applied in the Wedge formulation, $\Delta\theta' = \theta_c - \theta'_r$ is the angular distance of the reflected velocity vector from the most probable direction, or center of the lobe, θ_c. The actual direction of a reflected molecule in the incidence plane is θ'_r, which must be within the lobe of half-width ϕ_L centered on θ_c, measured from the surface normal direction. The vector $\boldsymbol{c}'_r$ is the projection of the actual reflected velocity onto the incidence plane. The unknown quantity in Equation 2.43 is the central reflection angle (lobe center angle) θ_c. Schamberg provides an average relation between this angle and the incident angle which is

$$\sin(\theta_c) = \sin^{\nu}(\theta_i) \quad , \quad \nu \geq 1 \tag{2.44}$$

where ν is a parameter that controls the weighting of specular and diffuse components. The parameter ν adjusts the degree of specularity, with $\nu = 1$ corresponding to purely specular where

[6] One reason for this is that traditional laboratory experiments may use a "2D detector", i.e., a detector that only detects impacts from molecules in a small slice of the azimuthal reflection space, thereby giving only a planar picture of the molecules' reflected distributions.

$\theta_i = \theta_c$ and $\nu \to \infty$ corresponding to fully diffuse, where $\theta_c = 0$, independent of θ_i. Necessary parameters for implementing Schamberg reflection are ν, ϕ_L and the effective reflected surface temperature, the latter of which can be determined from accommodation models and is used to select a reflected speed c_r. The choice of speed is not obvious, however. The method recommended by Moe [49] assumes the same speed for all reflections on a given type of surface. This choice, which assumes a constant reflection speed with only the direction changing, is the classical definition of the Schamberg model described by Schamberg [63] and Imbro et al. [32]. The reflected speed is given by

$$c_r = c_i \sqrt{1 + \alpha \left(\frac{T_s}{T_{k,i}} - 1 \right)} \tag{2.45}$$

where T_s is the surface temperature, $T_{k,i}$ is the kinetic temperature of the incident molecule and the thermal accommodation α can be either a constant or determined via another method such as the Goodman model [27, 26].

The Schamberg wedge model gives an angular scattering in the incidence plane only. However, unlike the CL kernel, there is no legitimate rule for projection of the tangential component $\boldsymbol{V}_r$ into the surface plane. This is because there is no inherent dependence on accommodation or of individual surface-plane incident velocity components. The Schamberg kernel, Equation 2.43, describes reflection in terms of a projected angle only. If the reflection plane is said to be arbitrarily oriented azimuthally, as was allowed for the CL analysis, the possibility exists for a molecule to reflect in a direction that is on the same side of the surface normal in the incidence plane. This event corresponds roughly to the cases where $\sigma_t > 1$ in the CL model (Figure 2.5) but is given by the diffuse limit ($\nu \to \infty$, $\phi_L = \pi$) of the Schamberg angle relation of Equation 2.44. That is, the reflections of the Schamberg kernel may only permit back-scattering when the lobe width and central lobe angle are such that part of the reflection lobe exists on the back (left) side of the surface normal, in the incidence plane. Although the surface should be considered isotropic in the tangential directions, choosing an azimuth angle at random from a uniform distribution over the full range of $[0..2\pi]$ without consideration for how the incidence plane reflected angle distribution

is affected will fail to produce the Schamberg scattering kernel.

The way in which the azimuth angle is chosen for the CL algorithm is such that the angle represents the component of scattering from a diffuse kernel only, where the contribution from the incident tangential velocity is then superimposed to, in effect, "push" the reflected velocity in a direction that is preferred based on the model's accommodation parameters. The Schamberg angular kernel has no such means to do this as it is only defined for a single planar reflection. Thus, it is necessary that any generation procedure chooses an azimuth that also satisfies the planar kernel. A discussion of this problem naturally does not assume that the incidence and reflection planes are coplanar, as this is rarely physically true for non-purely specular reflections. The geometry of a generalized reflection is shown in Figure 2.14, where primed variables denote incidence plane projections of actual reflected variables. The projection of the true reflected velocity $\boldsymbol{c}_r$ onto the incidence plane is shown as a dashed line segment. The planes each contain the surface normal but the reflected plane is rotated by inter-azimuth angle γ_{ir} from the incidence plane.

A reflecting molecule must be allowed to leave the surface at velocity $\boldsymbol{c}_r$, where $\boldsymbol{c}_r$ is not necessarily in the incidence plane. However, the Schamberg kernel must still be satisfied. The choice of the components of $\boldsymbol{c}_r$ must be made such that $\boldsymbol{c}_r'$ (the projection of $\boldsymbol{c}_r$ onto the incidence plane) forms an angle with the surface normal (deemed θ_r') which is distributed according to Equation 2.43, rather than the actual reflection cone angle θ_r.

What is clear is that the reflected azimuth angle γ_r must be chosen prior to enforcing a relation between it and θ_r and θ_r'. The relation of of γ_r to the other azimuth angles is shown in Figure 2.14 where the surface tangential reflected velocity $\boldsymbol{V}_r$ is shown projecting onto the incident tangential plane and the inter-azimuth angle γ_{ir} is defined as being positive from the incidence direction $\hat{\boldsymbol{t}}_i$. If this relation is established, then it is acceptable to select γ_r from a uniform distribution in an appropriate range. The relation between θ_r and θ_r' is established by first noting that both the incidence and reflection planes contain the surface normal. This implies that the normal component

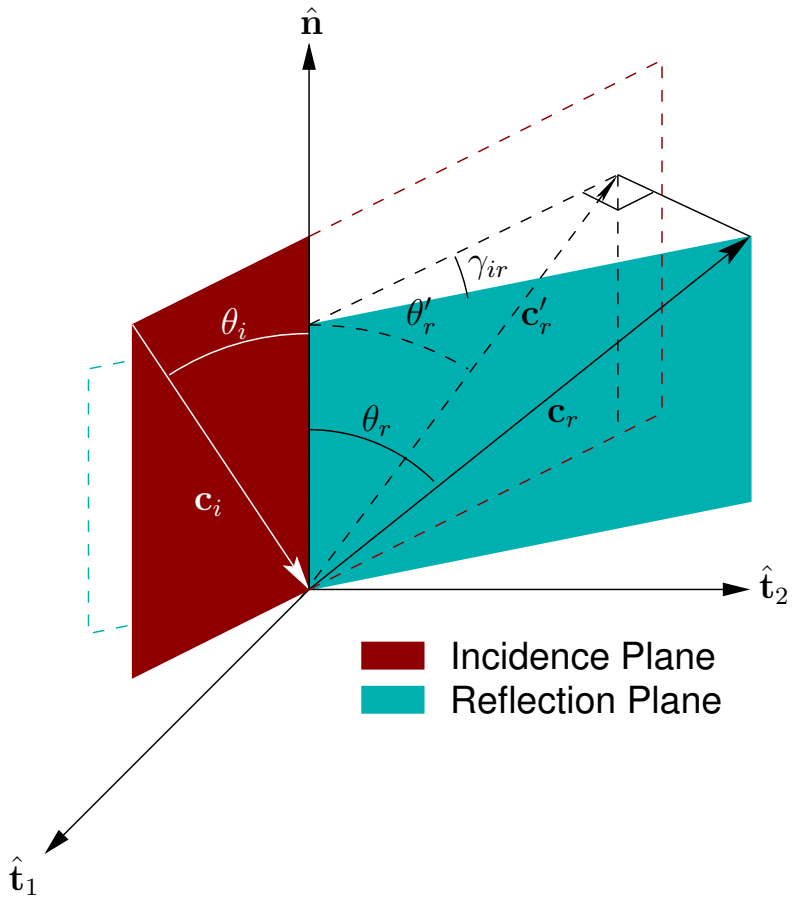


Figure 2.14: An illustration and definitions of a general surface reflection that features the incidence and reflection planes.

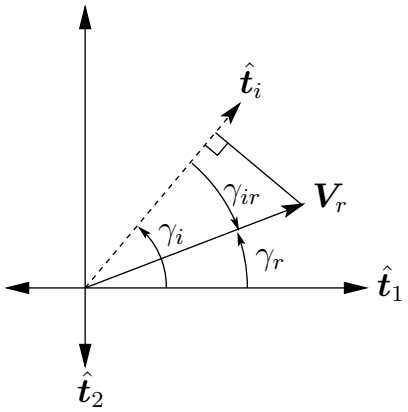


Figure 2.15: A top-down view of Figure 2.14 showing the surface plane in which tangential velocity components are projected and giving the definitions of relevant azimuth angles.

of reflected velocity in both planes must be identical and is given by

$$u_r = u'_r = c_r \cos(\theta_r) \tag{2.46}$$

where c_r is the magnitude of the true reflected velocity determined from Equation 2.45. From Figure 2.15, the following relation also exists:

$$V'_r = \boldsymbol{c}'_r \cdot \hat{\boldsymbol{t}}_i = V_r \cos(\gamma_{ir}) \tag{2.47}$$

where $\gamma_{ir} = \gamma_i - \gamma_r$ is the inter-tangential reflection azimuth angle. And finally, from additionally visible geometry, the relations $V_r = u_r \tan(\theta_r)$ and $V'_r = u'_r \tan(\theta'_r)$ allow the final definition of θ'_r in terms of appropriate variables as

$$\tan(\theta'_r) = \frac{V_r \cos(\gamma_{ir})}{c_r \cos(\theta_r)} = \tan(\theta_r)\cos(\gamma_{ir}) \tag{2.48}$$

When the reflection lobe is wide enough, and the center is positioned close enough to the surface normal, it is possible that molecules may reflect "back", or to the left of $\hat{\boldsymbol{n}}$ in Figure 2.13. Otherwise, they will reflect "forward". Each of these cases, as well as the case where a reflection is perfectly orthogonal to the incidence plane must be treated differently for the selection of γ_{ir}. When $\theta'_r > 0$, the reflection is forward, and the viable range is that which covers the forward half of the surface plane: $\gamma_{ir} \in \mathcal{U}(-\pi/2, \pi/2)$. When $\theta'_r < 0$, the reflection is backward, and the viable range is the other half of the surface plane: $\gamma_{ir} \in \mathcal{U}(\pi/2, 3\pi/2)$. In the rare case of $\theta'_r = 0$, there are only two possible, viable values for γ_{ir}: $-\pi/2$ and $\pi/2$. This occurrence also implies that the calculation of the true cone angle θ_r will fail, as $\cos(\gamma_{ir}) = 0$ at this location [7] . However, the true meaning is that any value of θ_r will suffice. Lacking any rule for how to choose θ_r in this situation, a concession will be made that will instead prevent γ_{ir} from ever exactly equaling the problematic values. If it is detected that $\theta'_r = 0$, γ_{ir} should be chosen(uniformly) to be either $\pi/2(1-\epsilon)$ or $\pi/2(1+\epsilon)$, where ϵ is a prescribed perturbation that should be given as a fraction or percentage of $\pi/2$. This method becomes less desirable of a solution as the Schamberg parameter ν becomes large, thus pushing the kernel towards fully diffuse scattering were $\theta_c = 0$ is the most probable reflection direction. In this

[7] The effect of a cosine in the relation of Equation 2.48 also enforces the condition $\theta_r \leq \theta'_r$.

case, using Schamberg interaction is already inadvisable, however, as a standard diffuse model is more efficient. The full algorithm is a follows:

(1) Determine the incidence cone angle: $\theta_i = \cos^{-1}(-\hat{\boldsymbol{c}}_i \cdot \hat{\boldsymbol{n}})$, where both $\hat{\boldsymbol{c}}_i$ and $\hat{\boldsymbol{n}}$ must be unit vectors expressed in the same coordinate system – either the inertial/computational or surface-local system.

(2) Determine the tangential plane incidence azimuth angle: $\gamma_i = \tan^{-1}\left(\frac{w_i}{v_i}\right)$ where $v_i = \hat{\boldsymbol{c}}_i \cdot \hat{\boldsymbol{t}}_1$, $w_i = \hat{\boldsymbol{c}}_i \cdot \hat{\boldsymbol{t}}_2$).

(3) Calculate the reflected lobe center angle θ_c using Schamberg's relation (Equation 2.44).

(4) If thermal accommodation α is not specified as a constant, calculate it from an appropriate model.

(5) Calculate the reflected speed c_r from Equation 2.45.

(6) Generate the incidence plane projected angle θ_r' by sampling from the Schamberg kernel (Equation 2.43) in some fashion (likely the acceptance-rejection method).

(7) Generate the tangential plane inter-azimuth angle γ_{ir} by sampling from the appropriate uniform distribution, taking into account the sign of θ_r' to restrict the possible range of values to the appropriate half-plane as discussed earlier.

(8) Calculate the final azimuth angle: $\gamma_r = \gamma_i - \gamma_{ir}$.

(9) Obtain the true reflected cone angle θ_r using Equation 2.48.

(10) Calculate the reflected components of velocity in the surface-local system:

$$\begin{aligned} u_r &= c_r \cos(\theta_r) \\ v_r &= c_r \sin(\theta_r)\cos(\gamma_r) \\ w_r &= c_r \sin(\theta_r)\sin(\gamma_r) \end{aligned} \tag{2.49}$$

(11) Transform the reflected velocity into the appropriate inertial/computational coordinate system if necessary.

There is no direct way to obtain θ_r' as the kernel's cumulative density distribution function is not explicitly invertible for θ_r'. Thus, Step 6 is the most costly part of the algorithm. However, when performing an acceptance loop, only values of θ_r' within the lobe range need be considered as $f(\theta_r') = 0$ elsewhere. Thus, the acceptance procedure efficiency need not degrade as ϕ_L decreases.

An attempt to define a similar algorithm has been made by Zuppardi[76]. However, this method appears incomplete and possibly erroneous as it does not extend into 3D, uses a random reflected lobe width and random relational parameter ν, and the presentation lacks an appropriately detailed explanation for many of the choices made. In the context of implementation in a numerical simulation such as DSMC, the currently presented algorithm will generate exactly what is required when parameters ν, α (or μ if the Goodman model is used), and ϕ_L are provided.

2.1.5.2 Schamberg Cone

The cone formulation appears as a more natural or intuitive description of a quasi-specular reflection, and provides, by definition, all the information required to determine a reflection direction. Figure 2.16 shows the cone of reflection of half-width ϕ_L with its central angle at angle θ_c. The actual reflected direction is shown as the velocity $\boldsymbol{c}_r$. If the cone reflection angle $\theta_r > 0$, the reflected direction will not be in the incidence plane. The angular kernel of Equation 2.43 now describes the distribution of the true cone angle θ_r (replacing $\Delta\theta$) as shown in Figure 2.16. Note that the central axis of the cone (offset from the surface normal by central angle θ_c) lies in the incidence plane and the plane normal to this axis forms circle (the cone's base) within which an azimuth angle can be chosen uniformly. This cone base plane is depicted in Figure 2.17 where the angles γ_i and γ_r now refer to base-plane azimuth angles.

To generate reflected velocities, the cone is first considered to be upright in that its axis is collinear with the surface normal vector, with its base parallel to and above the surface. The reflected velocity $\boldsymbol{c}_r$ in the base-plane's coordinate system (spanned by basis vectors in each tangential

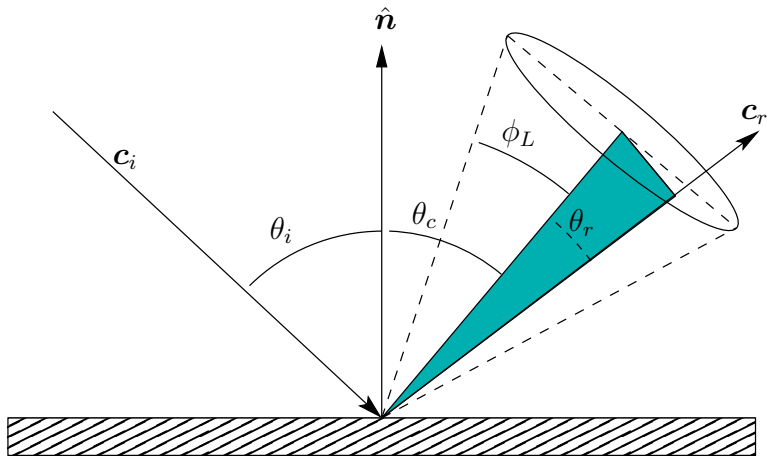


Figure 2.16: Reflection geometry for the Schamberg Cone formulation.

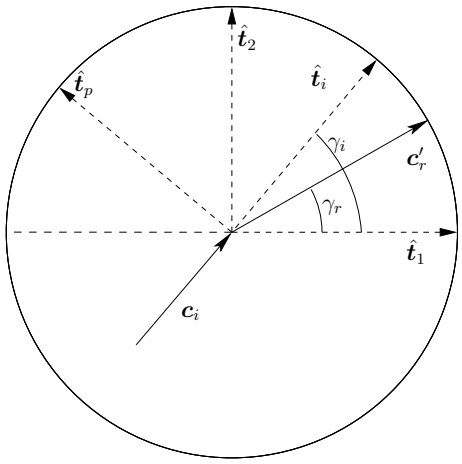


Figure 2.17: Schamberg Cone azimuthal geometry definitions in the cone base plane

direction of the surface $\hat{\boldsymbol{t}}_1$, $\hat{\boldsymbol{t}}_2$ as seen in Figure 2.17) is then written as

$$\boldsymbol{c}'_r = c'_r \cos(\gamma_r)\hat{\boldsymbol{t}}_1 + c'_r \sin(\gamma_r)\hat{\boldsymbol{t}}_2 \tag{2.50}$$

where the base-plane reflected speed and normal direction velocity component are, respectively,

$$c'_r = c_r |\sin(\theta_r)| \tag{2.51}$$

$$u'_r = c_r \cos(\theta_r) \tag{2.52}$$

and the total reflected speed c_r is obtained from Equation 2.45. This forms a pre-rotated velocity

of

$$\boldsymbol{c}_r' = c_r|\sin(\theta_r)|\cos(\gamma_r)\hat{\boldsymbol{t}}_1 + c_r|\sin(\theta_r)|\sin(\gamma_r)\hat{\boldsymbol{t}}_2 + c_r\cos(\theta_r)\hat{\boldsymbol{n}} \tag{2.53}$$

This velocity then undergoes the following rotations to place it in the proper orientation in the surface frame:

$$\boldsymbol{c}_r = Q_3(\gamma_i)Q_2(\theta_c)\boldsymbol{c}_r' \tag{2.54}$$

where rotations are defined as

$$Q_2(\theta_c) = \begin{bmatrix} \cos(\theta_c) & 0 & \sin(\theta_c) \\ 0 & 1 & 0 \\ -\sin(\theta_c) & 0 & \cos(\theta_c) \end{bmatrix} \tag{2.55}$$

$$Q_3(\gamma_i) = \begin{bmatrix} \cos(\gamma_i) & \sin(\gamma_i) & 0 \\ -\sin(\gamma_i) & \cos(\gamma_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.56}$$

The cone's axis remains in the incidence plane but the reflected velocity may be anywhere within the cone. The complete procedure is as follows and starts with the same steps as the Wedge model but differs beginning at Step 6:

(1) Determine the incidence cone angle: $\theta_i = \cos^{-1}(-\hat{\boldsymbol{c}}_i \cdot \hat{\boldsymbol{n}})$, where both $\hat{\boldsymbol{c}}_i$ and $\hat{\boldsymbol{n}}$ must be unit vectors expressed in the same coordinate system – either the inertial/computational or surface-local system.

(2) Determine the tangential plane incidence azimuth angle: $\gamma_i = \tan^{-1}\left(\frac{w_i}{v_i}\right)$ where $v_i = \hat{\boldsymbol{c}}_i \cdot \hat{\boldsymbol{t}}_1$, $w_i = \hat{\boldsymbol{c}}_i \cdot \hat{\boldsymbol{t}}_2$).

(3) Calculate the reflected lobe center angle θ_c using Schamberg's relation (Equation 2.44).

(4) If thermal accommodation α is not specified as a constant, calculate it from an appropriate model.

(5) Calculate the reflected speed c_r from Equation 2.45.

(6) Generate the true cone angle θ_r by sampling from the Schamberg kernel (Equation 2.43) in some fashion (likely the acceptance-rejection method).

(7) Generate the cone base-plane azimuth angle γ_r by sampling from a uniform distribution $\mathcal{U}(0, 2\pi)$.

(8) Create the unrotated reflected velocity using Equation 2.53.

(9) Rotate the velocity using Equation 2.54.

(10) Transform the reflected velocity into the appropriate inertial/computational coordinate system if necessary.

Whether the cone formulation is truly "correct" is unclear. Particularly, the choice of using a uniform distribution for γ_r is rather arbitrary. Under the assumption of a *cone* angle, however, a uniform distribution seems appropriate. Should more analysis or experimental data become available in the future to validate Schamberg's kernel and its interpretations, these implementations will become more pertinent.

2.1.6 Custom Incidence Plane Projected Angular Kernel

The velocity generation algorithm proposed for the Schamberg Wedge formulation can be generalized to work for any angular kernel. If future experiments, analysis or spacecraft flight data lead to new angular kernels, the wedge procedure can generalize them to any isotropic surface and provide an implementation in code. This would allow custom reflection kernels to be used in DSMC, further enhancing the overall fidelity of rarefied gas flow over real engineering surfaces. Either analytical or numerical kernels could be provided, and Step 6 of the Wedge algorithm in Section 2.1.5.1 would be replaced by a sampling procedure for these kernels, rather than for the Schamberg kernel.

2.1.7 SESAM for Accommodation

Many GSI models are composed of a scattering kernel and an accommodation model that acts as input to the kernel in the form of an effective reflected temperature. Sentman reflection makes inclusion of thermal accommodation obvious to implement. Schamberg reflection can include thermal accommodation if desired when choosing the reflected speed (Equation 2.45). Cercignani-Lampis includes more complicated relations between both thermal and momentum accommodations and the reflection kernels. Providing accommodation coefficients can be difficult to do accurately without experimental evidence to support a certain choice of a coefficient. Until Goodman's thermal accommodation model [27, 26], most, if any values for thermal accommodation coefficients were provided from analysis of what little experimental and in-situ data that were available. Goodman's model is based on a representation of surfaces as a 2D matrix of hard sphere molecules into which incident molecules are adsorbed and from which they are later re-emitted. It introduces more detailed surface physics into the problem of determining accommodation. The model predicts the thermal accommodation coefficient as a function of incident and resident molecular mass ratio. Equation 2.58 is Goodman's formula for determination of thermal accommodation, with a steric factor (the incidence angle θ_i measured from the surface normal as shown in Figure 2.13) included, and the parameter

$$\mu = \overline{m}_g / m_s \tag{2.57}$$

is the ratio of the mean molecular mass of incident molecules to the mass of molecules comprising the surface. However, most of these results are based on experimental correlations where clean surfaces (i.e. not "engineering" or real surfaces) were used.

$$\alpha_n = \frac{3.6\mu \cos(\theta_i)}{(1+\mu)^2} \tag{2.58}$$

Despite the Goodman model being in use since it was introduced, it has not been sufficient for application to real surfaces or to all gases and conditions. Even when restricted to atomic oxygen (a common adsorbate), it fails to accurately predict the drag coefficients measured on certain orbits by certain spacecraft. The Semi-Empirical Satellite Accommodation Model (SESAM), introduced

by Pilinksi [57, 58], gives improved results over Goodman for accommodation of atomic oxygen, and possibly for other adsorbates [8] . This model requires inputs of bulk flow incidence energy and partial pressure of the adsorbate and is restricted to use in a Maxwellian diffuse scattering kernel only. For simulations involving atomic oxygen where thermal accommodation is required to be known (such as most low-earth orbital problems between 200 and 400 km in altitude), SESAM is a useful new model.

The implementation of SESAM as it is presented by Pilinski is for flow with a defined stream or characteristic velocity and assumes that the other inputs such as normal incident energy and pressure are given in some averaged context. SESAM is formulated under the assumption of a traditional drag model being sufficient and applying the resulting accommodation coefficient to the entire object. In order to use SESAM in a general class of problems in a DSMC code (a requirement for performing dynamic analysis of objects in rarefied flow without restriction on geometry or other conditions), it must be adapted to apply in the context of an arbitrary surface element where only local flow information is available. This implies that, in real flow, different parts of an object's surface may experience different local flow properties and thus may evolve differently in terms of how they are affected by adsorption. Additionally, a properly generalized code may only specify boundary conditions, and not internal flow structure or state. Therefore, no assumptions of characteristic flow should be made.

An element-wise version of SESAM can be created by first noting what the model's parameters are and where they apply. SESAM requires four parameters:

- E_b : Binding energy of the surface [J]. Incident molecules whose normal kinetic energy is less than E_b are considered for possible adsorption on the surface.

- m_s : Mass of a surface molecule [kg]. The aforementioned ratio μ is used in the Goodman function, Equation 2.58, which is used by SESAM to determine a baseline accommodation coefficient.

[8] Verification data is not yet available for other adsorbates.

- $K_{L,o}$, $K_{L,f}$: Lower and upper range orbital eccentricity empirical fit parameters for a given adsorbate (e.g. atomic oxygen) [Pa^{-1}]. These are used to calculate the Langmuir fitting constant K_L, given by

$$K_L = s_0 K_{L,o} + K_{L,f} \tag{2.59}$$

  where s_0 is a "sticking coefficient".

The sticking coefficient s_0 is originally determined by Pilinski using analytical representations of energy distributions in free-molecular flow as the ratio of low-energy incident molecules to all possible energies in the flow. It is a probability of sticking to the surface based on a comparison of energy and *not* the fraction of incident molecules that have already stuck. This is not a valid way to determine this ratio in an element-wise formulation. Instead, since each surface element experiences actual impacts by simulation particles, the sticking coefficient can be directly determined by comparing the total number of incident molecules with low (i.e. less than E_b) normal kinetic energy to the total number of all molecules that have impacted it over the course of the simulation, as these records are easily kept for each element. Similarly, the current incident-only partial pressure p_i on the surface element should be available. From this pressure, and the element-wise Langmuir parameter K_L, a surface coverage parameter θ' is determined by using

$$\theta' = \frac{p_0 K_L}{1 + p_0 K_L} \tag{2.60}$$

where the pressure-proxy p_0 is

$$p_0 = \frac{-p_i}{k} \tag{2.61}$$

and k is Boltzmann's constant. It is assumed that the definition of surface pressure is such that $p_i < 0$.

The final procedure for calculating thermal accommodation with element-wise SESAM is as follows, and begins when a surface element detects an impacting particle:

(1) Calculate the incident normal kinetic energy $E_{in} = 1/2 m_g c_{in}^2$, where incident normal speed is $c_{in} = \boldsymbol{c}_i \cdot \hat{\boldsymbol{n}}$ and $\boldsymbol{c}_i$ is the incident velocity.

(2) Record and update the number of low-energy and total impacts by testing

`if(`$E_{in} < E_b$`)`

$N_{low} = N_{low} + 1$

`end if`

and also:

$N_{tot} = N_{tot} + 1$

These numbers are kept as a cumulative record for each surface element.

(3) Obtain the sticking coefficient as

$$s_0 = \frac{N_{low}}{N_{tot}} \tag{2.62}$$

(4) Calculate the Langmuir parameter using Equation 2.59.

(5) Calculate the pressure-proxy with Equation 2.61, assuming p_i is available for this molecule's species[9] .

(6) Obtain the Goodman baseline accommodation using Equation 2.58 and using the mass of the actual impacting molecule m_g instead of the mean mass in Equation 2.57. Call this $\alpha_{n,g}$.

(7) Calculate the coverage parameter θ' from Equation 2.60.

(8) Calculate the final value of thermal accommodation as

$$\alpha_n = (1 - \theta')\alpha_{n,g} + \theta' \tag{2.63}$$

(9) Determine the effective reflected translational temperature from Equation 2.20.

Whether this version of SESAM designed for element-wise operation properly recovers the results seen in the original, mean-value version has yet to be determined. Once this verification has been performed to satisfaction, application of SESAM in DSMC can be done with more confidence.

[9] The current value of macroscopic incident pressure p_i should be taken as a mean value over the most recent sample interval only, like other macroscopic variables in DSMC. This slightly contrasts with the tracking of N_{low} and N_{tot} which are summed over the entire computation. The reason for the the latter parameters being absolutely cumulative is that s_0 aims to provide some modeling of the surface becoming "dirtied" with adsorbing molecules, permanently changing its accommodation.

2.1.8 Blended Surface Models

One common deficiency in current DSMC codes is the lack of options for specifying surface models. Voldipar provides the user the option of creating custom surface models that use superpositions of the various models outlined in this section. For each surface element defined in the simulation, a surface interaction model may be specified by providing a list of models to use, with their appropriate parameters, and a corresponding list of weighting coefficients $w_j \in (0..1]$. The weighting blends the models for the surface such that model j will be selected with probability w_j. For example, a surface could be specified as 30% Maxwellian standard diffuse with SESAM thermal accommodation, 50% CL with full thermal accommodation and partial momentum accommodation ($\alpha_n = 1.0$, $\sigma_n = 0.8$, $\sigma_t = 0.35$) and 20% Schamberg with $\nu = 5.0$ and partial thermal accommodation ($\alpha_n = 0.3$). This configuration would be set with $w_1 = 0.30$, $w_2 = 0.50$, $w_3 = 0.20$ where it is necessary for the sum of weighting factors over j to be unity. To select a surface model, a uniformly distributed random number $R_f \in (0..1]$ is generated and its location with respect to the limits of each region is determined. The models must be arranged in some manner corresponding to space in the range (0..1] as shown in Figure 2.18.

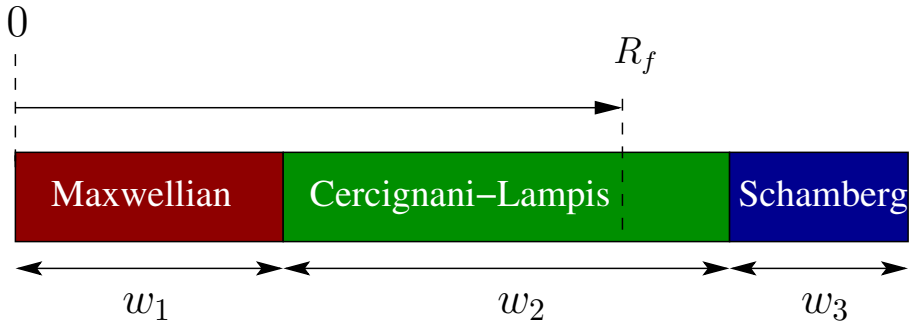


Figure 2.18: Selection of a surface reflection model from the example model profile consisting of three different models, each with a different weighting specified by the user. The CL model is shown to be selected.

2.2 Summary

Regardless of how spacecraft dynamics is represented, it remains important to understand how molecules interact with surfaces of solid bodies in the stream. With this motivation in mind, Voldipar was created to allow for direct insertion of gas-surface interaction (GSI) models in a proper manner. Rather than forcing the user of the code to attempt to approximate the complexities of a given GSI model by making use of a few commonly available parameters (e.g. surface overall temperature), the goal is to provide any information that a model would require to the user (e.g. gas velocity distributions or macroscopic variables near the surface), and to allow the user to provide additional information to the code (e.g. separate surface and effective reflection temperatures). With complete control over the contents of a DSMC code, GSI models can be investigated more thoroughly and accurate models of real engineering surfaces under more realistic conditions may be made.

Chapter 3

Rigid-Body Dynamics: Simple Cases and Initial Models

Much of the effort expended to create reliable, accurate and friendly rarefied gas dynamics simulation tools such as DSMC would be wasted without a greater goal. In this work, that goal is the improvement in modeling and prediction of the motion of spacecraft and orbital debris. Understanding the effects of dynamic forces on objects immersed in a fluid remains an obstacle to obtaining accurate models of such objects' natural motion. In the specific regime of rarefied gas flow, as is seen for spacecraft and orbital debris in low-earth orbit and beyond, most modeling makes little attempt to capture complicated motion beyond very general parameterization involving drag or ballistic coefficients and the imposition of effective geometric properties. With the help of modern computational tools such as DSMC, the fidelity with which spacecraft dynamics can be modeled can and should be increased.

Prior to attempting to solve the greater problem of modeling gas dynamic forces on rigid bodies in rarefied environments–which is the subject of Chapter 4–some attention should be paid to understanding the type of dynamic motion experienced by said bodies and what challenges are present. A simple approach to this class of problems can start with the assumption of a 2D body in free-molecular flow with classical atmospheric equations of motion to serve as an appropriate example case. No other restrictions on body geometry should be necessary. Additionally, one set of general conditions that are common in low-earth orbit will be imposed to limit the analysis to a tractable parameter space. Extensions to three-dimensional geometry and higher dimension parameter and state spaces should follow from this simple model. The restriction to 2D allows for

analysis using the current state of Voldipar, once the basic free-molecular results have provided proof of satisfactory framework.

3.1 2D Dynamic Equations of Motion

The equations of motion for a 2D rigid body in rarefied flow are first written in state-space form in the body-fixed coordinate frame shown in Figure 3.1. In this formulation, the body is assumed to be moving at a constant orbital speed of V_∞ without the direct influence of gravity. The center of mass is assumed to follow a constant trajectory with respect to the inertial coordinate frame fixed to the earth, reducing the motion to a single degree-of-freedom rotation about the center of mass. A 2D computational/inertial frame is constructed where the stream velocity is always aligned with the computational x-direction. These assumptions create a weather-vane behavior where the rigid body is pinned and rotates about its center of mass so that the equation of motion is written as

$$\frac{d\dot{\alpha}(t)}{dt} = \frac{M(\alpha(t), \dot{\alpha}(t))}{I_{yy}} \tag{3.1}$$

The free-stream Knudsen number Kn and Mach number Ma are assumed constant, with $\alpha(t)$ as the angle of attack and M is the sum of all aerodynamic moments acting on the body. The rotation rate of the body frame with respect to the computational frame is $\dot{\alpha}(t)$ so that $q(t) = \dot{\alpha}(t)$, where $q(t)$ is typically the body-frame pitch rate. Equation 3.2 gives the state-space form with the state vector $\boldsymbol{s} = [\alpha(t)\ \dot{\alpha}(t)]^{\mathrm{T}}$.

$$\dot{\boldsymbol{s}} = \begin{bmatrix} \dot{\alpha}(t) \\ \frac{1}{I_{yy}} M(\alpha(t), \dot{\alpha}(t)) \end{bmatrix} = \begin{bmatrix} s_1 \\ \frac{1}{I_{yy}} M(s_1, s_2) \end{bmatrix} \tag{3.2}$$

The aerodynamic moment M may be a function of any number of independent variables in the problem, though is expected that M will depend mostly on α.

Running a simulation of these dynamics requires that Equation 3.2 to be integrated in time with initial conditions $\boldsymbol{s}_0$ specified, and the general flow or environment conditions to be defined *a priori*. The free-stream conditions Ma , T_∞, Kn are assumed constant, as are the object surface

temperature T_S and gas properties. Furthermore, the object's geometry and mass properties must be given. With these parameters and the object's geometry and mass properties specified, the only remaining unknown is the fluid moment, which must be computed as a function of the state.

To compute M requires information extracted from a numerical simulation or experiment. The simplest solution is to integrate the equations of motion in time at each time step of the flow field simulation. This method will be referred to as "in-loop" (also called "coupled-field"). In the general case, the simulation may be any relevant fluid model (e.g. Navier-Stokes for continuum fluids) that can be formulated to take the state as an input. For the case of rarefied ($\mathrm{Kn} \gtrsim 10$) and continuum-rarefied transition ($0.1 \leq \mathrm{Kn} \leq 10$) flows, the simulation might employ DSMC. Whichever method is used for obtaining fluid forces in-loop, the cost is almost always too high for practical simulation purposes. What is required is a method of approximation of fluid forces that can be used in place of direct in-loop simulation. In the example of classical aircraft dynamics this is accomplished with a Taylor series expansion about a single point in order to reduce fluid effects to a set of constants. When this assumption cannot be made, as is the case almost everywhere outside of fixed-wing atmospheric aircraft flight, alternative methods must be explored.

A free-molecular gas ($\mathrm{Kn} > 100$) where an analytical formulation can be used to determine M is used for a simplified initial analysis. This enables an analytical approach, presented in Section 3.2, for the body forces and moments as a function of state. This will be referred to as the *source data* used as the input to the approximation methods. A result of this formulation is that the methods remain independent of the source data or the conditions under which they were computed or otherwise provided (including experimental data). Therefore, any method of approximation determined to be appropriate while using free-molecular source data should be equally viable for continuum flow source data.

3.2 Free-Molecular Source Data

Analytical force equations are available (e.g. [3]) for the free-molecular flow assumption. The 2D rigid body is represented by a set of points in a plane with the line segments between these

points defined in a winding order such that each segment has an outward-pointing normal vector as shown in Figure 3.1. The force on a segment is determined by examining the impacts and

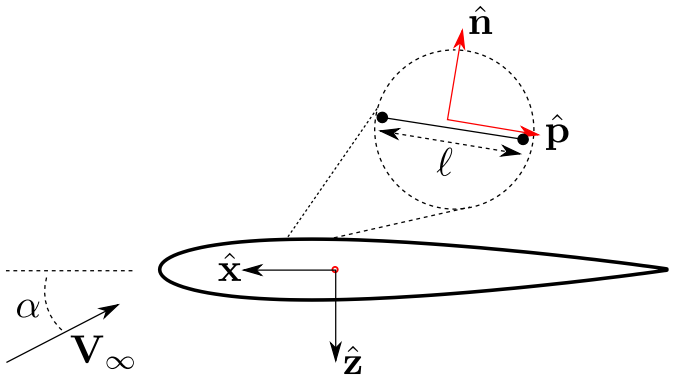


Figure 3.1: 2D airfoil geometry with body-fixed coordinates and angle of attack $\alpha > 0$.

reflections of molecules on each line segment or panel. The equations for pressure and shear on a panel are determined for a prescribed free-stream velocity and a standard diffusive surface reflection model. The equations for incident pressure and shear are:

$$p_{\text{inc}} = \frac{\rho}{2\beta^2\sqrt{\pi}}\left[-sV_n e^{-(sV_n)^2} + \sqrt{\pi}\left(1-\text{erf}(sV_n)\right)\left(\frac{1}{2}+s^2V_n^2\right)\right] \tag{3.3}$$

$$\tau_{\text{inc}} = \frac{\rho s\sqrt{1-V_n^2}}{2\beta^2\sqrt{\pi}}\left[e^{-(sV_n)^2} - \sqrt{\pi}sV_n\left(1-\text{erf}(sV_n)\right)\right] \tag{3.4}$$

$$n_r = n_\infty\left[\frac{T_\infty}{T_r}\right]^{\frac{1}{2}}\left[e^{-(sV_n)^2} - \sqrt{\pi}sV_n\left(1-\text{erf}(sV_n)\right)\right] \tag{3.5}$$

$$p_r = \frac{n_r m_g}{4\beta_r^2} \tag{3.6}$$

$$\beta = (2RT_\infty)^{-\frac{1}{2}} \tag{3.7}$$

$$s = \beta|\boldsymbol{V}_\infty| \tag{3.8}$$

$$V_n = \frac{\mathbf{V}_\infty\cdot\hat{\mathbf{n}}}{|\mathbf{V}_\infty|} \tag{3.9}$$

$$V_p = \frac{\mathbf{V}_\infty\cdot\hat{\mathbf{p}}}{|\mathbf{V}_\infty|} \tag{3.10}$$

The variable β is the reciprocal of the most-probable molecular speed of the incident gas. The reflected temperature T_r is distinguished from the true surface temperature T_S since gas-surface

accommodation models may be used to derive a value of T_r that is not equal to T_S. Although this formulation allows a distribution of temperature over the body surface, here T_S is constant. For the diffuse reflection model, there is no preferred reflection direction, thus $\tau_r = 0$. The resulting values of pressure and shear on each panel are given by

$$p = -(p_{\text{inc}} + p_r) \tag{3.11}$$

$$\tau = \begin{cases} -\tau_{\text{inc}} & \text{for } V_p < 0 \\ 0 & \text{for } V_n > 0 \\ \tau_{\text{inc}} & \text{otherwise} \end{cases}$$

For consistency, pressure is defined as positive along the panel normal direction or "outward". There is also a natural ambiguity in the sign of the shear stress that is corrected by defining τ with the different cases shown in Equation 3.12. A difference to note, however, is the case of $V_n > 0$. This case corresponds to a backward-facing panel with respect to the oncoming stream (the sample panel shown in Figure 3.1 is backward-facing, for example). For free-molecular flow, a backward-facing surface is in the aerodynamic shadow of the flow, and without intermolecular collisions, the only impacts with the surface will occur due to thermal back-flow. With no preferred direction imposed from the stream, these impacts cannot contribute any mean shear stress for the same reason there is no reflected shear in a diffuse model.

For a panel i, the surface stress vector is then defined as $\boldsymbol{\sigma}_i^{(p)} = [p_i \;\; \tau_i]^{\text{T}}$, where the p superscript indicates that it is expressed in the local panel coordinate system. This vector must expressed in the body-fixed coordinate system (Figure 3.1), multiplied by the panel length and summed over all panels on the body (Equation 3.12) to obtain the overall force vector in body coordinates. Body moment is computed according to Equation 3.13, where $\boldsymbol{x}_{\boldsymbol{c_i}}$ is the center of panel i and $\boldsymbol{x}_{\text{ref}}$ is a user-defined reference point (e.g. the center of mass of the body, $\boldsymbol{x}_{\text{cm}}$), both expressed in computational coordinates.

$$\boldsymbol{F}^{(b)} = \sum_i \boldsymbol{F}_i^{(b)} = \sum_i \ell_i (\boldsymbol{B}_i)_p^b \, \boldsymbol{\sigma}_i^{(p)} \tag{3.12}$$

$$\boldsymbol{M} = \sum_i \left[(\boldsymbol{x}_{\boldsymbol{c_i}} - \boldsymbol{x}_{\text{ref}}) \times \boldsymbol{F}_i^{(b)} \right] \tag{3.13}$$

$$(\boldsymbol{B}_i)_p^b = [\hat{\mathbf{p}}_i \ \hat{\mathbf{n}}_i] \tag{3.14}$$

For the purpose of further implementation, this process is represented by the calling of the function $M(\boldsymbol{s}, t)$ as shown in Equation 3.2. This function, $X(\boldsymbol{s}, t)$, $Z(\boldsymbol{s}, t)$, and any other functions of the state variables and fluid dynamics are *source functions*. The dynamics discussed in this investigation do not require the force source functions $X(\boldsymbol{s}, t)$ and $Z(\boldsymbol{s}, t)$, however, they are included for completeness.

Using this method, the pitching moment, as a function of α, is calculated for a 1m chord NACA-0012 airfoil section using Equation 3.13, for a range of Mach numbers, as plotted in Figure 3.2. The conditions shown in Figure 3.2 are for Earth's atmosphere (assumed to be pure diatomic nitrogen) at 155.5 km and $T_S = 300\,\text{K}$ with the moment reference location at the center of mass $\boldsymbol{x}_{\text{cm}} = (0.4205, 0.0000)\,\text{m}$ from the leading edge. Mach numbers Ma =14.97 and Ma =24.18 correspond to circular orbits at Kn = 100 and Kn = 1, respectively. For angle of attack in the approximate range $-10\text{deg} < \alpha < 10\text{deg}$, the moment experienced by the airfoil causes it to destabilize or diverge. The magnitude of the moment in this region increases as Ma increases. All other angles, however, cause a restoring or zero moment. Details of this behavior are examined in Section 3.3.1.

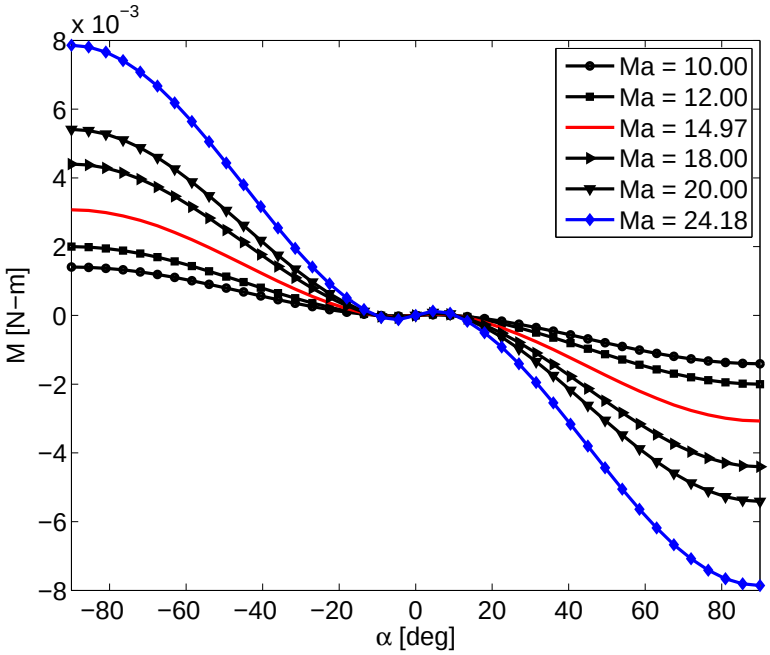


Figure 3.2: Pitching moment $M(\alpha)$ of a NACA-0012 airfoil with 1m chord in free-molecular flow at various free-stream Ma values.

3.2.1 Treatment of Angle of Attack Rate

The source functions, as implemented using the free-molecular equations, must be functions of the declared state variables. The incident velocity component V_n is related to the first state variable α. The angular rotation rate of the object, $\dot{\alpha}$ (which is assumed to remain constant for this formulation), is introduced by replacing the incident velocity on each panel with an effective velocity $\boldsymbol{V}_e$ for that panel, defined as

$$\boldsymbol{V}_e = \boldsymbol{V}_\infty + \begin{bmatrix} 0 \\ \dot{\alpha} \\ 0 \end{bmatrix} \times (\boldsymbol{r}^{(c)} - \boldsymbol{x}^{(c)}_{\text{ref}}) \tag{3.15}$$

where $\boldsymbol{r}^{(c)}$ is the location of the center of the panel in the computational frame. The state variable $\dot{\alpha}$ remains constant over an integration time step. Each panel's effective velocity is the sum the free-stream velocity and the instantaneous velocity derived from the motion of the panel as it rotates

about the given reference point $\boldsymbol{x}_{\text{ref}}$ at a rate $\dot{\alpha}$. This is sometimes described as a transpiration boundary condition and is implemented in continuum computational fluid dynamics as a cost-effective way to simulate moving boundaries [24, 61, 59]. For the general formulation presented now, use of transpiration to model rotation will be assumed to be acceptable.

3.3 Free-Molecular Simulations

Test cases involving a 1m chord NACA-0012 2D airfoil in free-molecular flow were performed for various initial conditions for both the in-loop baseline and the Sliding Taylor method of approximation. The airfoil is assumed to be comprised of a homogeneous material of mass-density ρ. Source function $M(\alpha)$ and moment of inertia I_{yy} are calculated with a reference location being the airfoil's center of mass of $\boldsymbol{x}_{\text{cm}} = (0.4205, 0.0000)$ m. As the dynamics are essentially flat-earth aircraft equations of motion in the absence of gravity, it was useful to examine the in-loop natural response of the body when starting from high and low angles of attack and to inspect the movement of the center of pressure to ensure that the responses conform to expected general stability behavior. The Sliding Taylor approximation is then integrated under the same conditions and the quality of the new, approximated dynamics are compared to the in-loop baseline. For all cases, the simulation conditions that remain constant are given in Table 3.1. These conditions, taken from the NRLMSISE model [56], correspond to an object in a circular orbit at an altitude of approximately 155.5km. Molecular nitrogen is dominant and thus, is assumed to be the only species.

If the characteristic length of the problem is defined as the chord length, then the mean free path of the stream and overall system is $\lambda_\infty = 100\,\text{m}$. From this definition, the free-stream number density is determined, using the Variable Hard Sphere approximation [3], as

$$n_\infty = \frac{1}{\sqrt{2}\pi d_g^2 \lambda_\infty} \tag{3.16}$$

3.3.1 Free-Molecular Baseline Cases: Airfoil

The first case used to validate the free-molecular dynamics is given in Table 3.2 as Case 1. Integrating Equations 3.2 with in-loop evaluation of source functions over an interval of $t \in$

Table 3.1: Constant rarefied gas flow conditions and mass properties for all simulations of a NACA-0012 2D airfoil section in molecular nitrogen (N_2) at gas reference temperature of 273 K.

Variable	**Symbol**	**Value**	**Unit**
Mach number	Ma	14.97	
Knudsen number	Kn	100	
Stream temperature	T_∞	655	K
Body surface temperature	T_S	300	K
Body moment of inertia about y	I_{yy}	12.43	kg–m^2
Gas molecular mass	m_g	46.5 $\times 10^{-27}$	kg
Gas diameter	d_g	4.17 $\times 10^{-10}$	m
Gas ratio of specific heats	γ	7/5	
Gas constant	R	296.8	kJ/(kg–K)

$[0, 2000]$ s gives the resulting oscillation seen in Figure 3.3a. This case starts with an initial angle of attack that is within the diverging central region of $M(\alpha)$ as seen in Figure 3.2. The airfoil initially rotates upward but quickly passes into the restoring range, causing a reversal of rotation direction. It proceeds to oscillate about an angle offset of approximately ten degrees without apparent physical damping for the remainder of the time interval. As the source functions impose a free-molecular set of forces and moments, no fluid damping should occur. With no intermolecular collisions, there is no net transport of momentum within the gas, thus preventing viscous damping. Due to the geometry, a symmetrically-centered oscillation occurs. If Case 2 is simulated, the airfoil is given some initial negative rotation, causing a symmetric oscillation that is centered about $\alpha = 0$ as seen in Figure 3.3b. The initial downward rotation allows the airfoil to start with enough momentum to avoid an initial upward rotation and immediately settle into its dynamically stable motion.

Simulation was then performed for the higher initial angles of Case 3, shown in Figure 3.4a. Another symmetric oscillation occurs. Since α_0 now starts within the restorative range of $M(\alpha)$, the airfoil immediately rotates downward. Enough moment is present to enforce a rotation past $\alpha = 0$, though a brief slowing occurs as the body re-enters the divergent region. Case 4 was then simulated, in which an initial *upward* rotation of 0.5deg/s was added. Figure 3.4b shows the airfoil initially rotating upward, but still enters a stable oscillation about $\alpha = 0$ as the initial angular momentum is not enough to cause an unbounded rotation. Although, a destabilization does occur

Table 3.2: Case definitions: Initial conditions

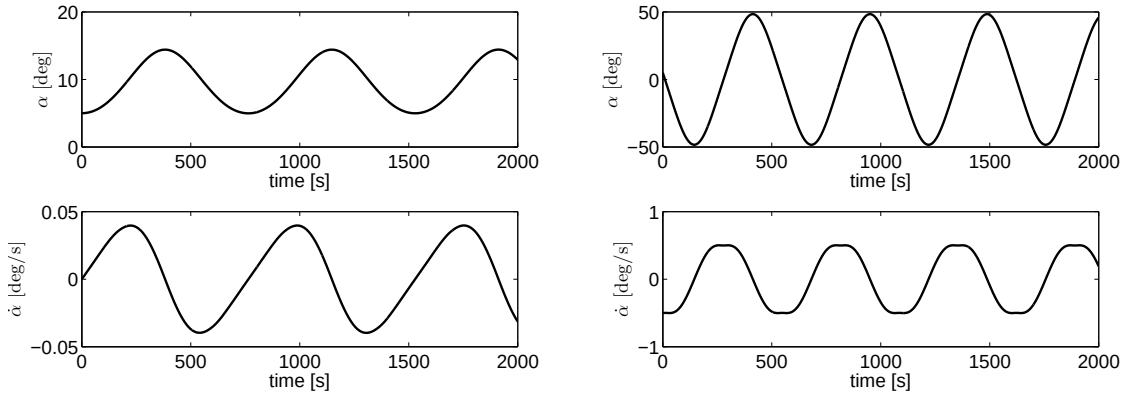
Case	α_0 [deg]	$\dot{\alpha}_0$ [deg/s]
1	5.0	0.0
2	5.0	-0.5
3	25.0	0.0
4	25.0	0.5
5	25.0	2.0

when the initial rate is raised to 2.0 deg/s, as shown in the time histories of Case 5, given in Figure 3.4c.

The dynamic stability exhibited by Case 1 can be examined by noting the location of center of pressure as α changes. The center of pressure $\boldsymbol{x}_{\text{cp}}$ changes as a function of the state (particularly of α) and is given by

$$\boldsymbol{x}_{\text{cp}} = \boldsymbol{x}_{\text{cm}} - \frac{M}{X^2 + Z^2} \begin{bmatrix} Z \\ -X \end{bmatrix} \tag{3.17}$$

The movement of the center of pressure can be seen in Figure 3.5 where the center of mass remains ahead of the center of pressure for most values of α. Angles within the divergent region lead to the center of pressuring becoming located ahead of the center of mass. Figure 3.5 also shows the consequence of symmetry with the center of pressure becoming identical to the center of mass at $\alpha = 0$. A deeper look at the dynamics begins by examining the state-space phase portraits with the conditions of Table 3.1 as shown in Fig. 3.6. There are several dynamic features to note. The first is the presence of periodic (in α) limit sets within which Lyapunov stable orbits exist. Each of these regions exists at intervals of $\alpha = 2n\pi$, where n is any integer (only the region for $n = 0$ is shown in Figure 3.6a). No value of $\dot{\alpha}_0$ gives a stable trajectory for $\alpha = n\pi$, except for the origin (which is an equilibrium). Saddle points exist that straddle neighboring stable regions located at $\dot{\alpha}_0 = 0$ for $\alpha = n\pi/2$. Figure 3.6b shows the central region in more detail, where some additional features are visible. In this region, there exist limit sets that correspond to stable oscillations about the points of $\alpha^* = |\alpha| \approx 10.3276$ deg with $\dot{\alpha} = 0$. These points represent the boundaries of the restorative region, to which trajectories that begin outside it converge. However, with enough energy in the

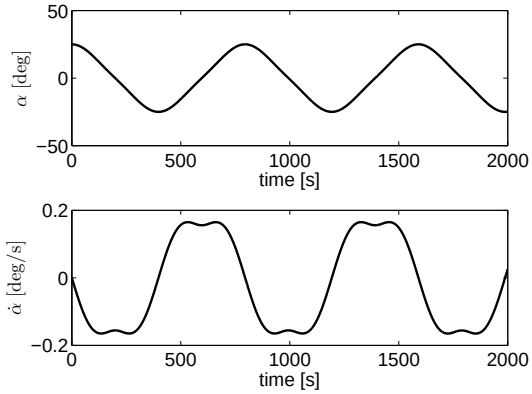


(a) Case 1.

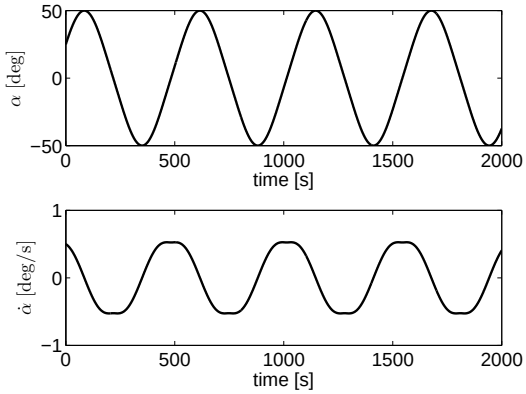
(b) Case 2.

Figure 3.3: State variable time histories for cases with initial angles of attack within the divergent region of $M(\alpha)$.

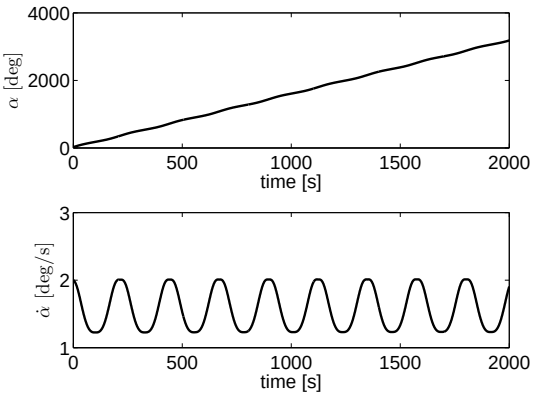
state, as exemplified by Cases 2 to 5, trajectories may pass these regions completely and instead appear to oscillate about the the origin. The origin itself is an equilibrium, but also a saddle point in that any small deviation in angle of attack (at $\dot{\alpha} = 0$) will push the motion onto one of the special, limited trajectories about α^*. The points at α^* are somewhat difficult to characterize due to necessarily high integration tolerances, but they appear to be unstable equilibria. The region surrounding positive α^* is given in detail in Figure 3.6c. By creating these phase portraits, these dynamic characteristics become apparent. Information that was previously unavailable or merely speculative yields more quantifiable conclusions.



(a) Case 3.



(b) Case 4.



(c) Case 5.

Figure 3.4: State variable time histories for cases with initial angles of attack within the restorative region of $M(\alpha)$.

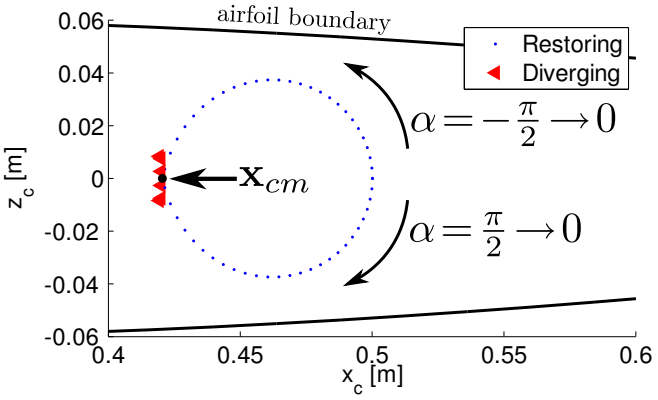
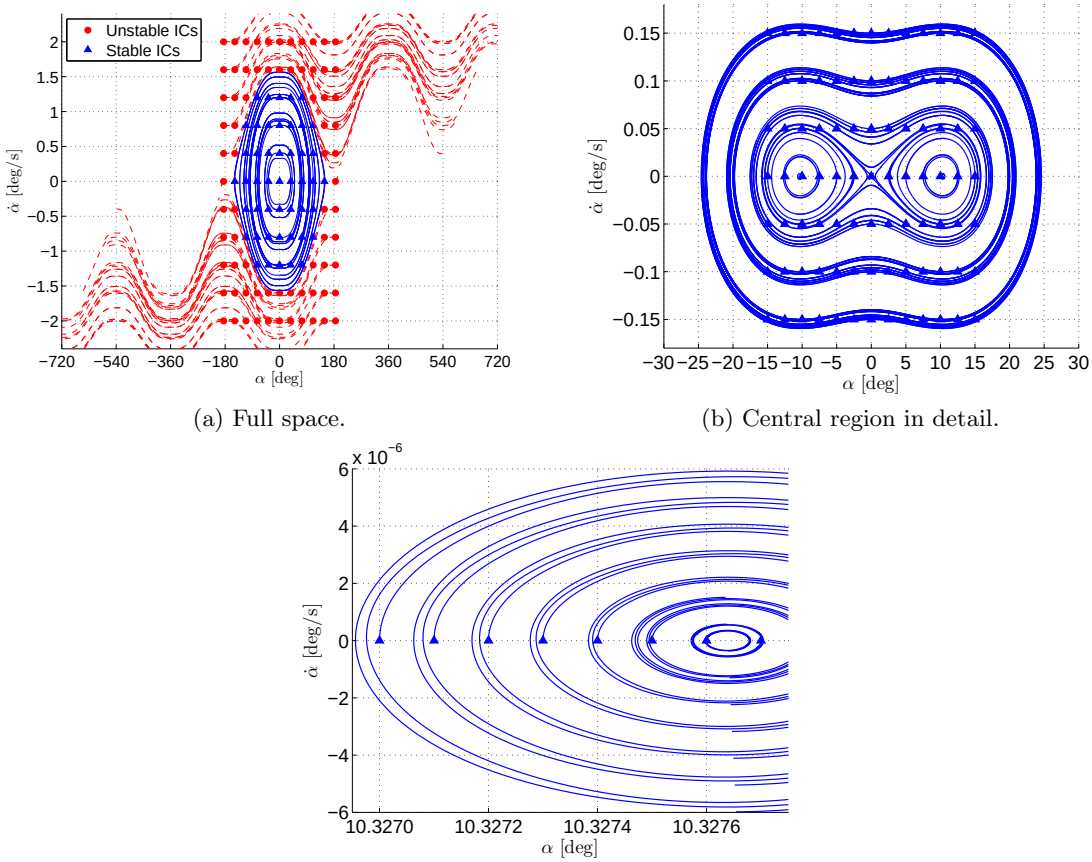


Figure 3.5: Movement of the center of pressure $\boldsymbol{x}_{\mathrm{cp}}(\alpha)$.



(a) Full space.

(b) Central region in detail.

(c) Region near unstable equilibrium $+\alpha^*$.

Figure 3.6: Phase portraits for conditions of Table 3.1. The initial condition (IC) for each trajectory is shown as a marker.

3.3.2 Free-Molecular Baseline Cases: Arbject

The symmetry of the NACA-0012 is an attractive property to ensure simple physical motion occurs as expected. To examine dynamic motion of more realistic objects, an object with asymmetry was created. This arbitrary object, or *arbject*, is described in detail in Appendix C. The Arbject used in this section has the parameters: $\boldsymbol{p}_1 = [0.75\ 1.00]$, $\boldsymbol{p}_2 = \boldsymbol{p}_1 + L[\cos(\theta)\ \sin(\theta)]$, $R = 0.15$, $d = 0.30$, $\omega = 6$, and the characteristic length is $L = 0.75$ with angle $\theta = 30$deg (all lengths are in meters). A simulation was performed for this object using the conditions of Table 3.1 with the appropriate changes for mass properties of the object being $I_{yy} = 111.90$ kg–m^2 about $\boldsymbol{x}_{cm} = [1.1109\ 1.0708]$m. Time histories of states for Case 1 are given in Figure 3.7. An oscillation

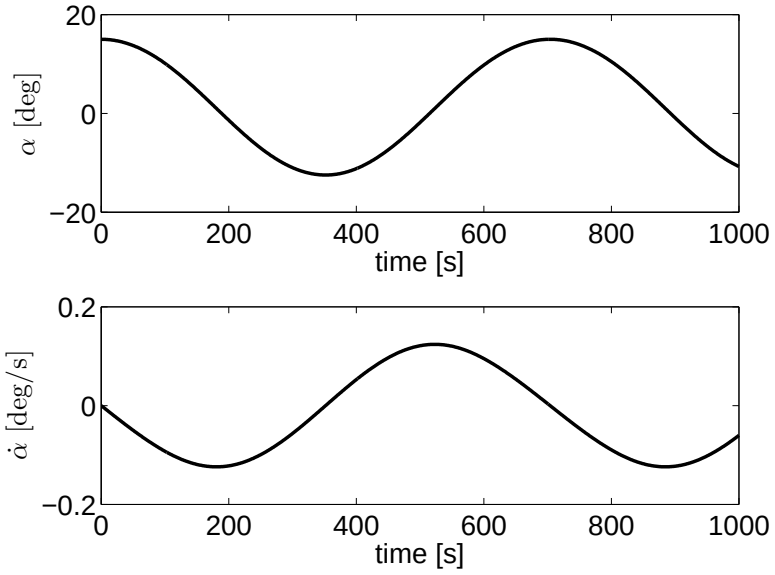


Figure 3.7: State variable time histories for an Arbject using initial conditions of Case 1.

about $\alpha = 0$ once again occurs, despite the geometry being asymmetric. Due to the low magnitude of moment in these FM cases, this motion is expected. However, when a set of high angle initial conditions are used, such as those given in Table 3.3, the Arbject rotates quickly upward in an apparent swift departure from $\alpha = 0$. Its oscillation is still stable, as it eventually returns to its initial state. Whether the Arbject under these conditions has an equilibrium at the origin is an

Table 3.3: Case 4 conditions

Symbol	Value	Unit
α_0	150	deg
$\dot{\alpha}_0$	0.1	deg/s
$\boldsymbol{x}_{\text{cm}}$	[1.1109 1.0708]	m

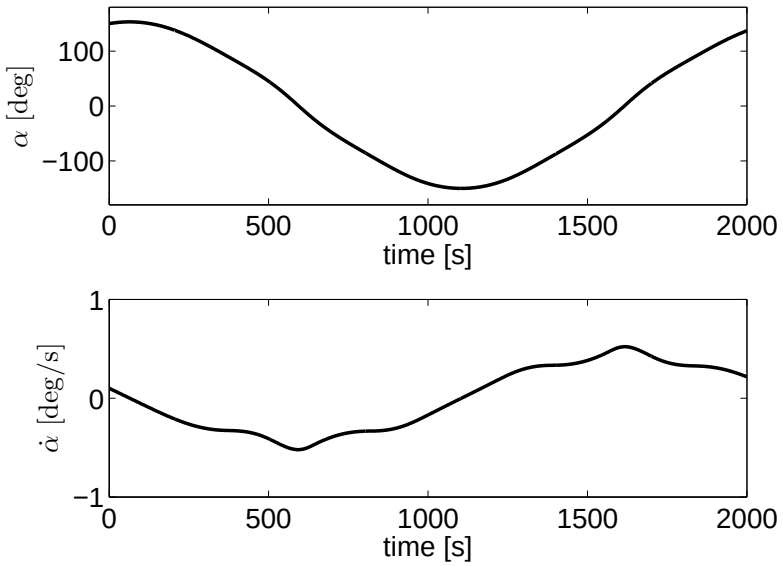


Figure 3.8: State variable time histories for an Arbject using initial conditions of Case 4.

obvious question. With asymmetric geometry, there should be a nonzero moment acting on the body at $\alpha = 0$. The moment source function, given as Figure 3.9 shows the expected asymmetry, and in fact, does indicate that there is a positive moment of 1.66×10^{-4}Nm at $\alpha = 0$, or roughly 5.3% of the maximum absolute value of momentum experienced in the range $[-\pi..\pi]$. This moment is enough to initiate an upward rotation with initial conditions of the origin: $\alpha = 0$, $\dot{\alpha} = 0$. A simulation of this motion is shown in Figure 3.10. A seemingly consistent oscillation occurs at a low amplitude. Thus it appears that the origin in this case is not an equilibrium, but there are trajectories near it that are likely stable limit sets. To examine these asymmetric dynamics in more detail, a phase portrait was constructed for the Arbject under a variety of initial conditions, shown in Figure 3.11. Somewhat similar to the phase portrait of the NACA-0012 airfoil, the Arbject

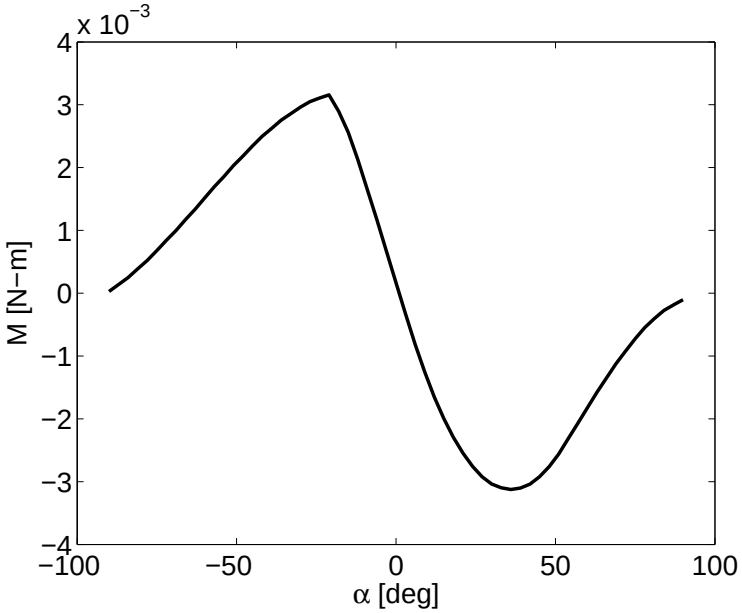


Figure 3.9: Moment source function $M(\alpha)$ of the Arbject with $\dot{\alpha} = 0$.

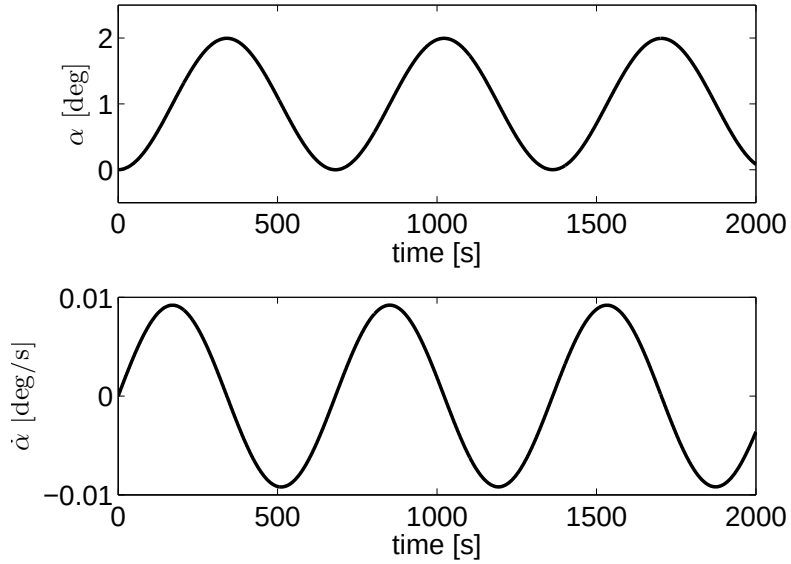


Figure 3.10: State variable time histories for an Arbject using initial conditions of $[\alpha = 0 \quad \dot{\alpha} = 0]$.

appears to have a central region in which stable trajectories exist, centered around $\alpha = 2n\pi$. However, the region is not symmetric about $\alpha = 0$. Unstable trajectories still exist for high initial

values of $\dot{\alpha}$, which is rather unsurprising when considering the lack of energy dissipation in FM flow.

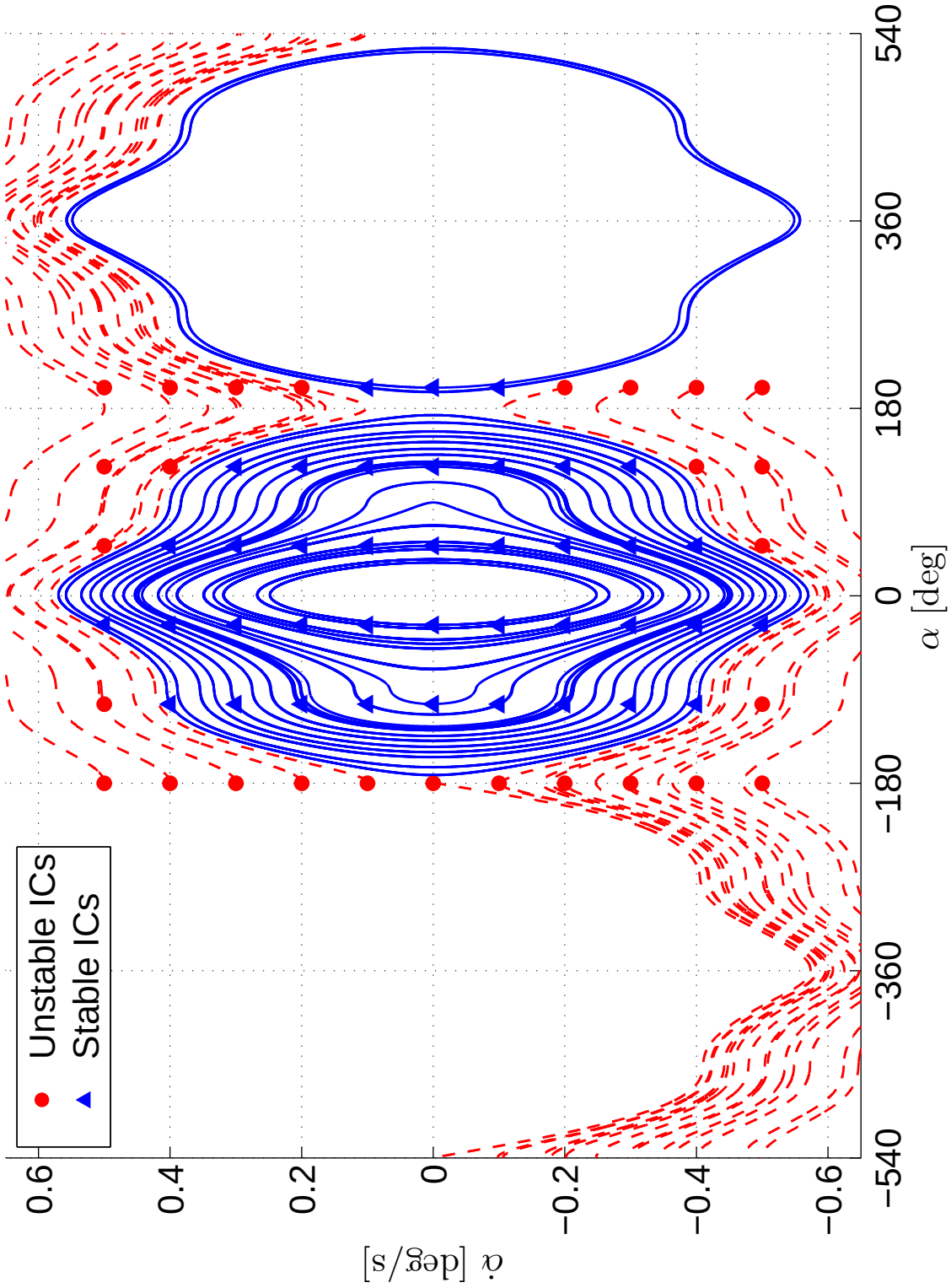


Figure 3.11: Phase portrait for an Arbject under the conditions of Table 3.1. The initial condition (IC) for each trajectory is shown.

3.4 Summary: Initial Dynamics Investigations

Although the examples cases presented here are simply some of many that could be chosen to elucidate the challenges present in understanding the dynamics of rigid bodies subject to rarefied and transition flow, other cases may be examined should the need arise. However, much of the insight to be gained and tools to be developed in rarefied rigid body dynamics require special methods like those presented in Chapter 4. Otherwise, analyses are limited to in-loop computations where source functions are provided in real time during a time integration from DSMC (or equivalent CFD methods), which is impractical.

Chapter 4

Approximation Methods for Dynamics

Aerodynamics in rarefied gas environments typically focuses on the problems involved in understanding how the gas itself affects a rigid-body immersed within it. Less often attempted, however, is the problem of how common bodies (often active spacecraft) respond to the rarefied environment dynamically. The problem of modeling gas forces on a rigid body has a rich history that begins with modeling atmospheric aircraft dynamics. As the forces and moments acting on a body in a continuum fluid are often highly nonlinear, many resulting models employ approximations that often linearize some or part of these forces in some context. For example, classic aircraft dynamics utilizes a single point Taylor series expansion that assumes the aircraft naturally possesses some description of stability about one equilibrium, which is described as the aircraft's "trim" flight. From this, it is also assumed that any deviations from this defined point are small enough such that the equations of motion are effectively linear within a small region surrounding the point. This assumption has proved sufficient in the very specific cases of fixed-wing, classical configuration aircraft at low Mach numbers which are not expected to attempt any large-perturbation maneuvers.

The conditions for linearized models cannot be met for many modern aircraft, and certainly cannot be met for spacecraft. Most spacecraft and orbital debris can make very few, if any, assumptions about their natural dynamics, and even fewer in the rarefied environment. Many actively controlled spacecraft engage in maneuvers that change the craft's attitude by very large angles, rendering any single-point expansion linear dynamics useless.

Thus far, the most common type of modeling of spacecraft dynamics falls under the "satellite

drag" category [16, 17] with emphasis on extracting atmospheric modeling from drag measurements [33, 34, 37, 32]. The goals of this area of research are to predict the drag force on a body moving through a rarefied gas by providing simple models that are functions of the few available inputs, namely cross-sectional characteristic area, body mass, gas conditions such as density and temperature, and possibly body surface properties. The widely-employed ballistic coefficient, which is a function of these variables, is the parameter of interest when looking to characterize the dynamic behavior of spacecraft in low Earth orbit (LEO). Much of this work has been successful at creating models of a one-dimensional nature that are in agreement with available data [49, 58] and often focus on the physics of molecular interactions [57] rather than rigid-body dynamics. However, most of these data and models of drag or ballistic coefficients typically tend toward the goal of predicting very general spacecraft orbital motion such as de-orbit trajectories and changes in orbital parameters [14]. It is obvious, however, that a single parameter such as a drag coefficient cannot possibly model the highly complicated non-linear phenomena that occur as a body moves through a rarefied gas. It is with this hindrance in mind that we look to form some means of approximating the effects of fluid forces on a rigid body that allows a wide range of states to be occupied by the body while retaining some as-yet undefined description of accuracy. The goal would be to provide a single set of constants for arbitrary geometry, mass properties, and flight conditions like Kn and Ma that can used as inputs to a simulation as is illustrated in Figure 4.1. The simplest way of providing this set of data is in the form of a look-up table–a version of which is presented in this chapter as a baseline for future comparisons with better methods. Possessing high-fidelity models of spacecraft dynamics can yield dramatic increases in capability of design and operation of spacecraft, and better predictions of natural motion of debris or other inactive objects much in the same way as high-performance aircraft are made possible by better non-linear plant and control models. With the advent of modern computational tools such as DSMC, we can now provide input to rigid-body dynamics models of rarefied flow systems much in the same way as we can with classical computational methods such as numerical Euler and Navier-Stokes to continuum fluid systems.

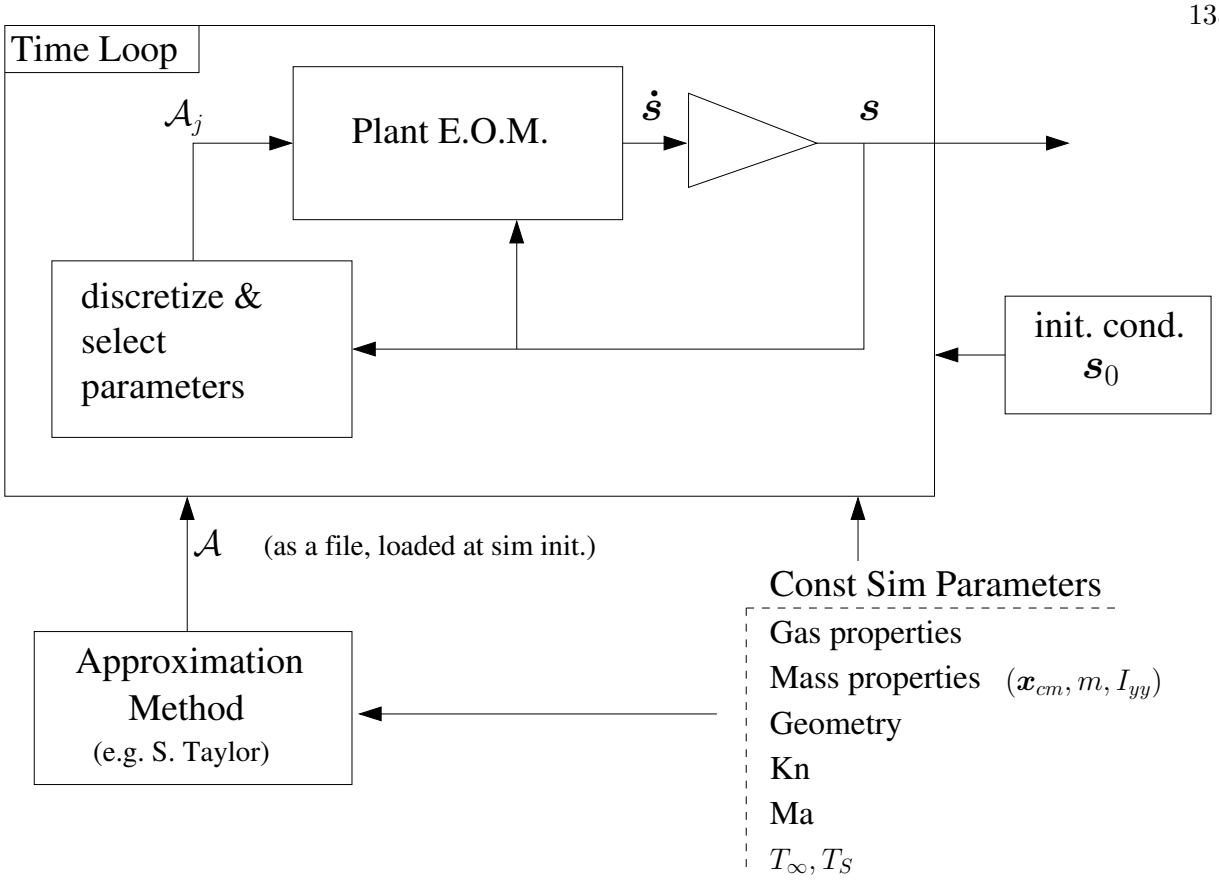


Figure 4.1: A block diagram that illustrates the logical location of an approximation method. Set $\mathcal{A}$ is defined in Section 4.1.1.

4.1 Introduction to Approximation Methods

The first approximation model to be examined is the single-point Taylor series expansion. It is clear that choosing a single point in state space to serve as the pseudo-equilibrium about which linear perturbations are assumed to occur is insufficient for the highly non-linear dynamic behavior expected of spacecraft and orbital debris. This is evident in the Figure 3.2. Considering how quickly $M(\alpha)$ changes about $\alpha = 0$, it is clear that no linear expansion about this point is valid for the entire range of α. Extending the order of the expansion to include higher-order truncation error, for example a 2nd-order Taylor series about $\alpha = 0$, might extend the valid range of the approximation but would also incur significant error for any source function that is not, at most, roughly quadratic.

Implementation of higher-order Taylor series also requires considerable effort with little return in accuracy, particularly beyond second order. A more robust method is required to capture the full range of α that does not rely on a single point of expansion and is overall independent of the source functions it uses as input. In order for this "black box" method to be viable, it should work for general cases regardless of the type, shape or otherwise "quality" of source functions. This requirement is particularly important since the case may often arise where the source functions contain noise or high frequency oscillations, discontinuities or other undesirable properties that make approximation difficult. It also enforces adherence to the general goal of dynamic simulation without knowledge of the fluid-mechanical effects on a given object.

4.1.1 Variable-Structure Model

The approximation model for the following analysis is the Variable Structure System (VSS) or "sliding" model[71]. A VSS model is one in which a different description of a system is used at different points in state space. It is typically applied in the form of Variable Structure Control rather than in the system's natural dynamics and much work in this area has involved determining proper control definitions and switching law[18]. The VSS model can be applied to form an approximation that can meet the aforementioned requirements. This application is essentially a table look-up method that is customized to include information required by the specific formulation of the dynamics presented in this section.

The VSS formulation begins with the single-point Taylor expansion about an arbitrary point $\boldsymbol{s}_j$ (the multi-dimensional index j refers to any arbitrary point in the state space in which $\boldsymbol{s}$ can exist). When truncated after the linear term it is written as

$$\dot{\boldsymbol{s}}(\boldsymbol{s}) \approx \dot{\boldsymbol{s}}(\boldsymbol{s}_j) + \mathcal{J}(\dot{\boldsymbol{s}})\Big|_{\boldsymbol{s}_j} (\boldsymbol{s} - \boldsymbol{s}_j) + \mathcal{O}(\boldsymbol{s}^2) \tag{4.1}$$

where $\mathcal{J}(\dot{\boldsymbol{s}})|_{\boldsymbol{s}_j}$ is the Jacobian matrix of the non-linear vector field $\dot{\boldsymbol{s}}$ with respect to the state space, evaluated at the point $\boldsymbol{s}_j$. The dynamics of Equation 3.2 are chosen, and the operations are performed to yield a set of linearized equations of motion with expansion offsets intact. Dropping the

error term (and replacing $\approx$ with $=$ since it should be clear that the following are approximations), the first equation is expanded as

$$\dot{s}_1 = \dot{s}_1(\boldsymbol{s}_j) + \left.\frac{\partial \dot{s}_1}{\partial \alpha}\right|_{\boldsymbol{s}_j} (\alpha - \alpha_j) + \left.\frac{\partial \dot{s}_1}{\partial \dot{\alpha}}\right|_{\boldsymbol{s}_j} (\dot{\alpha} - \dot{\alpha}_j) \tag{4.2}$$

with the other treated similarly. Each equation contains three constant terms and two terms that are products of constants and state variables. The offset constants can be evaluated as

$$\dot{s}_1(\boldsymbol{s}_j) = \dot{\alpha}_j \tag{4.3}$$

$$\dot{s}_2(\boldsymbol{s}_j) = \frac{M_j}{I_{yy}} \tag{4.4}$$

The state variables $\boldsymbol{s}_j = [\alpha_j \ \dot{\alpha}_j]^{\mathrm{T}}$ are the discretized continuous variables. This is discussed in detail in Section 4.1.1.1. After algebraic manipulation, the linear system that represents the dynamics at any point $\boldsymbol{s}_j$ is given as Equations 4.5 and 4.6 with state-specific parameter constants defined in Equations 4.7 to 4.9.

$$\dot{s}_1 = \dot{\alpha} = \dot{\alpha} \tag{4.5}$$

$$\dot{s}_2 = \ddot{\alpha} = \frac{1}{I_{yy}} \left(M_{\alpha j}\alpha + M_{\dot{\alpha}_j}\dot{\alpha} + k_j \right) \tag{4.6}$$

$$M_{\alpha j} = \left.\frac{\partial M}{\partial \alpha}\right|_j \tag{4.7}$$

$$M_{\dot{\alpha} j} = \left.\frac{\partial M}{\partial \dot{\alpha}}\right|_j \tag{4.8}$$

$$k_j = M_j - M_{\alpha j}\alpha_j - M_{\dot{\alpha} j}\dot{\alpha}_j \tag{4.9}$$

These simple equations capture the dynamics over a region of state space centered at point $\boldsymbol{s}_j$. When the state changes, it is examined to determine in what discrete region it currently resides (i.e. at which value of j). The appropriate set of parameters is then selected and used to evaluate the equations of motion. These constant parameters are similar to the linear stability derivatives seen in classic aircraft dynamics. Though, rather than being a single set of constants, they are a collection of sets of constants for each discrete region of state space, indexed by j. The collective of

sets of dynamic parameters is denoted by super-set $\mathcal{A}$. An extraction of a subset from this super-set for a point $\boldsymbol{s}_j$ is denoted with j-subscripts. This notation is used in Figure 4.1.

Employing the sliding dynamic model is akin to approximating all source functions with piecewise linear functions over the discrete regions centered at each point $\boldsymbol{s}_j$. Figure 4.2 illustrates this concept for an arbitrary source function. The state space is represented as a set of subsystems, each described by the set of parameters: $\mathcal{A}_j = \{M_{\alpha j}\ ,\ M_{\dot{\alpha} j}\ ,\ k_j\}$. Sliding over the state space presents a few possible problems. One problem is that (in this formulation) there is no consideration for how one subsystem makes a transition to another neighboring subsystem. Another problem is that to properly capture any high frequencies or stiffness in the source functions, parameters may need to be calculated at exceedingly high resolutions. Although all parameters can be programmatically stored and extracted, reducing the number of constants required to represent a source function is desirable.

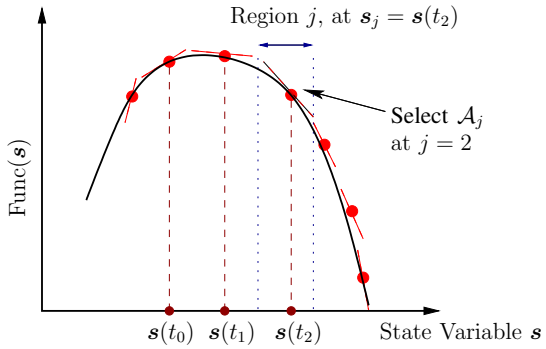


Figure 4.2: An illustration of Sliding Taylor dynamics showing subsystems covering discrete regions of the state space.

4.1.1.1 Discretization Methods

Sliding Taylor parameters must be calculated and stored for each point in a discretized state space. The resolution of this space determines how many parameters are required to describe the system. For example, the source functions of $M(\alpha)$–for each Ma -value–in Figure 3.2 are each evaluated at sixty-one points, including the end points at $\alpha = \pm 90$ deg. If a source function has

high frequency oscillations or high gradients, the discrete space may require higher resolution in these regions, if not the entire domain. It is important to make the distinction between the actual state space, which remains continuous and the discrete, approximated state space. The dynamic formulation makes no specific consideration for the current approximated state. The appropriate set of parameters are selected and injected into the dynamics at which point the equations of motion treat the system they have adopted as being identical *in form* to that present at any and all previous states.

The set of parameters corresponding to the current state is chosen by finding the multi-dimensional or flat index j of the state $\boldsymbol{s}$ through the following series of operations:

$$\begin{aligned} \boldsymbol{s}_{e,\min} &= \boldsymbol{s}_{\min} - \frac{\Delta \boldsymbol{s}}{2} \\ \boldsymbol{s}_{e,\max} &= \boldsymbol{s}_{\max} + \frac{\Delta \boldsymbol{s}}{2} \end{aligned} \tag{4.10}$$

$$s_k^{\text{ind}} = \text{fix}\left(\frac{s_k - s_{e,\min,k}}{\Delta s_k} \right) + 1 \tag{4.11}$$

$$j = (s_1^{\text{ind}} - 1) n_{\dot{\alpha}} + s_2^{\text{ind}} \tag{4.12}$$

where k is a free index of the state vector $\boldsymbol{s}$, $\Delta \boldsymbol{s}$ is the spatial resolution of the approximated state space (e.g in Figure 3.2 it is $\Delta\alpha = (\alpha_{max} - \alpha_{min})/(n_\alpha - 1) = 180/60 = 3\,\text{deg}$), $n_{\dot{\alpha}}$ and n_α are the number of points in the $\dot{\alpha}$ and α directions, respectively, of the discrete state space and the `fix()` function is an integer rounding towards zero. The terms $\boldsymbol{s}_{\min}$ and $\boldsymbol{s}_{\max}$ in Equations 4.10 are the extreme values of the state space defined by the user ($s_{\min,2} = -90\text{deg}$ and $s_{\max,2} = 90\text{deg}$ for the $M(\alpha)$ example, if α is the first state variable). The effective state $\boldsymbol{s}_e$ is defined because each point in the discrete space represents the center of its viable region, as shown in Figure 4.2. This definition allows the end points of the space to govern space that is actually slightly outside of the defined domain, rather than simply truncating each end point. Note that Equation 4.12 is specific to the two-state example used here and assumes 1–indexing. It also assumes a certain loop or packing order of the parameters–one in which the inner-most loop is over $\dot{\alpha}$, the next outer being over α.

It is important to note that, as with most table look-up methods, j will be out of range of available parameter data if the real state happens to leave the pre-computed range of discrete state space. Angles such as α alleviate the problem of choosing a suitable range over which to compute parameters, as they naturally define a periodic domain. Thus, when calculating the source functions, one needs only to define α over the range $[0, 2\pi]$ for all possible simulations involving α. The problem of defining a suitable range for non-periodic states such as $\dot{\alpha}$ still remains; however, dynamics depend most crucially on α for this example. Also, it is reasonable to expect scalings and characteristic constants will be available in most simulations such that choices of discrete state-space domains will not prove exceedingly difficult to make.

4.2 Application Results and Discussion

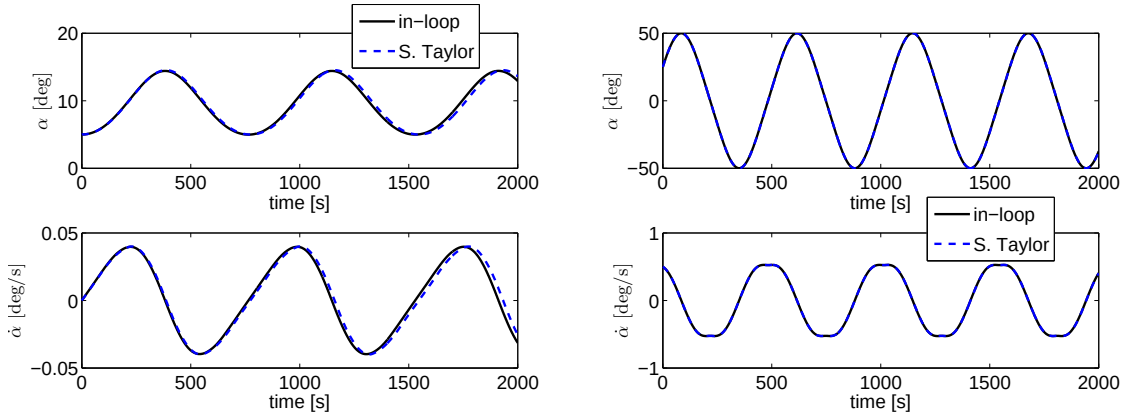
Using a 1m chord NACA-0012 2D airfoil section, the Sliding Taylor approximation is applied and integrated under the same conditions of Table 3.1. The quality of the new, approximated dynamics are compared to the FM in-loop baseline of Section 3.3. The objective of such a comparison is to determine the effectiveness of the ST method while it easy to generate comparison simulation data due to the baseline being in-loop. DSMC is then used to generate realistic source functions for the airfoil under transition flow conditions of Kn = 1, corresponding to a circular orbit at an altitude of approximately 109.5km. These data are then processed with ST to create a coefficient set (a single file) for the airfoil under these conditions. This file is used as an input (as indicated in Figure 4.1) to simulate transition flow dynamics.

4.2.1 Sliding Taylor Demonstration and Comparison

A comparison of Case 1 was made for the Sliding Taylor approximation method. Conditions were identical to those stated in Table 3.2 with the addition of discrete state space resolutions

$$\boldsymbol{s}_j \in \{[-180, 180] \text{ deg}, [-5, 5] \text{ deg/s}\} \tag{4.13}$$

with $\Delta\alpha = 4\text{deg}$ and $\Delta\dot{\alpha} = 1\text{deg/s}$. Figure 4.3a shows the state variable time histories of the approximation method compared to the baseline in-loop method, which is considered to be the reference or true result to which all future approximations for free-molecular flow are compared. Cases 1 was chosen due to its initialization within the divergent region and Case 4 was chosen for its larger angle and non-zero of initial angular momentum. The most serious departure in the

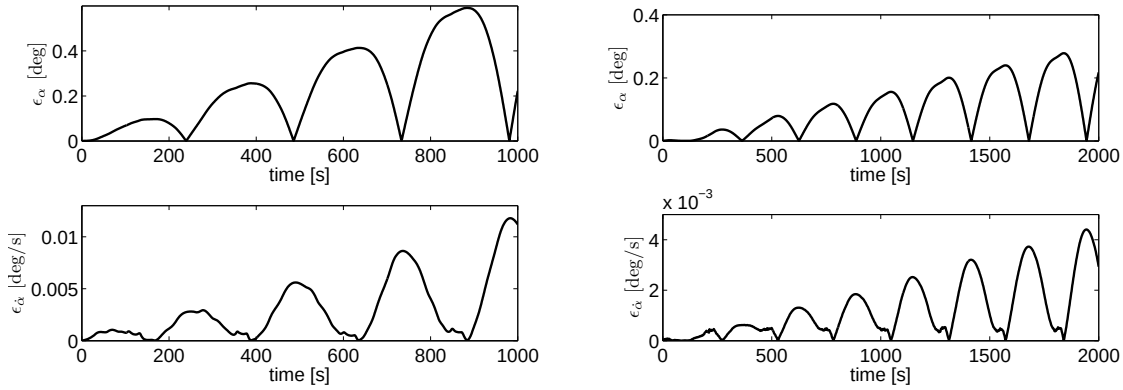


(a) Case 1.

(b) Case 4.

Figure 4.3: State-variable time histories of the Sliding Taylor approximation method compared to the baseline in-loop simulation for free-molecular flow.

approximated method occurs later in the evolution of the system where it appears some numerical drift has occurred. This error is most prevalent for Case 1 but not visible for Case 4. Some of this drift may be related to the relatively low resolution of discrete state space or small numerical error that slides the state into an incorrect neighboring discrete region, therefore using that region's set of parameters rather than the nominal set. Both of these possible problems should diminish as discrete state space parameter resolution increases. This convergence is examined in Section 4.2.2. The actual relative error for each state variable as a function of integration time is presented in Figure 4.4. Absolute error was used instead of relative error since as the body rotates past $\alpha = 0\text{deg}$, relative error becomes infinite. The drift that can be seen to occur does not amount to an absolute error of more than one degree in amplitude of oscillation of α over the course of two oscillations in either Case 1 or 4.



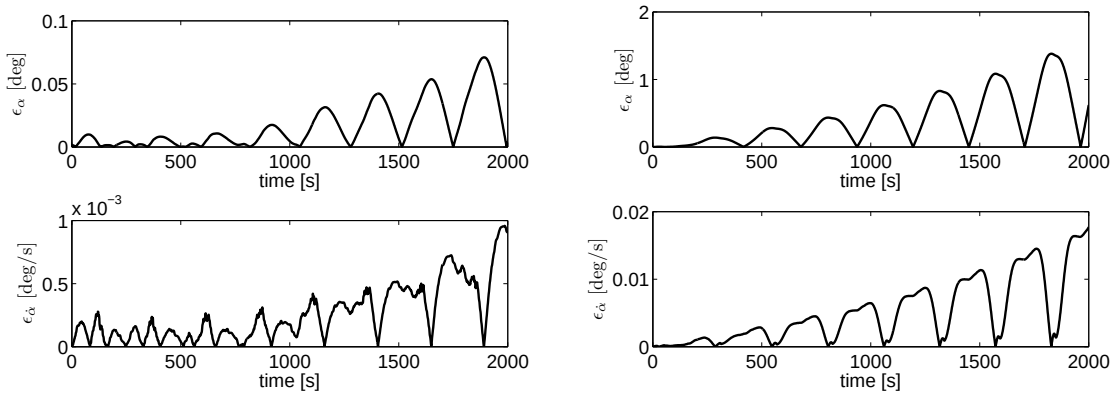
(a) Case 1. (b) Case 4.

Figure 4.4: Time histories of state-variable absolute error between the Sliding Taylor method and baseline in-loop simulation.

The initial conditions of cases in Table 3.2 contain moderate initial angles of attack. The method's robustness is examined by simulating at a higher initial angle of $\alpha_0 = 65$deg with $\dot{\alpha}_0 = 0$deg/s and $\dot{\alpha}_0 = -1.0$deg/s. The specific means of calculating the constant parameters in the Sliding Taylor method allow the simulation to be performed at any initial conditions that fall within the discrete state ranges of Table 3.2. However, these cases still do not have initial conditions outside of the stable region shown in Figure 3.6a. The error of the case in Figure 4.5a indicates a more accurate approximation while the error of the case in Figure 4.5b indicates less accuracy. The latter case's initial angular movement may be the cause of this, with drift occurring due to the equations of motion being initially stiffer. In all cases, the error appears to be periodic in time. Phase shift from numerical integration drift is evident which causes the error to be zero at times where the solutions for each method intersect.

4.2.2 Source Data Resolution Convergence Study

The most obvious problem with the Sliding Taylor method of approximation is that a great number of constants are required to represent a general set of conditions. Knowing how fine a resolution is necessary to achieve a desired level of accuracy is useful for avoiding unnecessary



(a) Case of $\alpha_0 = 65$deg for $\dot{\alpha}_0 = 0$deg/s.

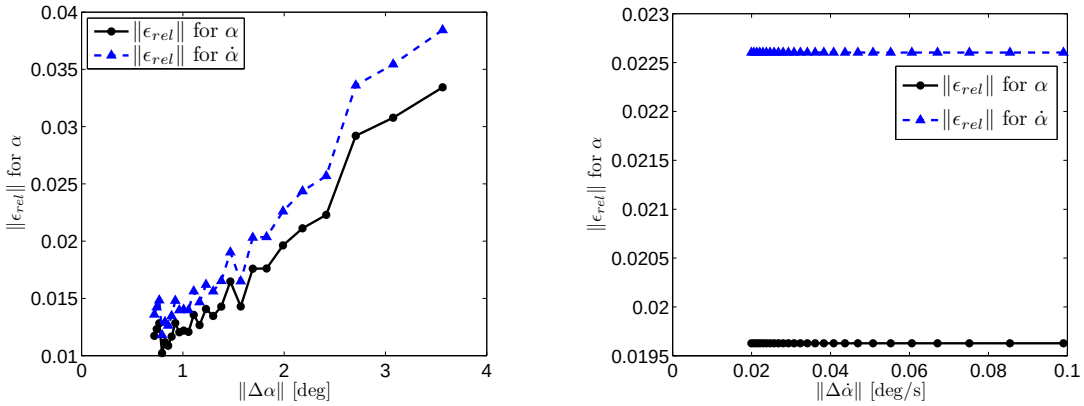
(b) Case of $\alpha_0 = 65$deg for $\dot{\alpha}_0 = -1.0$deg/s.

Figure 4.5: Time histories of state-variable absolute error between the Sliding Taylor method and baseline in-loop simulation for high-angle cases.

computations. Furthermore, it must be confirmed that increasing the resolution of the state space discretization actually leads to improvements in approximation accuracy. An error convergence study was performed for each state variable. Sets of data were created wherein the discrete space resolution of one state variable was varied while the other was kept constant. These sets were then integrated in time using the conditions of Case 1 for 10s. For each resolution of a data set, the $\mathcal{L}^2$–norm of the relative error between resulting state histories (for all time) for in-loop and Sliding Taylor results was computed using Equation 4.14 (where k is a free index of the state vector, $\tilde{s}_k$ is the Sliding Taylor state result and noting that s_k represents the state k for all time over which it was integrated).

$$||\epsilon_{\text{rel},k}||_2 = \frac{||\tilde{s}_k - s_k||_2}{||s_k||_2} \tag{4.14}$$

The reduction in error norms as a function of the resolution of α is apparent but not particularly rapid. Convergence appears to be roughly linear with respect to the resolution of α to within 1.5% for 1 deg/s resolution. There is virtually no improvement in resolution of $\dot{\alpha}$, which would be expected as the dynamics depend very little on angular rate compared to the significant sensitivity to the angle itself (i.e. $M_{\dot{\alpha}} \ll M_\alpha$). This is primarily because the rates of rotation seen in this study, with this geometry, never contribute to an induced linear velocity at the airfoil surface that



(a) Resolution of α ($\Delta\dot{\alpha} = 0.0671$ deg/s)

(b) Resolution of $\dot{\alpha}$ ($\Delta\alpha = 1.9890$deg)

Figure 4.6: Relative error between in-loop and Sliding Taylor simulations under the same conditions of free-molecular flow, as a function of discrete space resolution.

is comparable to the characteristic or stream velocity. It would require $\dot{\alpha}$ values on the order of 10^5 deg/s to yield similar magnitudes of induced velocity. Additionally, it is questionable whether the transpiration model of Section 3.2.1 would be sufficiently accurate at such rates.

While the L2 norm quantifies some measure of accuracy, it does not provide a measure for relative phase shifts in the solutions from the two models. Although Figure 4.6a may indicate that relative error remains below 5% for angular resolutions of up to 4deg, cumulative numerical or discretization error may still result in the type of drift seen in Figures 4.3a and 4.3b.

4.2.3 Comments on Sliding Taylor accuracy

The preceding section provides empirical evidence of convergence. The following are additional observations regarding the method's accuracy for arbitrary geometry and flight conditions. The first problem of interest is the possibility that the linear system that describes motion about each discrete point in the specified state space might be locally asymptotically stable (or generally unstable). Certain conditions may exist such that when the global system moves into a stable region, the state becomes trapped. This occurrence would artificially halt the body's motion, leading to incorrect conclusions about the global dynamic response. No artificial stability was observed in

the examples presented in the preceding section and would be unlikely to occur for such simple, symmetric geometry as a NACA-0012 airfoil. For more complicated and asymmetric geometries, this undesirable "locking" stability may be more plausible, creating a need to examine the global dynamics in more detail when using piecewise methods to ensure global consistency.

Another consideration is approximations that use source functions from a CFD method such as DSMC. The concern is that a DSMC simulation might not adequately simulate the feedback of the motion of the body on the surrounding flow field. This coupling would occur through rate variables such as $\dot{\alpha}$. For the airfoil example, when it rotates upward, changing α over time, a new steady-state simulation is performed at this new point. No consideration is made for the fact that the body has "pushed" gas out of its way as it rotated. Only a true coupled-field simulation, where DSMC is indeed in-loop, and the body geometry is altered each time step as it is allowed to move would capture this feedback. This is certainly true for $\dot{\alpha}$, which would require a moving-body DSMC simulation and would naturally eliminate the possibility of using a steady-state sampled computation as described here. However, as is often the case for rigid-body dynamics, it is assumed that time-integration of the body's equations of motion occurs at a sufficiently high temporal resolution such that source functions are, subsequently, assumed to change little over the time step of the integration.

4.3 Dynamics in realistic contexts

This section briefly examines the dynamic response of the NACA-0012. Source functions generated both the free-molecular panel method and from a set of DSMC simulations are used to produce dynamic rigid-body simulations for flows ranging from free-molecular into the transitions regime.

4.3.1 Frequency Study

The NACA-0012 airfoil is used to study how the mass properties and stream conditions affect the dynamic response of a spacecraft. The airfoil is assumed to be a solid body of a homogeneous

material of unit depth with mass density ρ. The initial conditions of $\alpha_0 = 15$deg and $\dot{\alpha}_0 = 0$ deg/s induce a stable dynamic oscillation about α_0. A series of simulations was performed for a range of ρ, which directly scales the total body moment of inertia I_{yy}, and for a small set of Mach numbers, which also contribute to the magnitude of the source functions. Each simulation results in a state variable time history from which the period of the dominant oscillation of any chosen state can be determined. The result of this frequency/period study is shown in Figure 4.7 for α. Values of Ma = 10 and Ma = 11 were chosen because of a shift in dynamic response between these

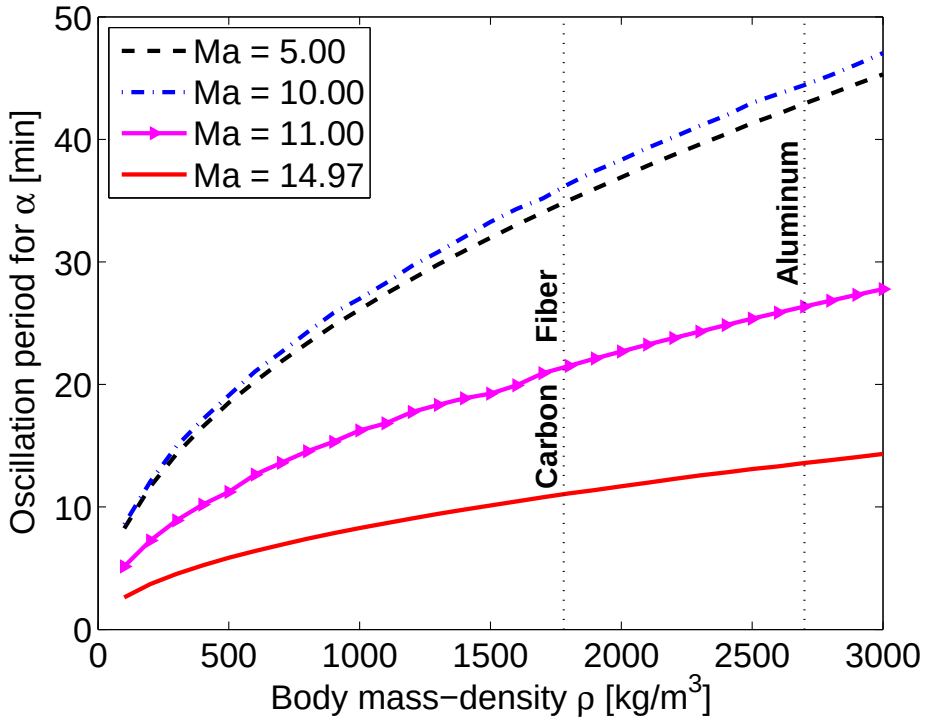


Figure 4.7: The oscillation period of the dominant mode in the dynamic response of the state variable α for a homogeneous NACA-0012 airfoil in free-molecular flow is shown as a function of body mass-density for three Mach numbers.

values. The oscillation changes from occurring about one equilibrium (i.e. α^*, from Figure 3.6) to oscillations about $\alpha = 0$. The value of Ma = 14.97 corresponds to a 155.5km-altitude circular orbit; with the assumption of a single-species N_2 atmosphere, Kn = 100 for an airfoil with a 1m chord. The magnitude of free-molecular source functions is often assumed too low to be of consideration

in dynamics, yet even at a very low Mach number of 5.0, the airfoil with the constant mass density of aluminum experiences an oscillation with a period of approximately 40 minutes. Since most orbits for which the free-molecular model is accurate have an orbital period of at least 90 minutes, a 40-minute period suggests that a spacecraft with airfoil-like geometry may experience significant attitude changes from gas forces alone over the course of a single orbit. As orbital altitude decreases, Ma increases and Kn decreases causing forces and moments acting on a body to also increase, which in turn creates faster oscillations.

4.3.2 Transition regime dynamics with DSMC

It is important to note that in-loop generation of source functions using DSMC is currently computationally impractical. Approximation methods such as those outlined in Sect. 4.1 provide a means to make the simulations tractable. A method such as Sliding Taylor provides an effective "black box" functional approach where dynamic coefficients $\mathcal{A}$ can be computed from DSMC-generated source functions. If DSMC simulations are performed in the transition regime, the approximation method will use transition-regime source functions, and the result will be the ability to perform rigid-body dynamic simulations in the transition flow regime without running DSMC in-loop.

A piecewise Taylor decomposition is not necessary to achieving this goal. Source functions may be computed at a high resolution and loaded at the time of simulation, in-loop. States that fall between computed data points may be linearly interpolated. Nevertheless, approximations like Sliding Taylor, Fourier series, or direct interpolations of higher order than linear can provide a means of analyzing the detailed behavior of the dynamic system rather than just simulating it. For spacecraft which possess a trim state, the analogy to aircraft is highly appropriate. In this case, linear stability derivatives provide an excellent description of the system, and may allow identification of characteristic modes of operation. Additionally, control schemes may be considered that take into account aerodynamics when the dynamic system is closed-form.

4.3.2.1 Generation of transition regime source functions

Source function generation with DSMC requires that a fluid simulation be performed for each point $\boldsymbol{s}_j$ in the targeted, discrete state space. For this study, a relatively low state space range and resolution in each direction was chosen to make the task of source function generation tractable. The ranges of the states and resolutions in Table 4.1 are chosen with the expectation that the transition dynamics of the NACA-0012 will be qualitatively similar to the free-molecular cases. The highest resolution is in α, since moment is expected to show the strongest sensitivity to α.

Table 4.1: Discrete state-space ranges for DSMC source function generation.

State var.	Range	Num. points	Resolution
α	$[-90, 90]$ deg	61	3 deg/s
$\dot{\alpha}$	$[-15, 15]$ deg/s	16	2 deg/s

These resolutions give 976 points to be simulated in DSMC in total. Flow conditions are given in Table 4.2 for a N_2 atmosphere to simulate transition-regime conditions for a 1m object in a 109.5km orbit with Kn = 1. A variable-adaptive time-step method is implemented based on the smallest local mean collision time in the domain. Domain geometry and boundary conditions are illustrated in Figure 4.8. The DSMC code used is Voldipar. Each simulation is run to a time at which the flow is determined to be adequately steady, then continued as a cumulative time-sample. This is the standard method of performing steady-state calculations for DSMC[3, 2]. The imposition of each discrete state space point $\boldsymbol{s}_j$ is made by first rotating the airfoil body through the appropriate angle of attack α, then to simulate angular rate $\dot{\alpha}$, the transpiration velocity method of Section 3.2.1 is applied.

The selection of appropriate DSMC resolutions requires cell size/number, number of simulation particles[1] , geometry resolution, and time (length of the simulation to steady state and to completion). Voldipar uses a voxel (or pixel in 2D) rasterization of the geometry. Resolutions are specified to minimize run time with acceptable solution accuracy. These choices of resolution

[1] The actual number of DSMC simulation particles is not specified, but rather the parameter F_{num}, the ratio of real-to-simulated particles. This is the actual independent variable for the study of particle number resolution as it directly determines the number of particles in the domain.

Table 4.2: DSMC conditions simulating transition flow in a 109.5km circular orbit for a 1m chord NACA-0012 airfoil in N_2 with gas reference temperature of 273 K and a fully diffuse gas-surface interaction model.

Variable	Symbol	Value	Unit
Mach number	Ma	24.18	
Knudsen number	Kn	1	
Stream temperature	T_{in}	252.9	K
Body surface temperature	T_S	300	K
Domain length	L_x	2.0	m
Domain height	L_y	2.0	m
Num. subcells per cell (each dir.)	N_{sc}	2	
Num. time steps per sample	N_{sam}	4	
Inflow number density	n_{in}	1.2944×10^{18}	m^{-3}
Inflow speed (aligned with +x)	V_{in}	7838.4	m/s
Gas molecular mass	m_g	4.65×10^{-26}	kg
Gas diameter	d_g	4.17×10^{-10}	m
Gas ratio of specific heats	γ	7/5	
Gas constant	R	296.8	kJ/(kg–K)

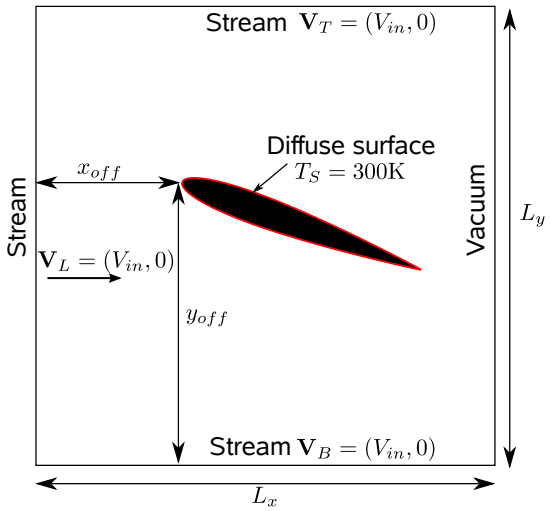


Figure 4.8: Geometry and boundary conditions for DSMC computation of source functions for a NACA-0012 airfoil.

were made by performing a convergence study for the code in the context of the current problem's geometry and flight conditions which can be found in Appendix A. The values for each resolution that were used for performing the parameter scan are given in Table A.2.

Transition-regime source functions were generated by performing all 976 DSMC runs and

calculating the aerodynamic moment for each. A slice of this parameter space that shows moment function dependence on α for $\dot{\alpha} = 0$ is shown in Figure 4.9. The curve in Figure 4.9 is similar

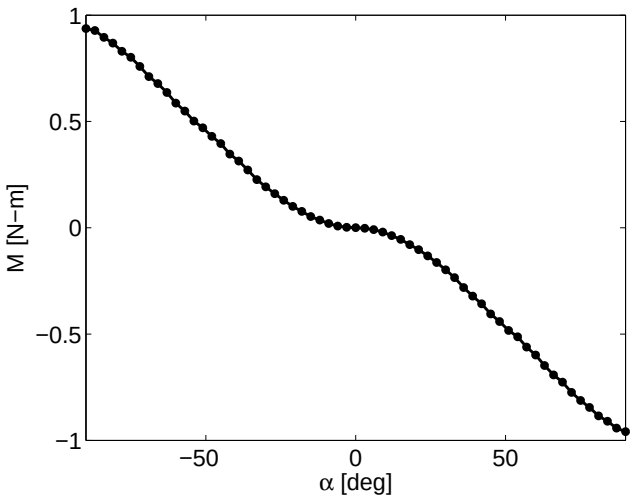


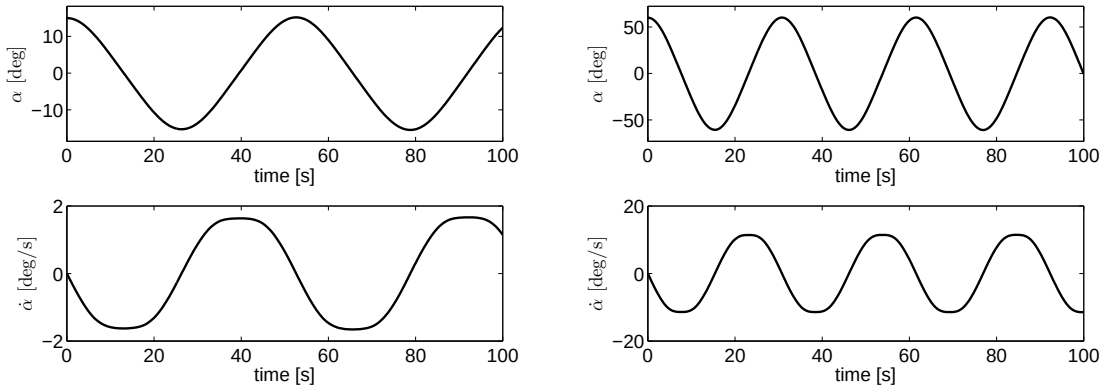
Figure 4.9: Moment source function for the airfoil in N_2 calculated with DSMC for Kn =1, Ma =24.18, and $T_\infty = 252.9$K over the discrete sub-spaces: $-90\,\text{deg} \leq \alpha \leq 90\,\text{deg}$, $\dot{\alpha} = 0$.

to those in Figure 3.2 for the free-molecular analytical method except that it appears to lack the small, central divergent region. This function, obtained from DSMC, is roughly two orders of magnitude greater than the free-molecular case at the same Mach number. This is a reasonable result considering Kn of the transition function is two orders of magnitude smaller.

4.3.2.2 Transition regime rigid-body dynamics

The computation of source functions for a small set of points in a bounded, discrete state space allows the approximation of rigid-body dynamics with the Sliding Taylor approximation method. Using the transition source functions as input to the Sliding Taylor method, sets of coefficients $\mathcal{A}$ were computed with a body mass-density of 2700 kg/m^3 (aluminum), which correspond to a moment of inertia of $I_{yy} = 12.43$ kg–m^2. Time-integrations were then performed to examine the dynamics of the airfoil for the conditions in Table 4.2. State variable time histories are shown in Figure 4.10a for the initial state $\boldsymbol{s}_0 = [15\text{deg} \quad 0\text{deg/s}]^{\text{T}}$. A case with much higher α_0 is shown

in Figure 4.10b to demonstrate the ability of the approximated state space to extend into non-linear dynamics. The resulting response for this set of conditions is a similar dynamically stable



(a) Case of $\alpha_0 = 15$deg for $\dot{\alpha}_0 = 0$deg/s.

(b) Case of $\alpha_0 = 60$deg for $\dot{\alpha}_0 = 0$deg/s.

Figure 4.10: State variable time histories for Sliding Taylor approximation dynamics of transition flow.

oscillation about $\alpha = 0$ as was observed in the free-molecular results, but with a higher frequency. The oscillation period of α is approximately 53s in Figure 4.10a and 31s in Figure 4.10b, which contrasts with the free-molecular case (Ma =14.97, $T_{in} = 655.1K$, 109.5km altitude) from Figure 4.7 where the period is approximately 11min.

This particular choice of initial conditions and time span is somewhat arbitrary, as more detailed dynamics may become apparent when the system response is observed at longer times and at various initial conditions. This behavior would typically be visible in a phase-space diagram similar to Figure 3.6. However, the range of available states for the transition dynamics was not nearly as wide as the free-molecular case because of the computational constraints. It is also difficult to visualize state trajectories for the transition flow case since very long integrations must be performed before any deviation from the canonical, symmetric oscillations can be observed. Some insight into the long-term stability behavior of the transition flow case can be defined in the form of an approximate amplitude profile function ϕ. This function is determined by scanning the time history of a given variable that is assumed to oscillate, such as α, and taking the extrema as

their own "profiling" function. More formally, this is the discrete, discontinuous function $\phi(x) = x(t_k),\ k = [1, 2, 3...]$ for some variable x that oscillates in time, where t_k is the set of points in time where $\partial x(t)/\partial t = 0$. Taking only the odd solutions ($k = [1, 3, 5...]$) yields one "half" of the original function $x(t_k)$, and taking the even solutions yields the other "half". Figure 4.11 illustrates the definition of ϕ in the sense of "peaks only", which is used when $\alpha_0 \geq 0$. The other (negative) half of ϕ is used when $\alpha_0 < 0$. Multiple integrations over a range of initial conditions (in this

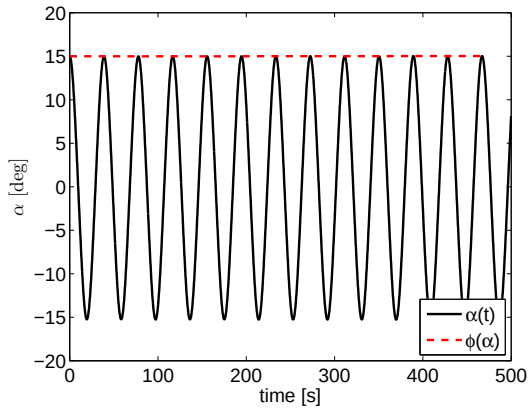


Figure 4.11: Illustration of the definition of the approximate amplitude profile function ϕ for state α, deemed $\phi(\alpha)$.

case, the variation is restricted to α_0) can then be used to gain a picture of long-term dynamics. Figure 4.12 shows transition dynamic response amplitude profiles for various initial angles of attack, with $\dot{\alpha}_0 = 0$.

One to two hours of evolution of the simulation must occur for changes in the response to become observable. Values of $\alpha_0 \gtrsim 70$deg cause angular rates greater than the range of available computed data (see Table 4.1) and so were not examined. At least two limit sets appear to be present: one at $\alpha \approx |18.5\text{deg}|$ and another at $\alpha \approx |35\text{deg}|$. Others may be present near $\alpha \approx |65\text{deg}|$ and $\alpha \approx |75\text{deg}|$. Due to the tendency of amplitude profiles of simulations that start at α values between these values to converge to them eventually, it may be acceptable to describe these angles as locations of limit cycles, as each represents a stable orbit in state space which is isolated from neighboring orbits. The initial state of $[\alpha_0 = |0\text{deg}|\ ,\ \dot{\alpha}_0 = |0\text{deg/s}|]$ is an equilibrium (though not

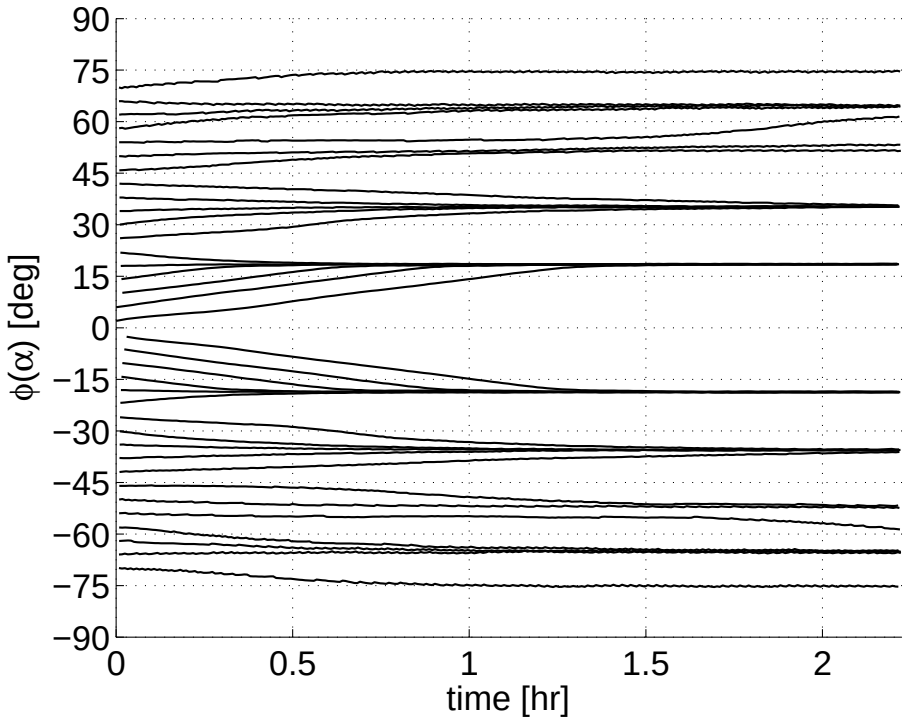


Figure 4.12: Approximate amplitude profile functions $\phi(\alpha)$ for a realistic body density of $\rho = 2700\,\mathrm{kg/m^3}$ for various initial angles of attack at $\dot{\alpha}_0 = 0$ (Note that $\phi(\alpha_0) = \alpha_0$ by construction of ϕ).

plotted in Figure 4.12). It is difficult to determine which initial conditions may create unstable trajectories here as the entire periodic range of α was not computed. With larger coverage of the discretized state space, more dynamic insight would be possible as more information becomes available.

4.4 Fourier Series Approximations

In the example dynamics given in this chapter, it is difficult to ignore the obvious beneficial property of the single degree of freedom, angle α: it is naturally periodic. The moment source function itself is then also periodic, which leads to the conclusion that an excellent choice of methods to approximate it is a Fourier series (FS) representation. A Fourier series has the advantage that its eventual approximation function is entirely continuous. This property contrasts with the ST

method which defines a separate system at each discrete region of state space, leading to the possible ambiguity of behavior during transition between these regions as noted in Section 4.2.3. For periodic source functions (like $M(\alpha)$) that have a sinusoidal quality, the number of Fourier coefficients required to approximate them scales with dominance of frequencies. For simple, low-frequency-dominant functions such as the moment source function given for the airfoil in Figure 4.9 and even the Arbject in Figure 3.9, very few coefficients may be required to obtain the same or better level of accuracy when compared to the ST method. An example of FS approximation of the function in Figure 3.9, which is for the Arbject of Section 3.3.2 in FM flow, can be seen in Figure 4.13 The approximate version, $\sim M(\alpha)$, is created in this example with nine Fourier terms.

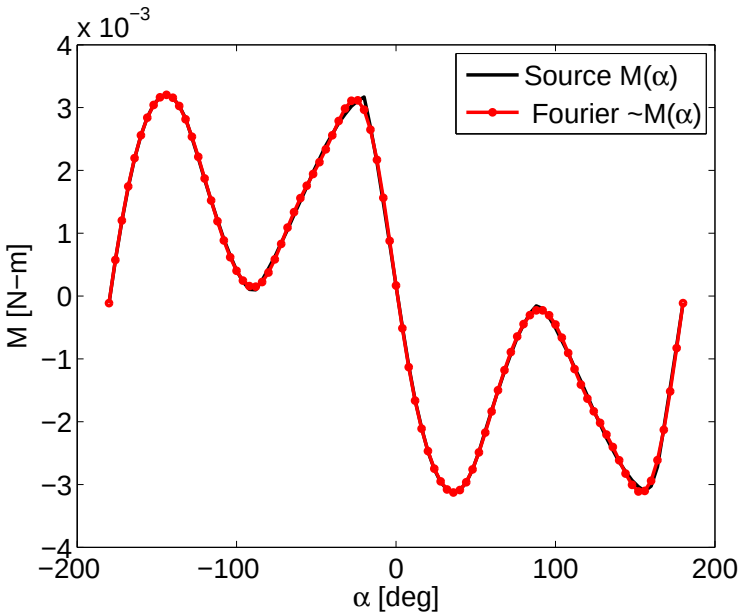


Figure 4.13: Fourier series approximation of the moment source function for the Arbject in free-molecular flow of N_2.

The equivalent ST version of this approximation would require as many coefficients as there are piecewise linear segments (in this case, there are 90 segments). Overall, FS is superior for this class of function; as it is a cleaner approach.

Where FS sees disadvantages is when being used on source functions of states that are not periodic (namely rates). For simpler functions that do not oscillate, as is expected for a function of

a state such as object speed or gas density, more Fourier terms are needed than would be required if a simpler linear or polynomial model were used. Non-periodic functions will also be very poorly approximated at the boundaries of their domains where FS naturally enforces a periodicity that is not truly present. These drawbacks may be of little concern, however, when larger dimension state spaces require multidimensional source functions. The additional level of complication in such cases is more cleanly and easily implemented with FS as generation of Fourier coefficients generalizes to $n-$dimensional functions more obviously than does the ST method (although ST is the most general method, and certainly works for any number of state subspace dimensions). Care is then taken to note that non-periodic states that move out of the appropriate domain of computed data cannot be used to evaluate a source function at those points. This is an acceptable limitation as the range of validity of states is already restricted to the computed domain of source data.

In a Fourier representation, the source functions are replaced by sums of Fourier terms upon integration of the equations of motion. No other treatment of the equations is necessary. This contrasts with ST in which a Taylor expansion of the equations is performed, with the intent of obtaining constant coefficients that are functions of partial derivatives of each source function with respect to the state. Thus, FS is also a simpler implementation at the integration stage, requiring no other special considerations (such as selection of an appropriate set of constants corresponding to the current state's discrete region in ST) other than the aforementioned enforcement of the state being in a valid location overall.

Chapter 5

Flow-Motion Coupling and Transpiration Models in DSMC

In order to generate accurate information that can be used in rigid-body dynamic simulations and analysis, a number of considerations must be made with respect to how DSMC can be used accurately. In most rigid-body dynamic representations of motion, aerodynamic force is a function of a number of state variables. In the most general case, the state will include positions or displacements (such as angles that represent orientation) as well as rates (such as the rates of change of the orientation angles) such as the single degree-of-freedom example given by Equation 3.2. It order to understand what forces, moments and heat transfer the rigid-body experiences as it moves through the gas, it may be necessary to consider motion that covers the full state range. This means that a multitude of DSMC computations must often be performed over various states, which may make impractical demands on computational resources.

Using the airfoil example examined in previous chapters, the effect of the rate of change of angle of attack $\dot{\alpha}$ on the flow is first considered. Since the moment source function M is a function of the angular position, a number of DSMC computations would be performed over a range of α in order to establish the dependence of moment on this angle. Also, each range of α must be computed at a certain $\dot{\alpha}$ in order to generate the basic set of data from which approximations to M may be made. In order to resolve the system's dependence on rates, some method performing DSMC calculations for a moving boundary must be employed. The simplest method ("real motion") is to employ a moving boundary method such that the immersed body rotates in real time as the computation evolves as shown in Figure 5.1. This method is typically extremely costly for any

numerical technique, and DSMC is little exception.

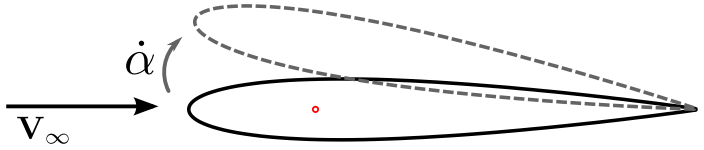


Figure 5.1: The real motion method for an airfoil that is rotating upward at $\dot{\alpha}$.

An alternative to true rotation is to adjust the stream direction to mimic a rotation ("true-equivalent" or just "true"). The flow entry angle is adjusted to $-\alpha(t)$ over time by changing the domain boundary conditions appropriately as illustrated in Figure 5.2. Adjusting the boundary conditions of the domain is a much more tractable approach as no re-meshing or other treatment of transforming geometry need be considered, and thus the additional computational overhead is minimal. Unfortunately, for very low rates, this method remains cumbersome. At lower rates, if a

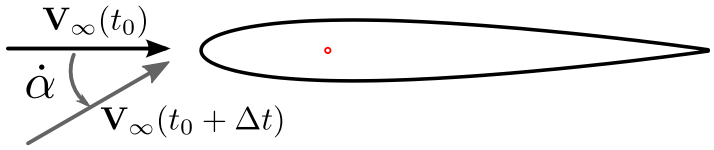


Figure 5.2: The true-equivalent method for an airfoil that is rotating upward at $\dot{\alpha}$. The freestream velocity is rotated the opposite direction at rate $\dot{\alpha}$.

given angular displacement is held constant, a DSMC simulation will take longer time to complete. The time scales of hypersonic flow (both transport and typical molecular collision times), are very small, and often orders of magnitude smaller than the characteristic time of body motion. For example, many flows that mimic a re-entering object at an altitude of 85km (Kn ≈ 0.01) will have characteristic flow times on the order of 10^{-7}s. A rotating object of one meter in scale displacing a small angle ($\sim$5 deg) must rotate at approximately 10^5 to 10^6 deg/s in order to be comparable to the characteristic flow time. This is much higher than any spacecraft is expected to rotate. Angular rates on the order of 1 to 100 deg/s are more realistic. If a simulation of $\dot{\alpha} = 1$ deg/s is desired, the body will require one second of simulation time to displace one degree. One full second

is exceptionally long in a DSMC simulation whose time scales are $\sim 10^{-7}$s, and thus the simulation will require a great deal of wall-clock time to complete. Seeing as how many of these simulations would be required to construct the necessary set of data, this "true" method is still undesirably slow.

Another approach that can be examined is the use of transpiration boundary conditions on the immersed body. Transpiration is mentioned in Section 3.2.1 in the context of free-molecular and DSMC simulation. It is the method used in the one degree-of-freedom studies presented in Chapters 3 and 4. In this method, the body remains fixed in the computational frame, and no domain boundary conditions are adjusted. Instead, each element of the body's surface is given a velocity $\boldsymbol{V}_m$ that corresponds to the velocity that element would experience if it were rotating about a given reference location $\boldsymbol{r}_{ref}$, as seen in Figure 5.3. Traditionally implemented in incompressible

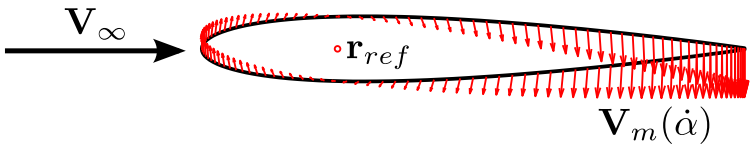


Figure 5.3: An illustration of a transpiration boundary condition on an airfoil that is rotating upward at $\dot{\alpha}$ about reference point $\boldsymbol{r}_{ref}$.

continuum flow CFD [24, 61, 59], the goal is to mimic the effect of rotating without changing anything during the simulation. This method permits computations to be performed in such a way as to cover a range of rates and angles in minimal time. What remains to be understood is the difference between the transpiration approximation of movement and actual movement. In other words, the nature of motion-flow coupling must be understood in order to make use of a reduced-order model such as transpiration. When the flow becomes independent of the body's motion, and thus depends only on its orientation, it can be termed "decoupled".

This investigation aims to provide some initial insight into the mechanisms of motion-flow coupling in a rarefied gas, focusing on the limit of angular rate in the aforementioned example problem. The specific goal is to demonstrate that transpiration can be an effective model at low

angular rates. Additional information regarding accuracy and specifics of DSMC simulations is presented.

5.1 Inaccuracy of the true-equivalent method

The true-equivalent method has been applied historically in continuum numerical simulations to great effect, often in establishing source functions for airfoils and the like. However, there is a serious problem with this method when applied to supersonic flow. In fact, the problem applies for any flow in which flow structures such as shocks, turbulence or vortices exist upstream of the immersed solid object. If the goal of the simulation is to compute body surface properties such as forces and power flux, molecules of a moving, entering stream will not impact the surface in the same location or with the same momentum and energy as if they had been unobstructed by flow structures. Consider the case of the airfoil at any given α, as shown in Figure 5.4, where the angle is enforced by adjusting the left and bottom stream boundaries to include appropriate components of velocity that give the desired stream angle. A particle that enters from either the left or bottom boundary will, in FM flow, certainly remain incident upon a given location on the airfoil's surface and will impact that location, imparting a fraction of its momentum and energy–quantities that are known (actually, *specified*, since they are generated by the program) at the time of entry. This is shown schematically as the *projected* impact position in Figure 5.4. However, if collisions are occurring, a shock forms upstream of the body, which the same entering particles must first cross before having any opportunity to impact the body. Crossing the shock effectively renders the history of the moving particles irrelevant. Having traversed the shock, they may no longer have the expected properties and may not even be incident on the body at all; and certainly not in the same location as if they had been free-molecules. The *actual* position shown in Figure 5.4 is of the case where a particle that has traversed the shock has not impacted the surface at all. The shock's existence prevents any prediction being made regarding how the surface will experience the domain's flow as it evolves if only the boundary conditions of the problem are known. In other words, no "shock transfer function" exists that would describe how a molecule incident upon the

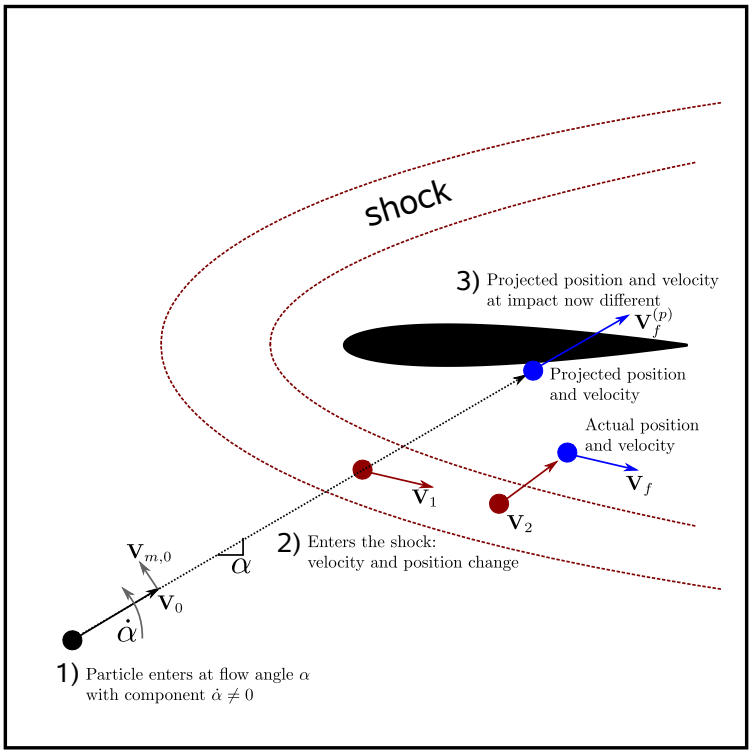


Figure 5.4: Illustration of shock traversal in DSMC by a particle entering at stream angle α and stream angular rate $\dot{\alpha}$.

shock is affected by the shock, and thus how it leaves the shock. Some analytical representation of the Boltzmann equation's collision operator would be required to obtain such a description. If such a means of analysis existed, numerical simulations wouldn't be necessary at all.

Changing the entering stream's angle is likely an acceptable means of simulating a statically-angled flow (e.g. $\dot{\alpha} = 0$). But when a moving boundary condition at the domain stream boundaries is imposed in order to simulate motion of the body, the assumption of equivalency is incorrect. In the example of $|\dot{\alpha}| > 0$, the desired feigning of a rotation of the body will not be achieved. This is because the addition of a component of velocity that changes at each flow time step (i.e. acceleration at the domain boundary, shown as $\boldsymbol{V}_{m,0}$ in Figure 5.4) will have no predictable effect, in general, on the body due to shock (or other flow structure) interference. When attempting to simulate a rotating body by increasing $\alpha(t)$ over time, one way in which the motion may be characterized is by the additional components of momentum and energy that the rotation alone brings to the flow. The

way in which a moving or transpiring surface affects the energy and momentum balance between itself and the gas is discussed in the next section. Appendix B also presents an explanation of an easily overlooked problem when simulating angled entry flow which pertains to implementations of true-equivalent methods in DSMC.

5.2 Energy and Momentum Balance

The most obvious inputs to the collisional feedback system are the additional energy and momentum that the moving or transpiring surface imparts to the gas. Calculation of these quantities can instead be examined with respect to the body. An energy or momentum balance between the surface and the gas requires the relative velocity of particles impinging on and reflecting from the surface. Consider a single molecule that is incident upon a moving or transpiring surface. The relative velocity of the molecule can be decomposed as the difference of the incident velocity $\boldsymbol{V}_i$ and the velocity due to body motion or transpiration $\boldsymbol{V}_m$ of the surface element upon which the molecule is incident, shown as Equation 5.1. The definition is similar for reflecting molecules but with the incident velocity replaced with the reflected velocity $\boldsymbol{V}_r^{GSI}$ which is due to the gas-surface interaction model only, shown as Equation 5.2. Figure 5.5 illustrates this arrangement, highlighting incidence for one direction of motion for a surface element, and reflection for another direction on a different element.

$$\boldsymbol{V}_i^{rel} = \boldsymbol{V}_i - \boldsymbol{V}_m \tag{5.1}$$

$$\boldsymbol{V}_r^{rel} = \boldsymbol{V}_r^{GSI} - \boldsymbol{V}_m \tag{5.2}$$

The movement velocity $\boldsymbol{V}_m$ is that of a specific location on a surface (i.e. the location of a surface element). When implementing transpiration on the surface in DSMC, $\boldsymbol{V}_m$ is prescribed directly. When performing a true motion simulation, $\boldsymbol{V}_m$ arises naturally from the prescribed motion. Transpiration is implemented by specifying $\boldsymbol{V}_m$ at every surface element. When a molecule impacts a surface element, the energy transferred to the element is given as a function of the relative velocity

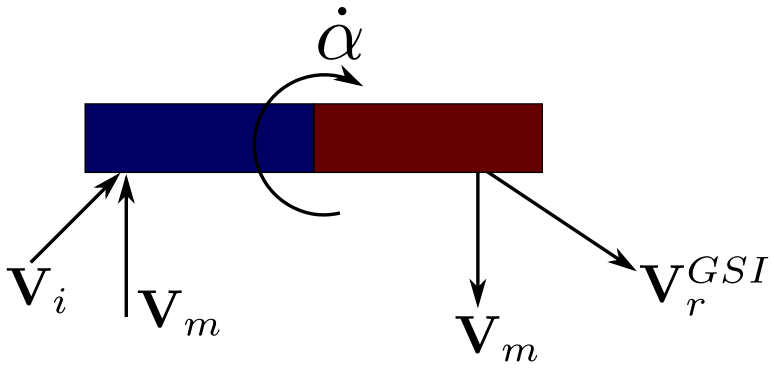


Figure 5.5: Velocity definitions of impact and reflection of molecules incident on a rotating object.

and can be written as

$$E_i = \frac{1}{2} m \|\boldsymbol{V}_i^{rel}\|^2 + E_i^{int} \tag{5.3}$$

where E^{int} is the sum of all energies of the internal modes (e.g. rotational, vibrational). This leads to the energy decomposition of

$$E_i = E_{i,tr} + E_i^{int} + E_{i,m} \tag{5.4}$$

$$E_{i,m} = \frac{1}{2} m \|\boldsymbol{V}_m\|^2 - m\boldsymbol{V}_i \cdot \boldsymbol{V}_m \tag{5.5}$$

where $E_{i,tr}$ is the incident translational energy and $E_{i,m}$ is the contribution of energy to the body solely due to actual or implied motion. The momentum is simply

$$\boldsymbol{M}_i = \boldsymbol{M}_{i,tr} + \boldsymbol{M}_{i,m} \tag{5.6}$$

where $\boldsymbol{M}_{i,m} = -m\boldsymbol{V}_m$ is the contribution of momentum due to motion. For reflecting molecules, these definitions are similar, but with $E_{i,tr}$ and $\boldsymbol{M}_{i,tr}$ replaced with $E_{r,tr}^{GSI}$ and $\boldsymbol{M}_{r,tr}^{GSI}$, respectively.

Using these definitions of energy and momentum, the total change of energy experienced by

the surface is then

$$\begin{aligned}\Delta E &= E_i - E_r \\ &= E_{i,tr} + E_i^{int} + E_{i,m} - (E_{r,tr}^{GSI} + E_r^{int} + E_{r,m}) \\ &= \frac{1}{2}m(|\boldsymbol{V}_i|^2 - |\boldsymbol{V}_r^{GSI}|^2) - m\boldsymbol{V}_m \cdot (\boldsymbol{V}_i + \boldsymbol{V}_r^{GSI}) + E_i^{int} - E_r^{int}\end{aligned} \tag{5.7}$$

and change in momentum is

$$\begin{aligned}\Delta \boldsymbol{M} &= \boldsymbol{M}_i - \boldsymbol{M}_r \\ &= \boldsymbol{M}_{i,tr} + \boldsymbol{M}_{i,m} - (\boldsymbol{M}_{r,tr}^{GSI} + \boldsymbol{M}_{r,m}) \\ &= m(\boldsymbol{V}_i - \boldsymbol{V}_r^{GSI})\end{aligned} \tag{5.8}$$

Equation 5.8 indicates that for momentum, there is no dependence on actual or implied motion. However, due to the cross-term in Equation 5.7, there is a contribution to energy of the surface by the motion that is a function of the motion-only velocity $\boldsymbol{V}_m$. When a surface element is not actually moving, but is instead using transpiration, the imposed transpirational velocity is equal to $\boldsymbol{V}_m$. There is no difference between actual motion and feigned motion via transpiration. When considering the effect of motion or transpiration on the surface, energy and momentum balance can be described identically for both models.

One important note to make is that the change in energy experienced by the surface can only occur at the surface itself. This distinction is made particularly to further explain why the true-equivalent model cannot accurately model motion in hypersonic rarefied flows over solid bodies, as discussed in the previous section. The body that lies behind the shock in Figure 5.4 experiences only local flow perturbations, as the shock prevents information upstream from reaching the surface unaffected. Particles that are near the surface will reach an energy balance with the surface in the way described in this section only if they are allowed to impact it, and the body is actually moving (thus, implicitly imposing $\boldsymbol{V}_m$) or transpiring. In the true-equivalent model, the imposition of $\boldsymbol{V}_m$ is performed at domain entry rather than at the surface, meaning that the additional energy given by the cross-term of Equation 5.7 will not "reach" the surface, as it is changed as soon as particles

traverse the shock, effectively destroying the target body-gas energy balance. Superimposing an accelerative motion at entry in the form of $\boldsymbol{V}_m$ intentionally creates an energy balance in the wrong context and location.

5.3 Positional Arguments for Transpirational Accuracy

Discussion of energy and momentum balance in cases of motion and transpiration derives from arguments relating to velocity and acceleration of particles and how those quantities are affected by the velocity and acceleration of an immersed body. Actual motion and transpiration appear to give the same results in the velocity context. Changes in energy and momentum described in the previous section are per-collision, and thus can be described as being collision-normalized (actually, mass-specific) quantities. A greater picture of true total changes in these quantities is impossible to obtain without consideration of dimensional energy, which requires knowledge of the rate of surface impact of particles. The proper mechanism for transfer to energy and momentum is given in terms of mass and velocity. If both models of motion introduce the same change in velocity (given as $\boldsymbol{V}_m$), then changes may still occur due to more or fewer particles impacting the surface. Therefore it is unlikely that an argument in a velocity context alone can properly explain if transpiration remains accurate for all conditions of motion. The other context to consider is necessarily position. Understanding the effect of particle positioning with respect to a surface under general conditions is necessary to knowing the number flux of particles to the surface, and thus the final description of energy and momentum balance.

One way to eliminate the effect of position in a motion/transpiration problem is to choose a simplified geometry for which all geometric transformations yield an identical flow problem. Flow over a disk (cylinder) meets this requirement. Classic continuum fluid dynamics uses the cylinder as a canonical problem. Transpiration is often applied at the boundary of a cylinder to simulate a rotation. For a cylinder of radius R, a pure rotation about the center point in 2D gives the magnitude of the motion-only velocity as $|\boldsymbol{V}_m| = R\dot{\alpha}$, which is constant for all points on (or segments comprising) the surface. During a numerical simulation, true motion can easily be created

by rotating the cylinder geometry the appropriate angle according to the time step; however the result is an identical circle to the starting geometry and thus the operation is trivial. The cylinder cannot extend or retract any portion of its surface into or out of the flow, as is possible in more general cases. It cannot create a conflict of position between itself and resident particles in the volume of the domain into or out of which it has moved. For other geometry, a surface that moves into a location where it was not located at the previous time must "push" particles in that location, effectively creating or inducing reflections that would not have occurred if the surface had remained static. Alternatively, if a surface retracts from a location, it creates a space in the domain that was not available previously. Particles may then enter this region instead of impacting the surface, as they would have had the surface remained static, again. In is unclear how these mechanisms actually occur in reality or how to model them in a DSMC simulation. A rotating cylinder never creates these conflicts, implying that if it is modeled with transpiration, a simulation should give identical results to a cylinder modeled by imposing true motion.

For general geometry, however, error owing to positional conflicts and uncertainties is likely to exacerbate as rates of motion increase. A surface being in a new position may even cause some particles that would have impacted it to avoid it completely, or cause particles that would have never be incident upon the surface to impact it.

5.4 Numerical Examination of Transpiration in DSMC

Without the ability to perform a true motion simulation in DSMC, the only conclusion that may be reached regarding the transpiration model is whether it is necessary at all. For the high Ma cases commonly addressed in problems involving low orbital flight, the stream speed is of the gas is often so high that the motion of the body may be negligible compared to its angular position with respect to the stream. Forces, moments and energy transfer to the body likely depend much more on flow incidence angles than on angular rates when said rates are low enough. To test this hypothesis, a numerical experiment was performed involving a 1m chord NACA-0012 airfoil and an Arbject of parameters ($L = 1.00$m, $\theta = 30$deg, $\boldsymbol{p}_1 = (0.75, 1.00)$m, $R = 0.20$m, $d = 0.40$m, $\omega = 6$).

For each geometry, physical conditions of Table 5.1 and were used along with the DSMC resolutions and additional parameters of Table 5.2. A range of $\alpha \in [0, 45]$deg was simulated for seven values

Table 5.1: DSMC physical conditions for a transpiration rate limit experiment in N_2 with gas reference temperature of 273 K and a fully diffuse gas-surface interaction model.

Variable	**Symbol**	**Value**	**Unit**
Mach number	Ma	28.40	
Knudsen number	Kn	0.01	
Stream temperature	T_{in}	184.0	K
Body surface temperature	T_S	300	K
Domain length	L_x	2.0	m
Domain height	L_y	2.0	m
Inflow number density	n_{in}	1.2944×10^{20}	m^{-3}
Inflow speed (aligned with +x)	V_{in}	7853.20	m/s
Gas molecular mass	m_g	4.65×10^{-26}	kg
Gas diameter	d_g	4.17×10^{-10}	m
Gas ratio of specific heats	γ	7/5	
Gas constant	R	296.8	kJ/(kg–K)
Gas rotational deg. of freedom	ζ	2	
Rotational relaxation constant	Z_{rot}	1/5	

of $\dot{\alpha}$, representing a scan over six orders of magnitude: $\dot{\alpha} \in [0, 1, 10, 100, 1000, 1\times10^4, 1\times10^5]$deg/s. At each angle, the body was given a distribution of transpiration velocity $\boldsymbol{V}_m$ along its surface corresponding to a specified angular rate $\dot{\alpha}$. A DSMC simulation was performed which allowed the flow to evolve from $t = 0$ to $t = t_{ss}$, at which point steady state sampling was activated and the transpirational velocity was applied for the remainder of the evolution to $t_{\max}$. Variable time steps were used with $\gamma_t = 5.5$ (see Section 1.2.4).

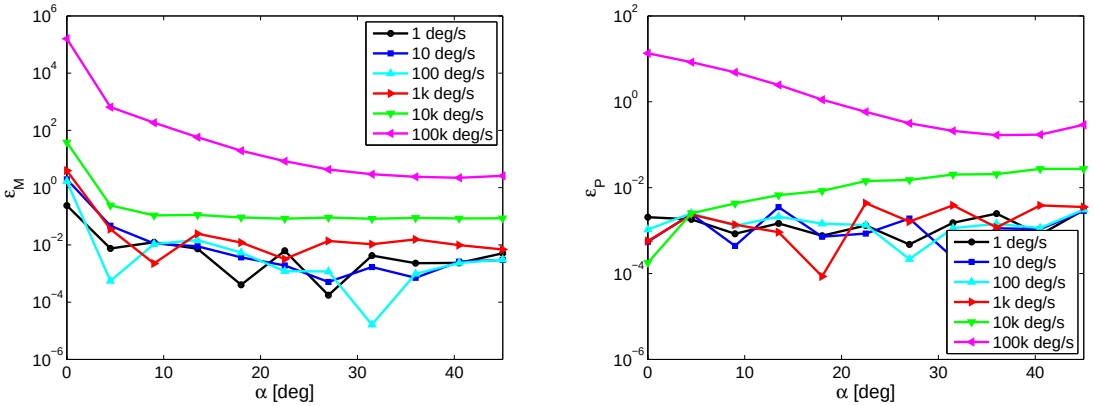
To examine the effect of increasing $\dot{\alpha}$ via transpiration, the static case $\dot{\alpha} = 0$ deg/s was used as a datum and the relative error of each other rate, for each physical quantity (forces, moment and power), was calculated with respect to this datum. For the airfoil, comparisons for moment error about the quarter chord and error in total power to the surface are given in Figure 5.6. For the Arbject, the same information (with moment taken about $\boldsymbol{p}_1$) is given in Figure 5.7. The objective of this method of comparison is to determine when transpirationally modeled rotation has any significant effect on the values of the aforementioned macroscopic variables. It does *not* offer

Table 5.2: DSMC resolutions and other parameters for a transpiration rate limit experiment.

Variable	**Symbol**	**Value**	**Unit**
Ratio of real-to-simulated particles	F_N	2.0×10^{15}	
Num. cells in x	N_{cx}	170	
Num. of cells in y	N_{cy}	170	
Num. of subcells per cell (each dir.)	N_{sc}	2	
Num. time steps per sample	N_{sam}	4	
Time to steady state	t_{ss}	0.001	s
Time to end	$t_{\max}$	0.003	s
Voxel resolution	n_{vox}	800	

any means of determining the conditions under which transpiration, as a model, is less accurate when compared to a true motion simulation. The lack of a true motion DSMC simulation prevents such a study from being performed. The current study emphasizes that the only valid means of simulating motion aside from a true motion method would be the use of transpiration (the true-equivalent method's natural inaccuracy, as described in Section 5.1, prevents it from being a valid candidate). However, if there appears to be little difference between using transpiration and simply ignoring motion when rates are below a certain limit, then there may be little reason to consider modeling motion below such a limit at all. The discussion of Section 5.3 indicates that it is insufficient to merely observe, numerically, that transpiration results in negligible change in forces at low rates in order to conclude that flow is decoupled at those rates. The "velocity argument" may be sufficient, but the "mass argument" (i.e. understanding that number flux changes in ways that may be strongly coupled to motion and the evolution of the near-field gas) is incomplete without much more detailed analysis, or the availability of a true-motion DSMC simulation code.

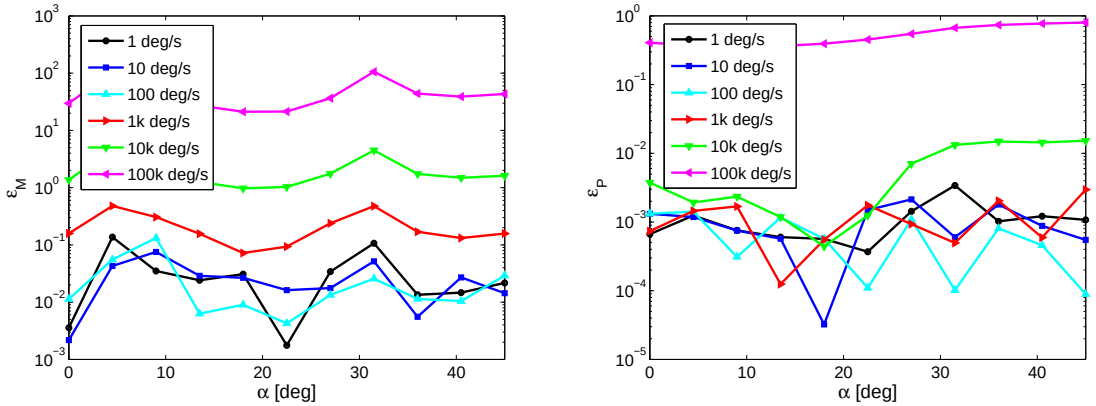
From Figure 5.7 it is clear that the highest rates result in severe departures from the static datum. Although the intuitive prediction that was initially posed would seem to be validated, i.e. that some rates are simply too high to ignore, these errors appear to be alarmingly large. Part of the initial assumption is that when $\|\boldsymbol{V}_m\|$ is on the order of the characteristic stream speed (or possibly just the near-field characteristic speed), the gas motion becomes more strongly coupled to the body motion. Again, this study is unable to validate such an assumption. However, there appears to be



(a) Relative error in moment about the quarter chord. (b) Relative error in total power to the surface.

Figure 5.6: Effects of transpiration velocity on a NACA-0012 airfoil for various values of $\dot{\alpha}$ over a range of α expressed in terms of relative error from the static case.

a reason why the large errors seen in Figure 5.7 that derives from the implementation of the DSMC movement algorithm. When the surface motion velocity $\boldsymbol{V}_m$ is great enough and aligned with a surface's normal direction, "receding" surfaces that are modeled with transpiration will artificially trap particles that impact them. Consider the example scenario in Figure 5.8. The surface is physically static in the simulation. The velocity $\boldsymbol{V}_m$ is the transpirational velocity appropriate to the surface's desired movement, which is directed upward. A particle that impacts the surface is then positioned as close as possible to the wall and given its new total reflected velocity $\boldsymbol{V}_r$, which is the sum of the velocity solely due to GSI, $\boldsymbol{V}_r^{GSI}$ and $\boldsymbol{V}_m$. The resulting velocity is directed upward, back towards the surface. Immediately upon the next movement step, this particle, having a velocity and position that make it incident upon the same surface it just impacted, will impact the same surface again. The same surface interaction will occur, placing it just off the surface, with a velocity that implies yet another immediate impact upon next movement. Many GSI models tend to employ reflection kernels that yield a most probable reflected direction that is close to the surface outward normal direction. The fully diffuse Maxwellian kernel is an example of this type of model, where the raised cosine distribution results in most molecules leaving the surface with a large normal component. Thus, on average, a transpiring surface with $\boldsymbol{V}_m \approx -\boldsymbol{V}_r^{GSI}$ will artificially



(a) Moment about $\boldsymbol{p}_1$.

(b) Total power to the surface.

Figure 5.7: Effects of transpiration velocity on an Arbject for various values of $\dot{\alpha}$ over a range of α expressed in terms of relative error from the static case.

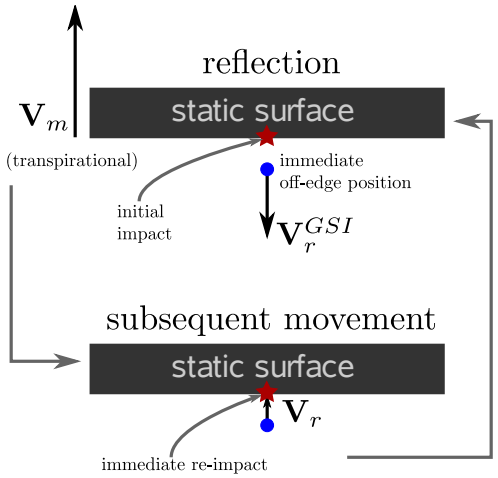


Figure 5.8: Illustration of transpiration artificial particle entrapment.

accumulate particles on (or immediately near) itself, effectively retaining them until collisions force some to leave. Each particle in this region will continue to re-impact the surface, transferring more energy and momentum to it, and thus inflating the final results for quantities such as force and power.

This artificial entrapment can be witnessed in simulation. Figure 5.9 shows the distribution of particle impact rate over the surface of the NACA-0012 airfoil and Arbject under the conditions

of Tables 5.1 and 5.2 for a number of different values of $\dot{\alpha}$. The distribution of $\dot{N}$ is plotted along the surface coordinate s, (which is shown graphically in the figure) where $s = 0$ corresponds to the trailing edge for the airfoil geometry and the indicated point for the Arbject. For the airfoil, transpiration was applied to mimic rotation upward (also illustrated in the figure), causing the upper surface close to the trailing edge (where s close to zero) to recede. For values of $\dot{\alpha} \leq 10,000$ deg/s, the distribution is relatively symmetric about the airfoil's plane of symmetry, indicating only weak, if any, effects of the imposition of transpirational motion. As $\dot{\alpha}$ approaches the value of roughly 32244 deg/s, entrapment begins to occur in the receding region, causing $\dot{N}$ to increase dramatically. The value of $\dot{\alpha} = 32244$ deg/s is the characteristic angular rate of the system that is determined by equating the induced transpirational/motion speed at the trailing edge, $V_m = \dot{\alpha} L_c$, with the most probable reflected speed $c_{r,mp} = 1/\beta_s = \sqrt{2kT_s/m_g}$ of the surface in a diffuse GSI model. In this case $c_{r,mp}$ is roughly 422 m/s, and $L_c = 0.75$m is the maximum moment arm of the airfoil about its quarter chord). Solving for $\dot{\alpha}$ gives the aforementioned value of 32244 deg/s, which marks the point at which the transpirational velocity roughly counters the reflected velocity due to GSI only. Above this value, artificial particle entrapment creates incident number rates orders of magnitude greater than the natural greatest region – the leading edge. In Figure 5.9b, the freestream flow is angled at 45 deg, thus causing $\dot{N}$ to be highest on the lower surface as would be expected. Entrapment occurs in the receding region nonetheless, causing number incident rates approximately equal to those occurring at the lower surface. The Arbject's special rotation rate of $\dot{\alpha} = 38768$ deg/s is based on the same $c_{r,mp}$ acting from the first point of the highlighted upper region in Figures 5.9c and 5.9d. This region is the most normal to its induced movement velocity $\boldsymbol{V}_m$. Less drastic entrapment effects are present for the Arbject, possibly because of the overall bluntness of the geometry. Surfaces that are normal to the incident stream velocity naturally induce more collisions upstream of themselves, which may serve to expedite the evacuation process of entrapped particles.

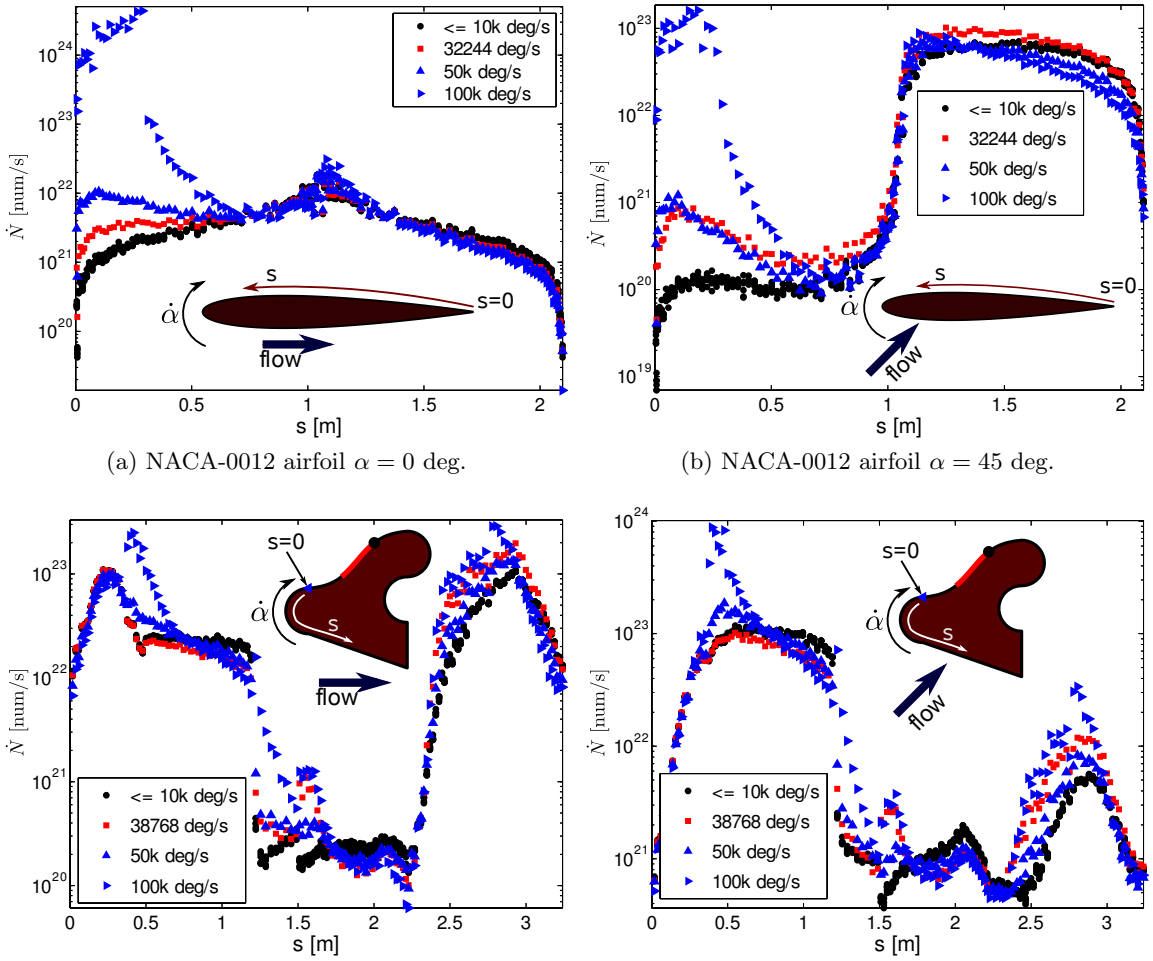
Contrast this entrapment mechanism with a surface that is actually moving, as shown in Figure 5.10. Particles are reflected with the same value of $\boldsymbol{V}_r$, on average, as in the transpirational

case. However, the surface is repositioned to its new location by moving a displacement $\boldsymbol{\delta}_m = \boldsymbol{V}_m \Delta t_f$, where Δt_f is the flow time step. The particle, when moved, displaces by $\boldsymbol{\delta}_r = \boldsymbol{V}_r \Delta t_f$ in the same time interval. If over this interval, neither the particle or surface is said to accelerate (as would likely be the assumption in any numerical scenario), then the particle will only re-impact this receding surface if $\boldsymbol{\delta}_r < \boldsymbol{\delta}_m$. In the context of the figure, this inequality is satisfied when $\|\boldsymbol{V}_r^{GSI}\| < 0$. In general terms, this means that particles which have just been reflected cannot catch up to a moving surface as long as $\boldsymbol{V}_r^{GSI} \cdot \hat{\boldsymbol{n}} > 0$, which is always true for any GSI model, usually by construction of the model (Note that, in Figures 5.8 and 5.10 the off-edge position should considered to be infinitesimally close to the surface). Thus it can be concluded that true motion on receding surfaces will never be responsible artificially entrapping incident particles in a simulation that performs surface motion in a manner consistent with Figure 5.10.

On the opposite side of the body in Figures 5.8 and 5.10, a compression or *encroaching* occurs. In this region, it is natural to assume that re-impacts will occur which are not artificial. Though for the transpiration case, since the surface is static, fewer impacts will occur than would be expected in the true motion case. And although this effect is artificial, its effect on the state of the nearby gas should not be as drastic as the receding case. The creation of a local region with high density and high collision rate will serve to collect more particles at a faster than linear rate. This is particularly true if the NTC method (Section 1.1.2) is used to select collisional pairs since it is a function of what is essentially number density squared. Furthermore, it relies on a continuously-updated value of the collision cross-section $(\sigma_T c_r)_{max}$, which changes proportional to $1/n$. Some of this uneven error is visible in Figure 5.9b in the encroaching region ($s \gtrsim 1$). Here, for $\dot{\alpha} > 32244$ deg/s, number rates drop slightly below those of the < 10k deg/s data, owing to artificial expansion of the nearby region. Still, the difference in $\dot{N}$ of these high-$\dot{\alpha}$ cases *below* the lowest rate cases in the encroaching region is much smaller than the difference *above* the lowest rate cases in the receding region.

It is likely artificial entrapment causes the large departure in macroscopic variables for high rotational rates seen in Figure 5.7. In these computations, it is worthy to note that not only does

transpiration at these high rates cause unphysical error, but computational performance is greatly degraded. Performing simulations for the $\dot{\alpha} = 100$k deg/s cases took an order of magnitude more time to complete due to the overly high collision rate near receding surfaces.



(a) NACA-0012 airfoil $\alpha = 0$ deg.

(b) NACA-0012 airfoil $\alpha = 45$ deg.

(c) Arbject $\alpha = 0$ deg.

(d) Arbject $\alpha = 45$ deg.

Figure 5.9: Particle impact rate for simulated rotation of each geometry under various values of $\dot{\alpha}$ using transpiration. Rates in the $\leq$ 10k deg/s category were for $\dot{\alpha} \in [0, 1, 10, 100, 1000, 10000]$.

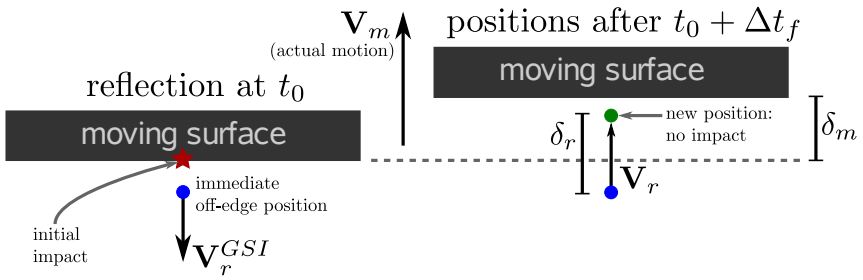


Figure 5.10: Illustration of true motion particle reflection freedom: particles are not artificially trapped by receding surfaces.

5.5 Conclusions

At least one of the differences between a transpiration model and true motion when implemented in a numerical method such as DSMC has become evident. When a surface is physically static and its motion is merely simulated with transpiration, very little difference between no motion and any motion can be observed in cases of high Ma unless the induced motion, applied via transpiration, approaches (or exceeds) the characteristic rate of the system that places receding surfaces in modes of artificial entrapment. For the example problem of a NACA-0012 airfoil under pure rotation, this rate is

$$\dot{\alpha}_c^{TP} = \frac{1}{L_c}\sqrt{\frac{2kT_s}{m_g}} \tag{5.9}$$

for a diffuse GSI model. Furthermore, this is an argument deriving from position, as anticipated in Section 5.3, though by no means the entire argument. The details of gas-body feedback are still difficult to describe, and require a true motion DSMC simulation to make proper numerical comparisons. The results in this chapter simply present some initial paths of investigation.

In Chapters 3 and 4, evidence was presented for ignoring high rates of motion in the context of rigid-body dynamics deriving from rarefied gas flow. Dependence on $\dot{\alpha}$ in approximation methods was observed to be orders of magnitude weaker than functionality on the angle itself. In this Chapter, the question of whether to consider angular rates in rigid-body dynamic simulations in transition flow at all was posed. The various methods of modeling motion were presented, with transpiration being identified as being the only reduced-order model that may be both viable and generally feasible in practice. Without a simulation code that models true boundary motion, a direct comparison could not be made between transpiration and true motion. Examining particle-specific energy and momentum balance between the gas and solid surfaces is unlikely to show that transpiration is any different from true motion. However, in practice, simulations show a significant change in macroscopic quantities when very high values of transpirational velocity are imposed. Part of the reason for these changes stems from artificial increases in the number of particle impacts on certain surface segments. These increases, noted as an entrapment of particles that impact receding

surfaces, can account for the dramatic increase in macroscopic variable values on immersed bodies when transpirational velocities are higher than some description of characteristic velocity such as Equation 5.9; which is determined by noting when the reflected velocity due to GSI of a surface is comparable in magnitude to the transpirational velocity. In real motion, artificial entrapment is not expected to occur as surfaces will physically move away from particles in time to avoid improperly collecting particles and putting the near-field into a self-compressing loop.

Though more detailed explanations of coupling may be required, it seems acceptable for the purposes of this work to assume that flow-motion decoupling occurs in realistic systems such as spacecraft flying in low Earth orbit, at high Ma . Only at extremely high velocities (e.g. rotations) is a coupling likely to create additional forces, moments and power that are large enough to be distinguished from those observed without any body motion at all. Thus, the remaining development of rarefied aerodynamics problems presented in the following chapter will ignore dependence on angular rates and focus primarily on angular positions.

Chapter 6

Rigid-Body Dynamics in Orbit

Introducing body dynamics to orbital motion is the final goal of this work. The improvements in DSMC, the Voldipar code and lessons learned from development of single-degree-of-freedom dynamics may be combined to form the basis for creating time-accurate, unsteady simulations of objects in orbital flight. Particular attention in this chapter is paid to continuum-rarefied transition dynamics. Example problems are given which simulate very low Earth orbital flight and atmospheric entry. These high-fidelity simulations, and the dynamic information they yield form the demonstrative conclusion of the overall goal of improving spacecraft dynamics from the perspective of rarefied aerodynamics.

6.1 Equations of Motion

To perform a 2D simulation of a rigid body in orbit, the perifocal frame (the natural orbital plane created by the gravitational two-body problem), is chosen. The center of the Earth is considered to be the origin of an inertial frame in space which is spanned by the two basis vectors $\hat{X}$ and $\hat{Z}$. The center of mass of an orbiting body experiences a gravitational force acting inward towards the origin. In addition to gravity, the body experiences force exerted on it from its motion through the gas of the atmosphere. Figure 6.1 illustrates the necessary definitions of angles and basis vectors. At a given time in orbit, the center of mass is located at position $\boldsymbol{r}$, moving at velocity $\boldsymbol{V}$. For general orbits, the instantaneous velocity makes an angle γ with the angular direction basis vector $\hat{\boldsymbol{e}}_\theta$ which is classically termed the *flight path* angle. For purely circular orbits

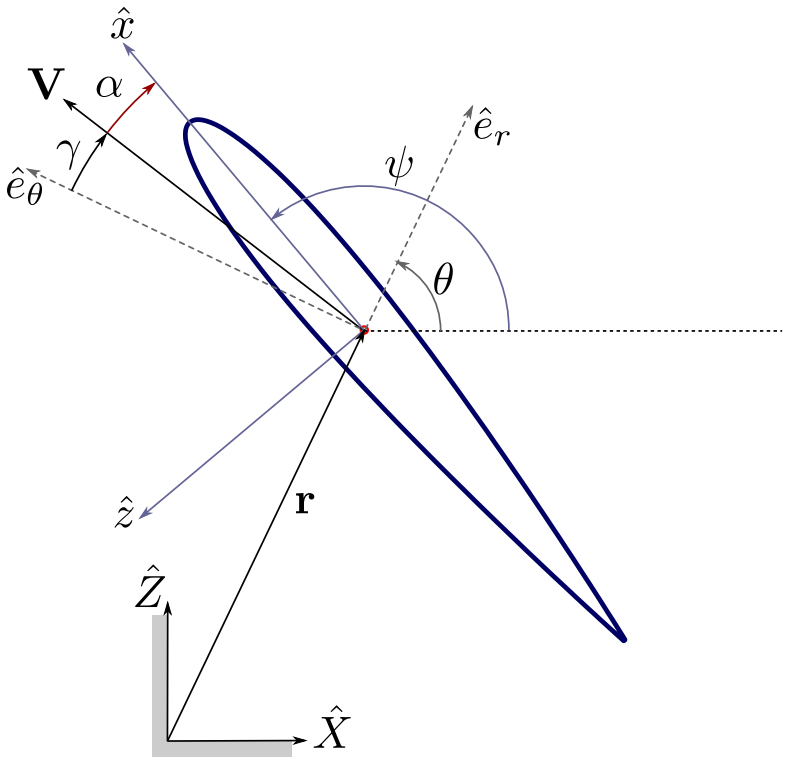


Figure 6.1: Angle and coordinate system definitions for an object in a 2D perifocal-only orbit.

$\gamma = 0$ (i.e. $\boldsymbol{V}$ remains tangent to the local horizon). The angle of attack α is defined as usual, with it being the angle between the velocity vector and the body frame's $\hat{\boldsymbol{x}}$ vector. The angle θ corresponds to the perifocal true anomaly. The angle ψ serves as a pitch angle in this formulation.

In order to add aerodynamic force into the system, the two-body pure gravitational equations must be augmented with additional force terms. Similarly, the body frame dynamic equations of motion must include gravitational force terms. No gravitational moments exist as the object is assumed to be small enough such that all gravitational force can be said to act at its center of mass, which is typical of artificial satellites. The Newtonian[51] gravitational equations are written first in the Earth-centered inertial frame (X-Z) as

$$\boldsymbol{F}_g^{(XZ)} + \boldsymbol{F}_a^{(XZ)} = m\ddot{\boldsymbol{r}}^{(XZ)} \tag{6.1}$$

where m is the orbiting body's mass and the position is $\boldsymbol{r}^{(XZ)} = [r_X \; r_Z]^{\mathrm{T}}$. Vectors $\boldsymbol{F}_g^{(XZ)}$ and $\boldsymbol{F}_a^{(XZ)}$ are gravitational and aerodynamic forces in (X-Z) frame, respectively. Gravity can be

expressed in this frame as

$$\boldsymbol{F}_g^{(XZ)} = -\frac{m\mu}{r^2}\begin{bmatrix} \cos(\theta)\ \hat{\boldsymbol{X}} \\ \sin(\theta)\ \hat{\boldsymbol{Z}} \end{bmatrix} \tag{6.2}$$

where r is the magnitude of vector $\boldsymbol{r}$ and μ is the gravitational parameter of the fixed/central body (e.g. Earth). Noting that θ and r are functions of time and given the definition of $\boldsymbol{r}^{(XZ)}$ as

$$\boldsymbol{r}^{(XZ)} = r(t)\begin{bmatrix} \cos(\theta(t))\ \hat{\boldsymbol{X}} \\ \sin(\theta(t))\ \hat{\boldsymbol{Z}} \end{bmatrix} \tag{6.3}$$

then the components of Equation 6.1 are calculated as

$$F_{g_X} + F_{a_X} = m\ddot{r}_X = m\left(\ddot{r}\cos(\theta) - 2\dot{r}\dot{\theta}\sin(\theta) - r\dot{\theta}^2\cos(\theta) - r\ddot{\theta}\sin(\theta)\right) \tag{6.4}$$

$$F_{g_Z} + F_{a_Z} = m\ddot{r}_Z = m\left(\ddot{r}\sin(\theta) + 2\dot{r}\dot{\theta}\cos(\theta) - r\dot{\theta}^2\sin(\theta) + r\ddot{\theta}\cos(\theta)\right) \tag{6.5}$$

Notation $r(t)$ and $\theta(t)$ have been dropped with the understanding that these variables are functions of time. Solving for $\ddot{r}$ and $\ddot{\theta}$ implicitly puts the system into the more natural r-θ frame. Defining $\boldsymbol{r}^{(r\theta)} = [r\ \theta]^{\mathrm{T}}$ gives

$$\begin{bmatrix} \ddot{r} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} r\dot{\theta}^2 \\ -\frac{2r\dot{\theta}}{r} \end{bmatrix} + \frac{1}{m}\boldsymbol{S}_{XZ}^{r\theta}\left(\boldsymbol{F}_g^{(XZ)} + \boldsymbol{F}_a^{(XZ)}\right) \tag{6.6}$$

where $\boldsymbol{S}_{XZ}^{r\theta}$ is the rotation matrix which transforms from the inertial frame X-Z to the r-θ frame defined as

$$\boldsymbol{S}_{XZ}^{r\theta} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \tag{6.7}$$

Using the definition of gravity from Equation 6.2, Equation 6.6 reduces to the simple form of

$$\begin{bmatrix} \ddot{r} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} r\dot{\theta}^2 - \frac{\mu}{r^2} \\ -\frac{2r\dot{\theta}}{r} \end{bmatrix} + \frac{1}{m}\boldsymbol{S}_{XZ}^{r\theta}\boldsymbol{F}_a^{(XZ)} \tag{6.8}$$

The aerodynamic force term $\boldsymbol{S}_{XZ}^{r\theta}\boldsymbol{F}_a^{(XZ)}$ is a force vector in the r-θ frame which may be denoted $\boldsymbol{F}_a^{(r\theta)}$. This vector may also be expressed as a transformation from the body frame (denoted as (b)) to the r-θ frame as

$$\boldsymbol{F}_a^{(r\theta)} = \boldsymbol{S}_{XZ}^{r\theta}\boldsymbol{F}_a^{(XZ)} = \boldsymbol{S}_b^{r\theta}\boldsymbol{F}_a^{(b)} \tag{6.9}$$

where the transformation operator is

$$\boldsymbol{S}_b^{r\theta} = \begin{bmatrix} \sin(\gamma+\alpha) & -\cos(\gamma+\alpha) \\ \cos(\gamma+\alpha) & \sin(\gamma+\alpha) \end{bmatrix} = \begin{bmatrix} \cos(\psi-\theta) & -\sin(\psi-\theta) \\ \sin(\psi-\theta) & \cos(\psi-\theta) \end{bmatrix} \tag{6.10}$$

which follows from the geometric relations illustrated in Figure 6.1, namely:

$$\gamma + \alpha + \psi - \theta = \frac{\pi}{2} \tag{6.11}$$

In this transformation, the coupling of aerodynamics and gravitational orbital kinematics finally appears.

The 2D equations of motion for the orbiting body in the rotating body frame are

$$\begin{aligned} \boldsymbol{F}_g^{(b)} + \boldsymbol{F}_a^{(b)} &= m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{w} + pv - qu \end{bmatrix} \\ &= m \begin{bmatrix} \dot{u} + qw \\ \dot{w} - qu \end{bmatrix} \end{aligned} \tag{6.12}$$

$$M_{a_y}^{(b)} = I_{yy}\dot{q} \tag{6.13}$$

where unit vectors have been now been omitted. Variable $q = \dot{\psi}$ is the orbital-inertial pitch rate, u and w are body frame components of velocity $\boldsymbol{V}$, and $\boldsymbol{F}_g^{(b)}$ and $\boldsymbol{F}_a^{(b)}$ are gravitational and aerodynamic forces in the body frame, respectively. The products rv and pv in Equation 6.12 are eliminated when enforcing 2D dynamics, as the component of velocity in the y-direction v and body angular rates about the x- and z-directions p and r[1] are zero. The moment equation features only the scalar variable $M_{a_y}^{(b)}$, which is the sum of all moments acting on the body in the body frame. Here, again, no gravitational contribution will be present. Thus, $M_{a_y}^{(b)}$ is entirely a function of aerodynamics and acts only to rotate the body about its center of mass, as was the case in Chapters 3 and 4 for a single degree-of-freedom system.

[1] This yaw rate r should not be confused with the first component of the position $\boldsymbol{r}$. The yaw rate is unused in this work and is merely shown here along with the roll rate p for completeness of derivation.

In the interest of developing more natural state variables, body linear velocities shall be defined in terms of speed V and angle of attack α as

$$u = V\cos(\alpha)$$

$$w = V\sin(\alpha)$$

which is typical of atmospheric aircraft dynamics formulations[65]. The body frame-expressed gravity vector $\boldsymbol{F}_g^{(b)}$ can be obtained using Equations 6.2 and 6.7 as

$$\boldsymbol{F}_g^{(b)} = [\boldsymbol{S}_b^{r\theta}]^{\mathrm{T}}\boldsymbol{F}_g^{(r\theta)} = \frac{\mu m}{r^2}\begin{bmatrix} -\cos(\psi-\theta) \\ \sin(\psi-\theta) \end{bmatrix} \tag{6.14}$$

Equation 6.12 can now be written in component form as

$$-\frac{\mu m}{r^2}\cos(\psi-\theta) + F_{a_x}^{(b)} = m[\dot{V}\cos(\alpha) - V\dot{\alpha}\sin(\alpha) + qV\sin(\alpha)] \tag{6.15}$$

$$\frac{\mu m}{r^2}\sin(\psi-\theta) + F_{a_z}^{(b)} = m[\dot{V}\sin(\alpha) + V\dot{\alpha}\cos(\alpha) - qV\cos(\alpha)] \tag{6.16}$$

Now note that speed may be defined in terms of perifocal variables as

$$V = \sqrt{\dot{r}^2 + r^2\dot{\theta}^2} \tag{6.17}$$

which follows from the definition of orbital position given in Equation 6.3. With this definition, the scalar acceleration $\dot{V}$ may be expanded as

$$\begin{aligned} \dot{V} &= \dot{V}_h\frac{\partial V}{\partial V_h} + \dot{\Omega}\frac{\partial V}{\partial \Omega} + V_h\frac{\partial V}{\partial r} \\ &= \frac{\dot{V}_h V_h + \dot{\Omega}\Omega r^2 + V_h\Omega^2 r}{V} \end{aligned} \tag{6.18}$$

where the following identities are used:

$$\begin{aligned} V_h &\equiv \dot{r} \\ \Omega &\equiv \dot{\theta} \end{aligned} \tag{6.19}$$

Substituting this expression for $\dot{V}$ into Equations 6.15 and 6.16 and collecting derivative terms yields

$$a_{11}\dot{\alpha} + a_{14}\dot{V}_h + a_{16}\dot{\Omega} = -\frac{\mu m}{r^2}\cos(\psi-\theta) + \frac{1}{m}F_{a_x}^{(b)} - \frac{V_h}{V}\Omega^2 r\cos(\alpha) - qV\sin(\alpha) \tag{6.20}$$

$$a_{21}\dot{\alpha} + a_{24}\dot{V}_h + a_{26}\dot{\Omega} = \frac{\mu m}{r^2}\sin(\psi - \theta) + \frac{1}{m}F_{a_z}^{(b)} - \frac{V_h}{V}\Omega^2 r\sin(\alpha) + qV\cos(\alpha)] \tag{6.21}$$

where

$$\begin{aligned} a_{11} &= -V\sin(\alpha) \quad a_{14} = \frac{V_h}{V}\cos(\alpha) \quad a_{16} = \frac{\Omega r^2}{V}\cos(\alpha) \\ a_{21} &= V\cos(\alpha) \quad\quad a_{24} = \frac{V_h}{V}\sin(\alpha) \quad a_{26} = \frac{\Omega r^2}{V}\sin(\alpha) \end{aligned} \tag{6.22}$$

The state vector is now defined as

$$\boldsymbol{s} = \begin{bmatrix} \alpha \\ q \\ h \\ V_h \\ \theta \\ \Omega \end{bmatrix} \tag{6.23}$$

where $h = r - r_e$ is the altitude above the Earth (r_e being the radius of an assumed–spherical Earth), which is chosen instead of the direct variable r as a matter of slight convenience.

Equation 6.8 may be written in the state variable component form

$$\dot{V}_h = \frac{F_{a_x}^{(b)}}{m}\cos(\psi - \theta) - \frac{F_{a_z}^{(b)}}{m}\sin(\psi - \theta) - \frac{\mu}{(h + r_e)^2} + (h + r_e)\Omega^2 \tag{6.24}$$

$$\dot{\Omega} = \frac{F_{a_x}^{(b)}}{m(h + r_e)}\sin(\psi - \theta) + \frac{F_{a_z}^{(b)}}{m(h + r_e)}\cos(\psi - \theta) - \frac{2V_h\Omega}{h + r_e} \tag{6.25}$$

Treating ψ as a state has been avoided since it relates to other states and γ via Equation 6.11 along with the relation

$$\gamma = \tan^{-1}\left(\frac{V_h}{r\Omega}\right) = \tan^{-1}\left(\frac{V_h}{\sqrt{V^2 - V_h^2}}\right) \tag{6.26}$$

With Equations 6.20 and 6.21, Equations 6.24 and 6.25, Equation 6.13, and the two identities for Ω and V_h (the latter of which also equals $\dot{h}$ since the radius of the earth is assumed constant), the system is now overdetermined by one equation. The inversion procedure that places the system in a closed form for $\dot{\boldsymbol{s}}$ by solving for $\dot{\alpha}$ may use either Equation 6.20 or Equation 6.21. If Equation 6.20 is used, the final equation for $\dot{\alpha}$ will contain a term with $1/\sin(\alpha)$ whereas the use of Equation 6.21 results in $1/\cos(\alpha)$. The latter is slightly preferable in that most motion is more likely to be

centered about $\alpha = 0$ (at least if any assumption at all can be made in this regard). The final set of state equations is

$$\dot{s}_1 = \dot{\alpha} = \left[(V_h^2 - r^2\Omega^2)\sin(\alpha) + 2rV_h\Omega\sin(\alpha)\tan(\alpha)\right]\frac{F_x}{mV^3} + \frac{F_z\cos(\alpha)}{mV} + \frac{\mu\Omega}{rV^2} + q \tag{6.27}$$

$$\dot{s}_2 = \dot{q} = \frac{M_y}{I_{yy}} \tag{6.28}$$

$$\dot{s}_3 = \dot{h} = V_h \tag{6.29}$$

$$\dot{s}_4 = \dot{V}_h = \left[V_h\cos(\alpha) + r\Omega\sin(\alpha)\right]\frac{F_x}{mV} + \left[V_h\sin(\alpha) - r\Omega\cos(\alpha)\right]\frac{F_z}{mV} - \frac{\mu}{r^2} + r\Omega^2 \tag{6.30}$$

$$\dot{s}_5 = \dot{\theta} = \Omega \tag{6.31}$$

$$\dot{s}_6 = \dot{\Omega} = \left[r\Omega\cos(\alpha) - V_h\sin(\alpha)\right]\frac{F_x}{mVr} + \left[V_h\cos(\alpha) + r\Omega\sin(\alpha)\right]\frac{F_z}{mVr} - \frac{2V_h\Omega}{r} \tag{6.32}$$

where care has been taken to express them in terms of α to emphasize the presence of body dynamics, which are introduced primarily through α. Notation of source functions $F_{a_x}^{(b)}$, $F_{a_z}^{(b)}$ and $M_{a_y}^{(b)}$ has been reduced to F_x, F_z, M_y, respectively, with the understanding that each is a contribution of aerodynamic force or moment in the body frame. They are assumed to be functions of states V_h, Ω, α and h only. The conclusions of Chapter 5 make the case for excluding q and an assumption that the atmosphere is angularly symmetric will suffice for now in order to ignore dependence on θ. Since V_h and Ω are components of velocity in the r-θ frame, dependence on each variable separately is unnecessary, and the speed V (Equation 6.17) can be used instead. This is a more natural arrangement of variables considering that the gas dynamic portion of the system is easily expressed in terms of a speed and incidence angle given by the relations of Equations 6.14. Each source function is then a scalar-valued function of three variables which requires a parameter scan in each variable in order to generate the function.

6.2 Problem Construction Details

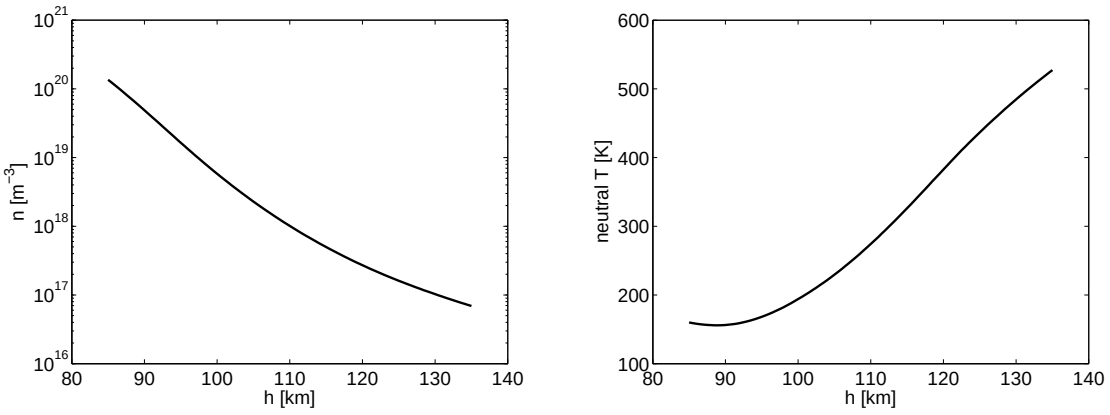
Source functions are defined as functions of altitude h, rather than the individual variables of interest such as density and temperature. The reason this is done is that density and temperature are typically assumed to vary independently with altitude, e.g. $n(h)$, $T(h)$. Thus any function that

depends on $n(h)$ and $T(h)$ may be parameterized by h. This simplifies the definitions of source functions greatly, which is useful in that it ensures the dimension of the parameter space does not grow unnecessarily. Density and temperature (as well as gas composition in terms of resident species and their partial densities) may vary with h as well as θ. The dependence on θ stems from a number of physical phenomena such as differences in day and night sides of the Earth and various space weather conditions. Although some atmospheric models may describe properties as a function of angular location, this work uses the MSISE[56] model with the assumption that density and temperature have only radial dependence.

To prepare the DSMC parameter scan, a configuration for each point in the parameter space is generated. A choice of parameter ranges is made which covers the space of interest. For state α, the range is naturally periodic as $\alpha \in [-\pi, \pi]$. For pseudo-state V and state h, ranges should be chosen which cover an expected set of possible orbital trajectories. In the most general sense, the range in altitude may start at the ground and end at the maximum design apoapsis of an orbit. Speeds would then range from zero to the maximum speed experienced in the design orbit. Since DSMC is ill-suited for very low Kn flows, a different numerical method or hybrid code which joins DSMC and a continuum CFD method would be required where $\mathrm{Kn} \lesssim 0.001$. For Earth, this means that DSMC is best used for simulations above roughly 80km in altitude (the upper mesosphere or lower thermosphere). A value near this is an acceptable lower limit for h, with the speed of a circular orbit at this altitude being a good choice for the lower limit of V. Upper limits may be chosen then with the only restriction being the computational effort required to resolve the parameter space within these limits to an acceptable accuracy.

Note that an even, or linearly spaced grid along the h-direction may require an unacceptably high number of points in order to resolve the exponential nature of density with altitude. A non-uniform grid which is weighted towards the low end of the range of h would be ideal. However, an unevenly spaced grid may create difficulty in using the source function data which is to be generated on this grid for approximation methods, should they be required. In the case of application of the data directly in numerical integration of the equations of motion, as would often be the goal, such

a grid would only require non-uniform trilinear interpolation. Considering the grid spacing in h is particularly necessary because forces, moments and energy transfer tend to be much more sensitive to changes in density than in speed. The range of speeds experienced by an orbiting object in the thermosphere may be less than 1.0 km/s; while the range in density may span many orders of magnitude, giving perhaps: $0.01 \lesssim \mathrm{Kn} \lesssim 1000$. Figure 6.2 gives an example range of the first 50km of the thermosphere where density changes by three orders of magnitude while temperature only approximately triples.



(a) Number density.

(b) Neutral temperature.

Figure 6.2: MSISE[56] Low thermospheric properties ($85\mathrm{km} \leq h \leq 135\mathrm{km}$).

This problem is unique to the continuum-rarefied transition flow regime. In continuum fluids problems, forces are typically more sensitive to speed if the aerodynamic force relation

$$F = \frac{1}{2} m_g n V^2 S C_F \tag{6.33}$$

is taken as a general rule. Quadratic dependence on speed contrasts with linear dependence on density. However, this equation assumes that no serious changes in gas mechanics occur regardless of the value of n. A gas dynamics problem such as an atmospheric egress or entry of an object experiences a drastic shift in physical mechanisms that dominate the flow as it moves through the transition region. At the top of the region, free-molecular flow eliminates the ability of the system to dissipate energy and transfer momentum within the gas itself. A body in FM flow will

not experience damped motion, as was shown in Chapter 3, and the flow has no knowledge of the existence of the body prior to impacting it. As the body descends, it experiences collisional flow, allowing shocks to form and the body to experience a coupling with the gas. The presence of more molecules in closer vicinity which arises from increasing density causes exponentially more collisions to occur, affecting the flow field and its interaction with the body faster than increases in speeds or temperature. Sections 6.4 and 6.6 demonstrate this sensitivity using real source functions computed for bodies in the thermosphere, which perhaps best characterizes continuum-rarefied transition for spacecraft.

6.3 Computational Setup

Earth's lower thermosphere is most representative of continuum-rarefied transition. This region also presents the greatest difficulty in performing numerical simulations, and is thus an appropriate choice for the demonstration of DSMC as a method suited for transition flows. The parameter space and configuration parameters have been chosen with the goal of performing high-fidelity simulations of spacecraft body dynamics under the influence of transition flow.

Computation of source functions $F_x(V, \alpha, h)$, $F_z(V, \alpha, h)$ and $M_y(V, \alpha, h)$ requires a large scan over each of the directions V, α and h. Each point in the space $[V\ \alpha\ h]$ requires a single DSMC simulation. To provide these functions at high resolutions, this scan becomes very costly. Prior to applying any further reductions of order or approximations that may reduce the computational requirements of this scan, it must be performed in order to establish an authoritative set of data with which to compare future methods. For each set of data, only the body geometry and surface boundary conditions need remain constant. For higher-order simulations, boundary conditions would likely require promotion to state variables; which would, unfortunately, further increase the dimension of the parameter space. In the interest of keeping this space minimal, all segments of the surface of the objects used in simulations were set to use the standard Maxwellian diffuse gas-surface interaction model described in Section 2.1.1, with a single constant surface temperature T_s. A surface defined in this way may be described as being adiabatic. The remaining configurations

for each DSMC simulation (or "run") were constructed to minimize total run time to completion without sacrificing more than a small amount of accuracy.

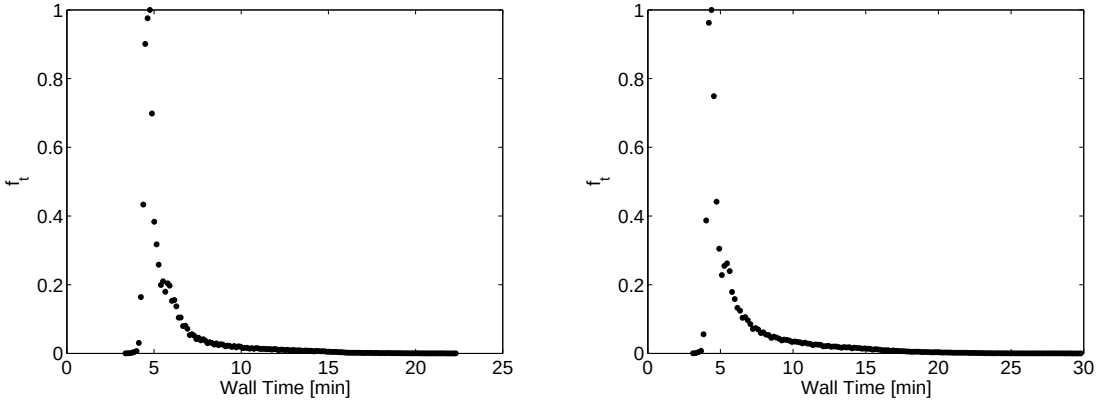
Given a specific choice of object, each run was chosen to use the configuration parameters given in Table 6.1. The space covered by the entire set of runs is given as the lower section of Table 6.1. Each direction is spaced evenly. Choices for the sizing of this space derive from the arguments in Section 6.2. The range in h somewhat drives the remainder of the configuration. By choosing a region of the atmosphere, the necessary density and temperatures are obtained via linear interpolation of the MSISE model. The choice of $h \in [85, 135]$km assures flight within lower thermosphere with free stream Knudsen numbers between approximately 0.01 and 20 for one-meter-sized objects. Above 135km, diatomic nitrogen no longer dominates the species composition as dramatically. Below roughly 80km, the computational burden per run quickly increases to unacceptable levels, as DSMC is not the optimal method for continuum flow. In the interest of maximizing performance, the simulations are restricted to this region such that the flow can be assumed to consist of only N_2. These altitudes also provide a range of V, if the rough assumption of a circular orbit is made. Because density changes exponentially with altitude, the highest resolution is given to the h-direction. The V-direction is given the lowest resolution because it is expected that source functions have the weakest sensitivity to V, and also that V itself changes very little over the range of altitudes chosen. The range in α is naturally periodic and is required to span the full range of possible angles in order for the data set to be capable of describing body dynamics in orbit. A high resolution in α is desirable, particularly for bodies without symmetry and bodies with geometric features that differ greatly in the angular direction. For example, a spacecraft with a long protrusion such as a boom might require high resolution in α close to the angular location of the boom. The two bodies that were used in this study were introduced in Chapters 3 and 4: the NACA-0012 airfoil and the Arbject. The Arbject in this case is given by the parameters: $L = 0.75$m, $\theta = 30$deg, $R = 0.15$m, $d = 0.3m$, $\omega = 6$rad/m.

Table 6.1: Voldipar DSMC configuration variable values and parameters common to all runs of a large parameter scan for generating dynamic source functions.

Description	**Symbol**	**Value**	**Unit**
Domain length	L_x	2.00	m
Domain height	L_y	2.00	m
Num. cells in x	N_{cx}	160	
Num. of cells in y	N_{cy}	160	
Num. of subcells per cell (each dir.)	N_{sc}	2	
Variable time step adjustment	γ_t	5.5	
Time to steady state (imposed)	t_{ss}	0.0010	s
Time to end	$t_{\max}$	0.0025	s
Num. time steps per sample	N_{sam}	4	
Temp. of body surface (all segments)	T_s	300	K
Gas molecular mass	m_g	4.65×10^{-26}	kg
Gas viscosity power	ω	0.74	
Gas reference temp.	T_{ref,N_2}	273	K
Gas reference diameter	d_g	4.17×10^{-10}	m
Gas rotational deg. of freedom	ζ	2	
Rotational relaxation constant	Z_{rot}	1/5	
Voxel resolution (Voldipar only)	n_{vox}	700	
Range of V	-	$[7500, 8000]$	m/s
Number of points in V-direction	N_v	41	
Range of α	-	$[-\pi, \pi]$	
Number of points in α-direction	N_α	121	
Range of h	-	$[85, 135]$	km
Number of points in h-direction	N_h	181	

6.4 Computed Source Functions for Thermospheric Flight

With the parameter space resolutions chosen, a total of 897,941 individual runs were required to complete the set of data required to define source functions. In order to handle such a computation, a super cluster was employed to spread the set over as many processors as possible. The University of Colorado JANUS super cluster was used to generate all necessary source function data for the NACA-0012 airfoil and the Arbject. For the code resolutions used (i.e. number of cells and voxels given in Table 6.1), each run required between three and twenty-two minutes of time to complete for the airfoil and between three and thirty minutes for the Arbject. Normalized distributions of wall-clock total computation time per run are plotted for both geometries in Figure 6.3 which each show a most probable run time of approximately 4.5 mins. The short times are due to the computations being mostly within regions of Kn $\gtrsim 0.1$ where collisions begin to become relatively scarce, alleviating much of the computational burden of that part of the overall DSMC algorithm.

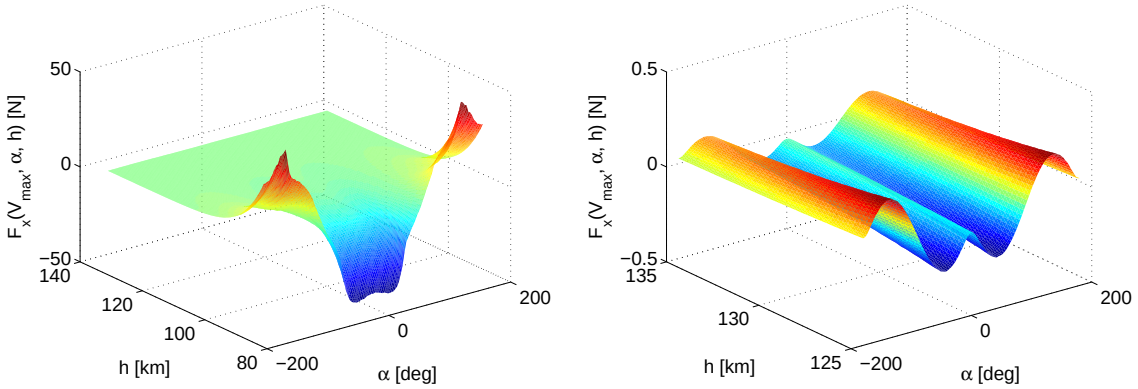


(a) Airfoil. (b) Arbject.

Figure 6.3: Normalized distribution of total DSMC computation time per run over the full set of runs which comprise the parameter space defined by Table 6.1.

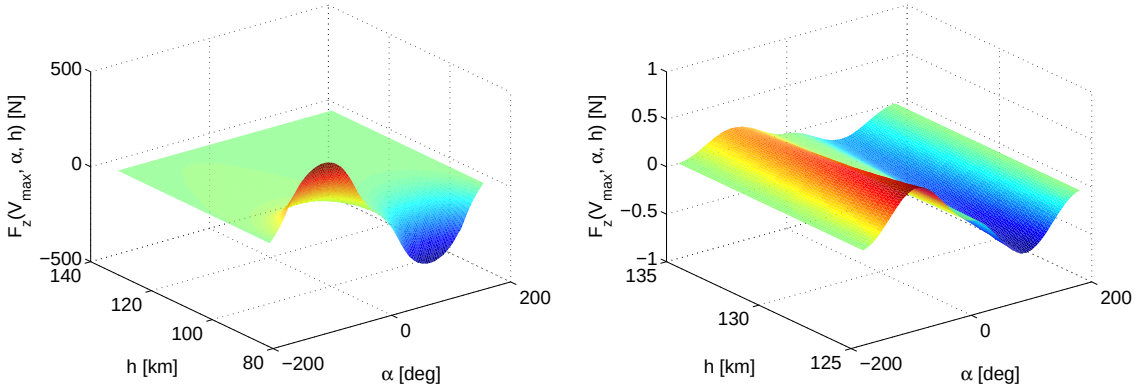
6.4.1 Airfoil

A sampling of computed source functions for the NACA-0012 airfoil is given in Figures 6.4 to 6.7. Each plot is a slice at the maximum velocity of the parameter space, $V_{max} = 8000\text{m/s}$. The highly exponential nature of each function in density (and thus in h), makes visualizing qualitative features difficult if the entire range of h is plotted. Thus, each function is shown over the top ten kilometers of altitude in addition to the entire range of h.



(a) Full range. (b) Top 20% of the computed range.

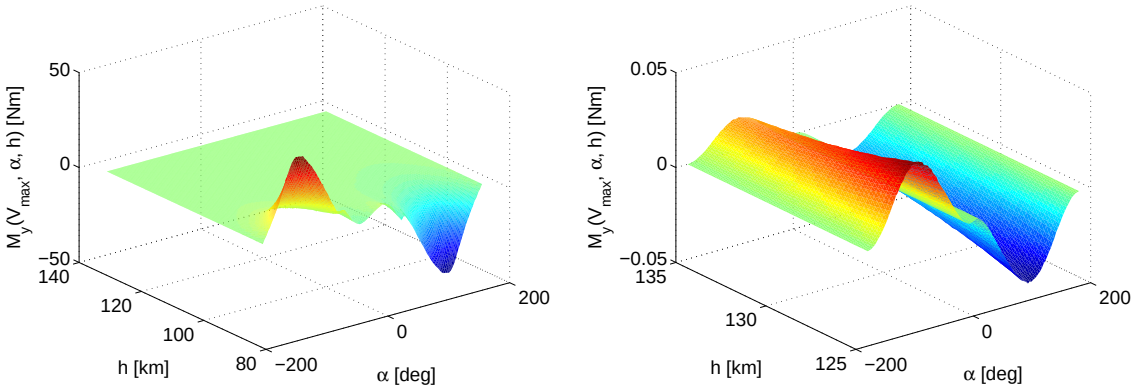
Figure 6.4: Source function for body x-force: Sample slices $F_x(V = 8000\text{m/s}, \alpha, h)$.



(a) Full range. (b) Top 20% of the computed range.

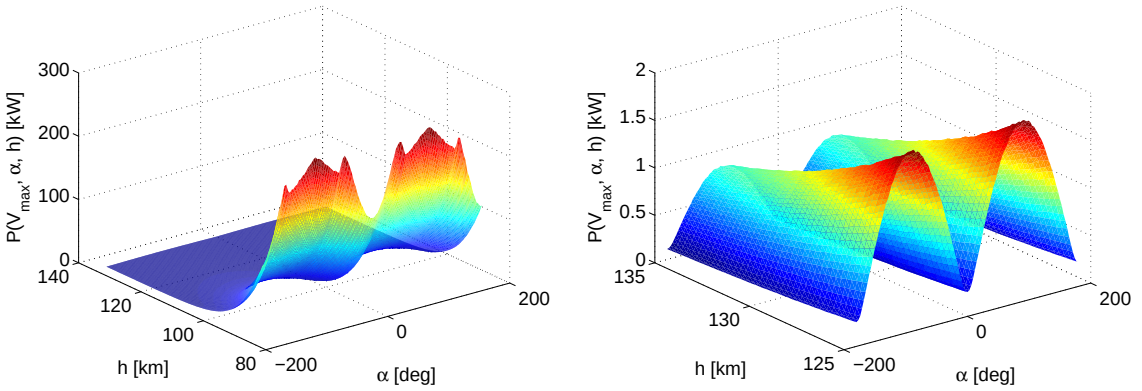
Figure 6.5: Source function for body z-force: Sample slices $F_z(V = 8000\text{m/s}, \alpha, h)$.

All functions exhibit the expected symmetries about $\alpha = 0$. At the lowest altitudes, M_y



(a) Full range. (b) Top 20% of the computed range.

Figure 6.6: Source function for body moment about y: Sample slices $M_y(V = 8000\text{m/s}, \alpha, h)$.



(a) Full range. (b) Top 20% of the computed range.

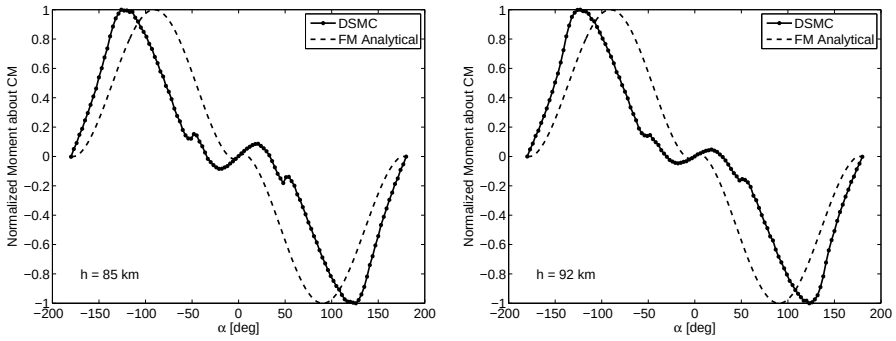
Figure 6.7: Source function for total power to the body's surface: Sample slices $P(V = 8000\text{m/s}, \alpha, h)$.

possesses some higher frequencies (seen in detail in Figure 6.8a) for small angles. It is clear that all functions experience a smoothing of these high frequencies as h approaches h_{max}. The maximum altitude h_{max} does not constitute a definite free-molecular regime since Kn ≈ 20 at that altitude can still be considered transitional. Nevertheless, detailed features tend to smooth as the body experiences flow consisting of fewer collisions. Perhaps the most relevant function is M_y, since it contributes the most directly to the body's rotation. A visualization of the effect of moving through the transition regime can be made by scaling M_y in a special way such that for a given speed (V_{max})

and at every point in h, the function is normalized by

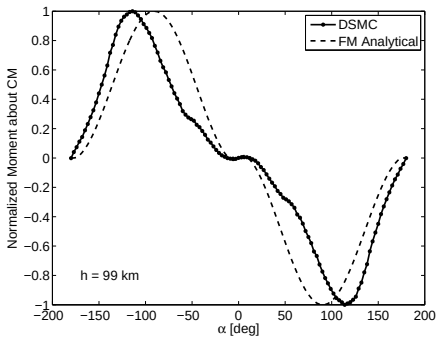
$$M'_y(V_{max}, \alpha, h) = \begin{cases} \frac{M_y(V_{max},\alpha,h)}{\max(M_y(V_{max},\alpha,h))} & \text{for } M_y(V_{max}, \alpha, h) \geq 0 \\ \frac{-M_y(V_{max},\alpha,h)}{\min(M_y(V_{max},\alpha,h))} & \text{for } M_y(V_{max}, \alpha, h) < 0 \end{cases}$$

This function is plotted for a set of chosen values of h at V_{max} in Figure 6.8. A roughly equivalent free-molecular solution is shown on each plot. This solution was obtained via the FM-panel method discussed in Section 3.2. As the object ascends in h, its high frequency features are smoothed and the scaled function approaches that of the free-molecular solution. The qualitative features of these functions as they approach the FM limit agree with those given in Section 3.2. The central unstable region discussed in Section 3.3.1 is present.

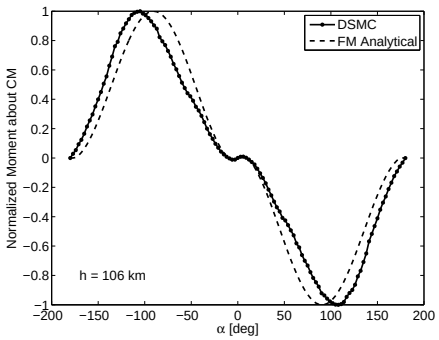


(a) $h = 85$km.

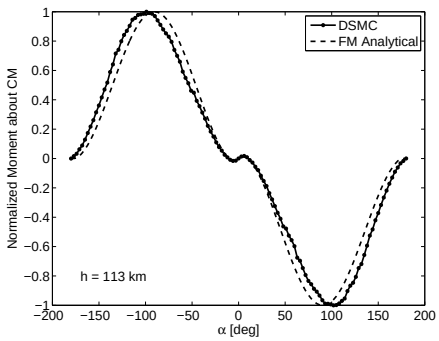
(b) $h = 92$km.



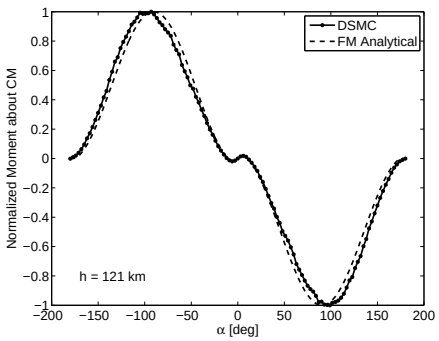
(c) $h = 99$km.



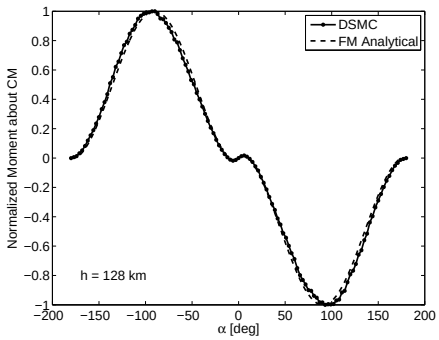
(d) $h = 106$km.



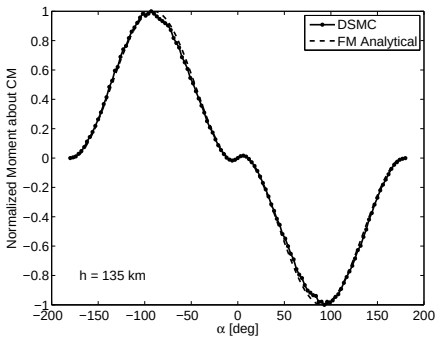
(e) $h = 113$km.



(f) $h = 121$km.



(g) $h = 128$km.

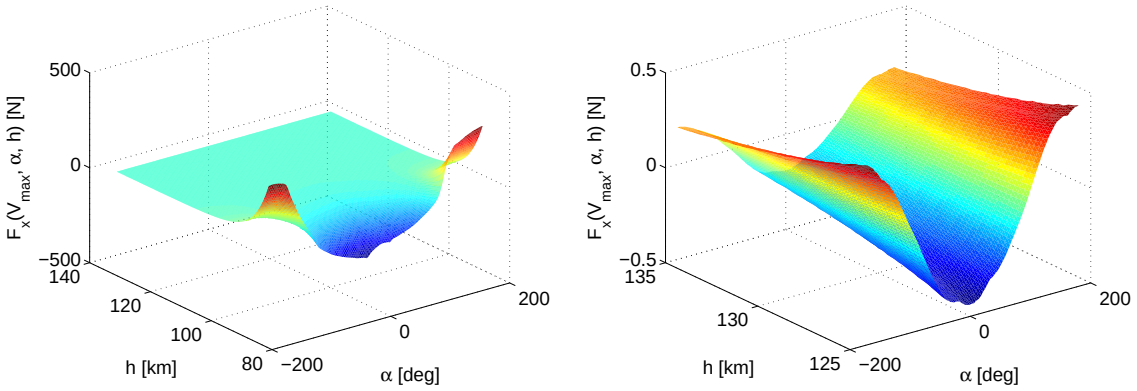


(h) $h = 135$km.

Figure 6.8: Specially-normalized moment source function for the airfoil shown at a small set of selected altitudes spanning the entire range of computed altitudes.

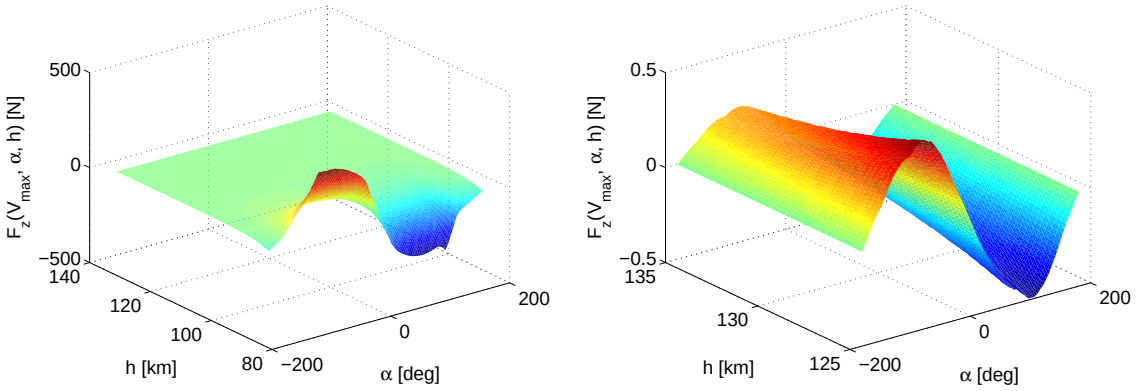
6.4.2 Arbject

Analogous to the presentation of the NACA-0012 airfoil's results, a sampling of computed source functions for the Arbject is given in Figures 6.9 to 6.12. Transition of M_y is visualized for



(a) Full range. (b) Top 20% of the computed range.

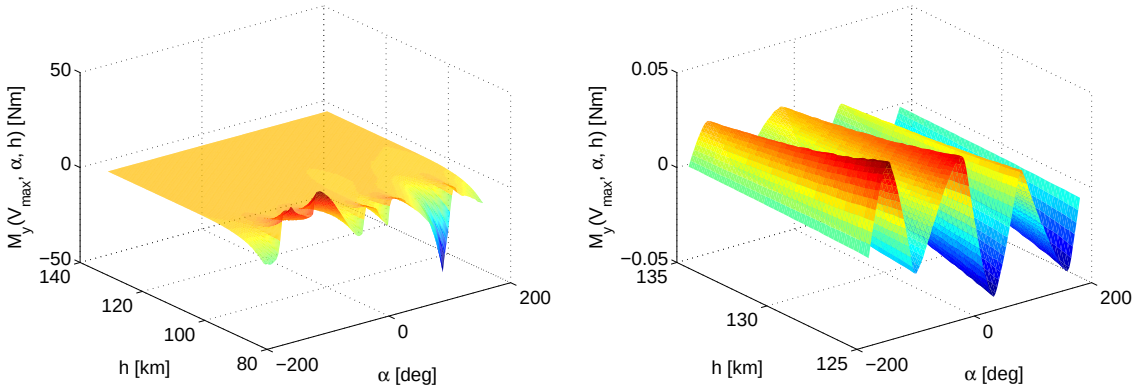
Figure 6.9: Source function for body x-force: Sample slices $F_x(V = 8000\text{m/s}, \alpha, h)$.



(a) Full range. (b) Top 20% of the computed range.

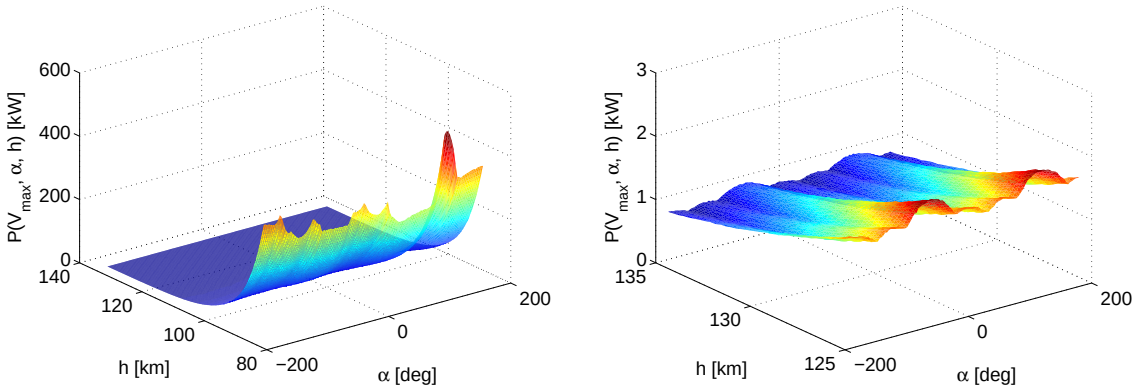
Figure 6.10: Source function for body z-force: Sample slices $F_z(V = 8000\text{m/s}, \alpha, h)$.

the Arbject by plotting Equation 6.34 for a set of chosen values of h at V_{max} in Figure 6.13. Again, the moment function appears to approach a free-molecular solution. The Arbject's geometry, which possesses no symmetry, yields source functions which are unsurprisingly asymmetric in α for low altitudes. However, the eventual limit of M_y shows that it begins to exhibit the property of an



(a) Full range. (b) Top 20% of the computed range.

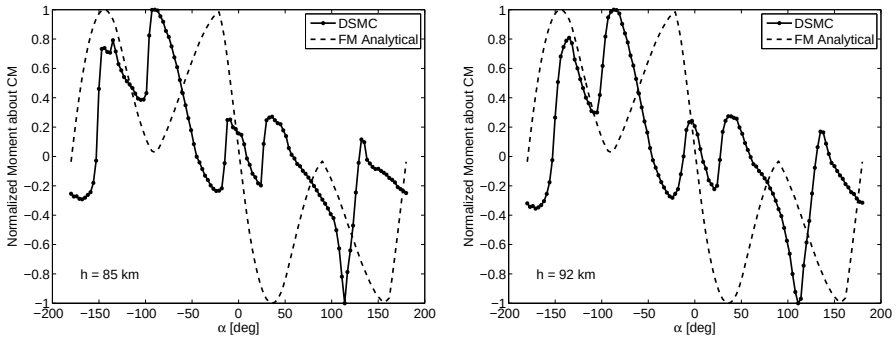
Figure 6.11: Source function for body moment about y: Sample slices $M_y(V = 8000\text{m/s}, \alpha, h)$.



(a) Full range. (b) Top 20% of the computed range.

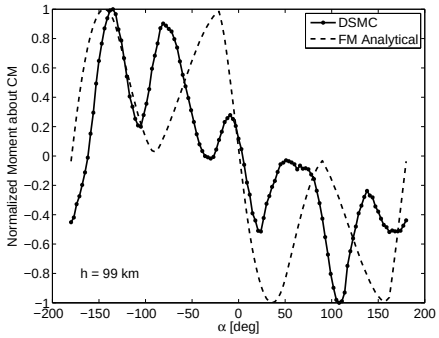
Figure 6.12: Source function for total power to the body's surface: Sample slices $P(V = 8000\text{m/s}, \alpha, h)$.

odd function in α (i.e. $-M_y(\alpha) \approx M_y(-\alpha)$), indicating a further insensitivity to geometry as Kn increases. This property provides some reasoning for the common treatment of objects in free-molecular flow in a purely ballistic manner. A comparison of body dynamics-inclusive modeling with traditional ballistic methods is made in Section 6.6.

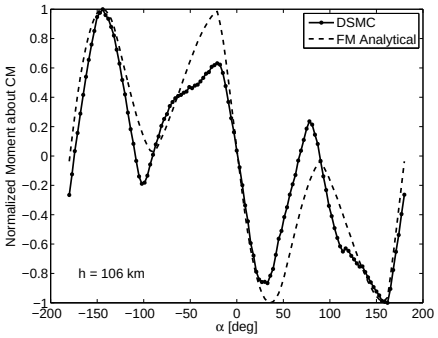


(a) $h = 85$km.

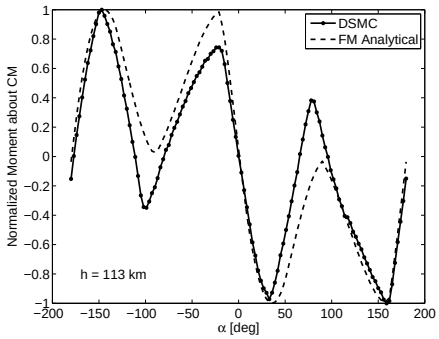
(b) $h = 92$km.



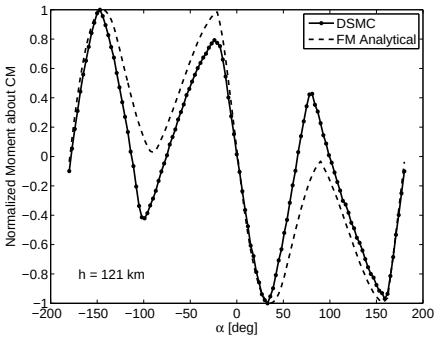
(c) $h = 99$km.



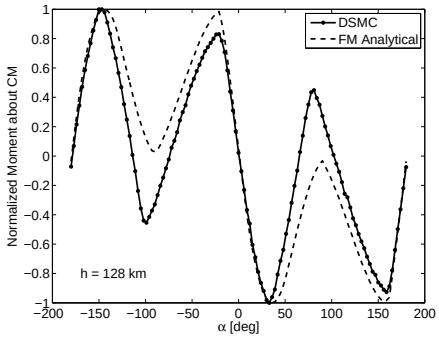
(d) $h = 106$km.



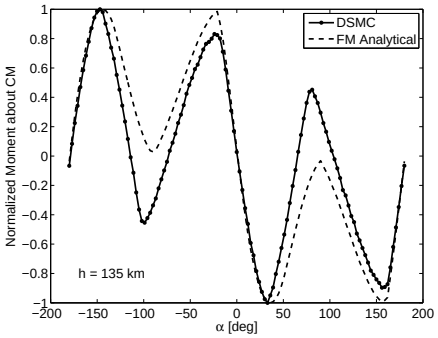
(e) $h = 113$km.



(f) $h = 121$km.



(g) $h = 128$km.



(h) $h = 135$km.

Figure 6.13: Specially-normalized moment source function for the Arbject shown at a small set of selected altitudes spanning the entire range of computed altitudes.

6.5 Full Orbital Simulation Results for Thermospheric Flight

With the appropriate source data obtained, high-fidelity simulations of body dynamics in orbit were made by integrating Equations 6.27 to 6.32. Source functions were evaluated directly by using a trilinear interpolation scheme. These simulations give a time-accurate history of orbital and body motion within the transition region of the lower thermosphere. With the data available, in this configuration, any initial conditions may be chosen which fall within the range of computed data, or which induce motion that does not leave this range. In the case of the altitude h climbing above h_{max}, the source functions are set to zero in the proceeding latter cases for each object, effectively enforcing a free cruise. This assumption is not accurate, as forces are not zero immediately above $h = 135$km. This concession was made for the sake of allowing the simulations to continue unabated. When data is computed up to an altitude where Kn $\gtrsim 100$, free-molecular flow may assumed and the analytical panel method described in Chapter 3 may be employed in-loop to allow the simulation to continue without an upper bound in altitude. For the data provided, h_{max} does not give a free-molecular regime.

When the object descends below h_{min}, the integration stops. Below h_{min} no approximation or concession can truly be made without more data. Once the body enters continuum flow, a set of source function data generated from continuum solvers must be obtained. And although DSMC is capable of extending to continuum-level Knudsen numbers, continuum methods are more computationally efficient in these lower–Kn regions.

A number of cases of interest which illustrate the detailed dynamics of the NACA-0012 airfoil and the Arbject are now presented. For each case, the relevant time histories of V, α, h, q and γ are given first. Next, a set of images are given which show the object's geometry in the Earth-centered, fixed inertial frame, evenly spaced in time over the simulation. Each frame shows the orientation of the object and its body basis vectors (dashed lines), the natural orbital directions $\hat{\boldsymbol{e}}_r$ (pointing outward from the earth) and $\hat{\boldsymbol{e}}_\theta$ and the current velocity vector[2] . The Earth is shown as a solid

[2] The velocity vector may be difficult to see in these figures as many of the orbits presented are very close to circular. In these orbits, the direction of $\boldsymbol{V}$ is extremely close to the orbital $\hat{\boldsymbol{e}}_\theta$ direction.

patch. The history of the orbital trajectory is shown as a line which trails the center of mass, thus allowing direct visualization of orbital decay. The view in each image is of a frame that is centered on the object's center of mass and follows it without rotating, the edges of the frame remaining aligned with the inertial $\hat{\boldsymbol{X}}$ (horizontal, positive to the right) and $\hat{\boldsymbol{Z}}$ (vertical, positive upwards) directions. The center of mass's position is to scale with respect to the Earth, while the geometry itself is exaggerated for visualization purposes.

A set of images are then presented which each plot the distribution of power to the surface. Each segment of the surface is given a color to indicate the value of power it is currently experiencing. The units of the values given on the color bar are in Watts. The GSI model used in these simulations was fully diffuse with full thermal accommodation, and the surface temperature was kept fixed, thus enforcing an unrealistic adiabatic boundary condition. This de-coupling of heat transfer and its effects from the overall dynamic system prevents the simulation from being truly accurate with respect to its energy balance with the gas. Nevertheless, the power distributions given in the following figures illustrate what locations on the object's surface are receiving the most heat.

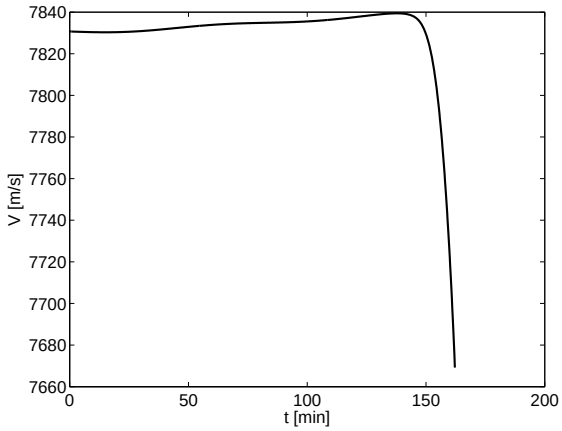
6.5.1 Airfoil

Cases of interest for the NACA-0012 airfoil are summarized in Table 6.2. The body is assumed to be 1m deep and composed of a homogeneous material of density $\rho = 2700\,\mathrm{kg/m^3}$. Relevant

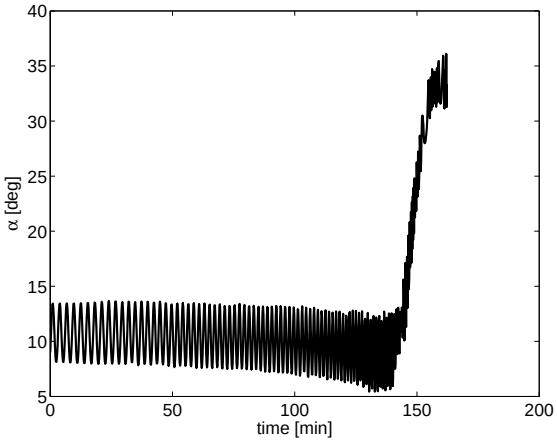
Table 6.2: Initial conditions for each case of the full orbital simulation of the airfoil (*Circular* speed is the speed calculated for a circular orbit at h_0).

Case	V [m/s]	α_0 [deg]	h_0 [km]	q [deg/s]	θ_0 [deg]	γ [deg]
1	Circular	9	122	0	35	0
2	Circular	-95	128	4.6	35	0
4	7840	75	128	0	35	0

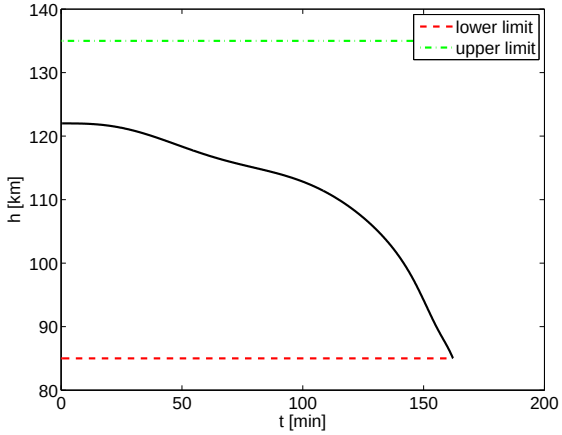
dynamic variable time histories for Case 1 are given in Figure 6.14. Geometric visualization for eight frames, starting at $t = 0$, is shown in Figure 6.15 and corresponding power distribution diagrams are shown in Figure 6.16. Case 2's results are given in Figures 6.17 to 6.19 and Case 3's in Figures 6.20 to 6.22.



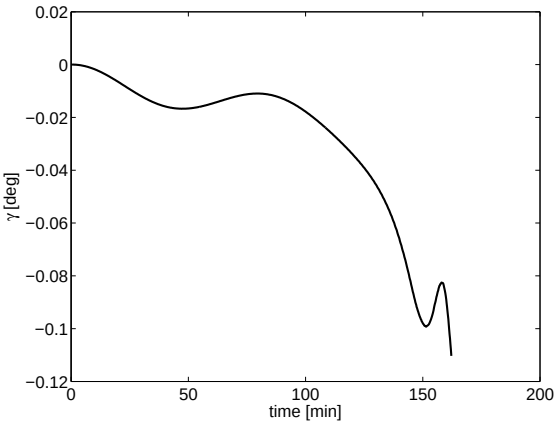
(a) Orbital speed V.



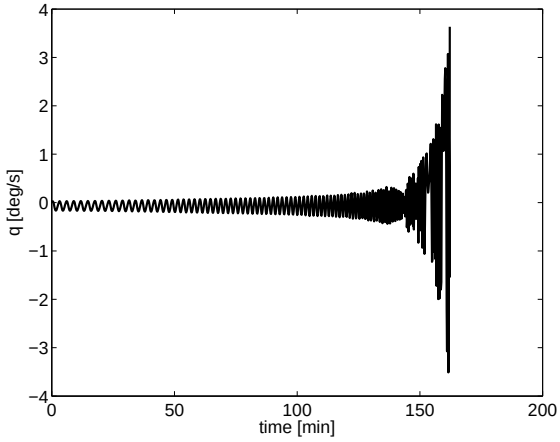
(b) Angle of attack α.



(c) Altitude h.



(d) Flight-path angle γ.



(e) Pitch rate q.

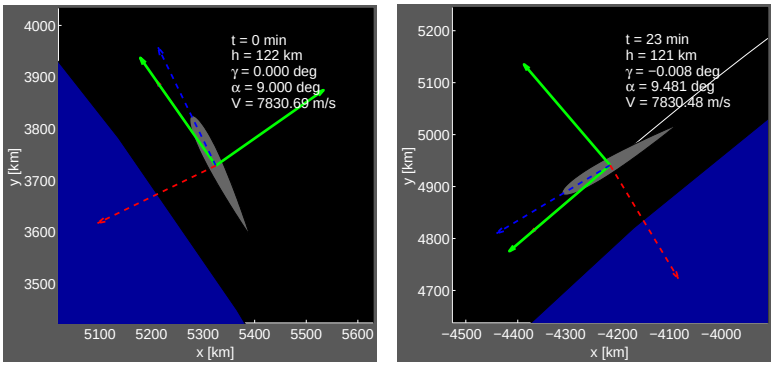
Figure 6.14: Case 1: Time histories of dynamic variables for the airfoil in flight until h_{min} is reached.

Case 1 for the airfoil exhibits behavior similar to that seen in Chapter 3 at first. Figure 6.14b

shows the airfoil oscillating about an angle of attack close to (but not exactly) α_0. The unstable region centered around $\alpha = 0$ in free-molecular flow also exists for high h in these data (seen in Figure 6.8). As the object descends, its oscillation increases in frequency as forces and moment are increasing. Eventually the object switches to a different region of state space where it oscillates between $\alpha \approx 30$deg and $\alpha \approx 35$deg. Its descent occurs more rapidly as the mean angle of attack increases. The negative-definite flight-path angle γ, as seen in Figure 6.14d, shows that the airfoil continues to cut further downward into the atmosphere throughout the simulation.

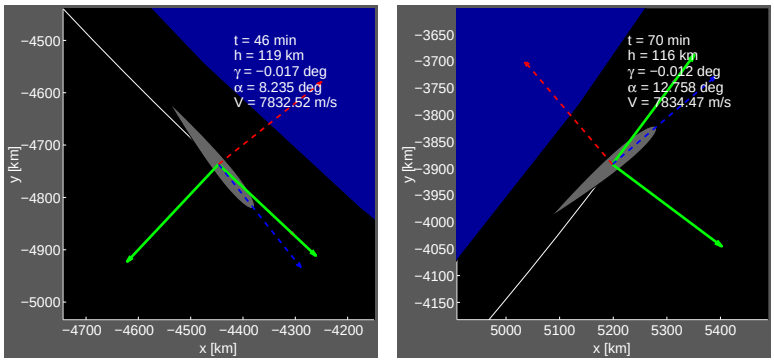
Case 2 begins with the airfoil at a high negative angle of attack and a non-zero initial rotation. Figure 6.17b shows that it experiences roughly ten full rotations and then enters an oscillation which decays in amplitude but grows in frequency as the airfoil descends. Similar to Case 1, the flight-path angle begins a sharp decline once the object passes roughly 110km in altitude–a location very close to Earth's atmospheric turbopause.

Case 3 starts the airfoil in a mildly non-circular orbit. When it ascends above h_{max} aerodynamic forces are ignored. Figure 6.20c shows six full orbits being completed before the final descent occurs, with apoapsis decaying faster than periapsis, indicating a circularization of the orbit. Throughout the simulation the airfoil experiences an unbounded rotation in α. When it descends into the viable region, near periapsis, its pitch rate increases. On its final orbit, the airfoil remains below h_{max} and its rotation rate increases dramatically.



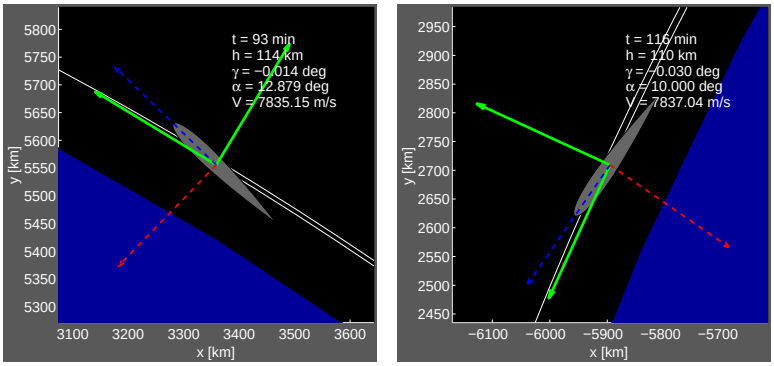
(a) Initial state: 1 of 8.

(b) 2 of 8.



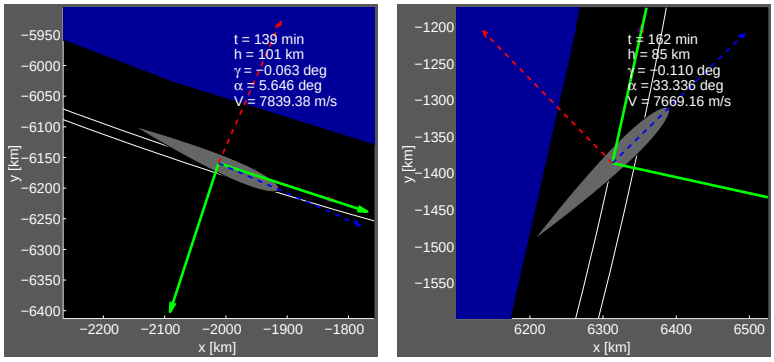
(c) 3 of 8.

(d) 4 of 8.



(e) 5 of 8.

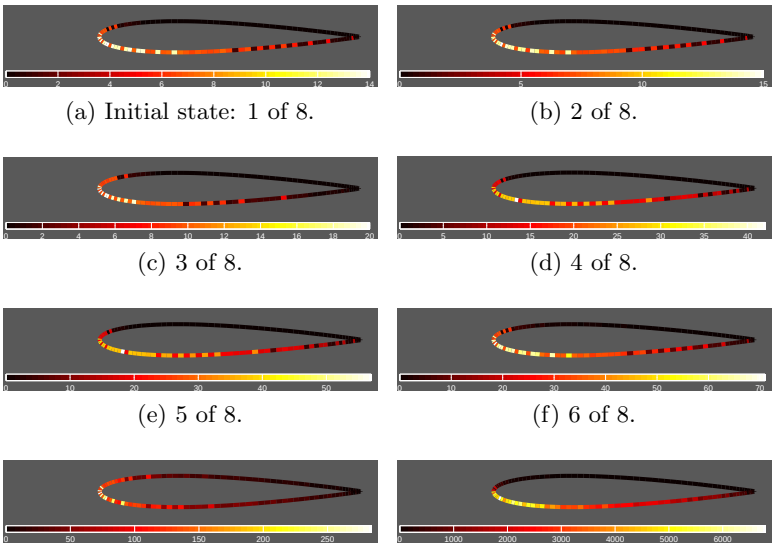
(f) 6 of 8.



(g) 7 of 8.

(h) h_{min} reached: 8 of 8.

Figure 6.15: Case 1: Geometric visualization for the airfoil in flight.



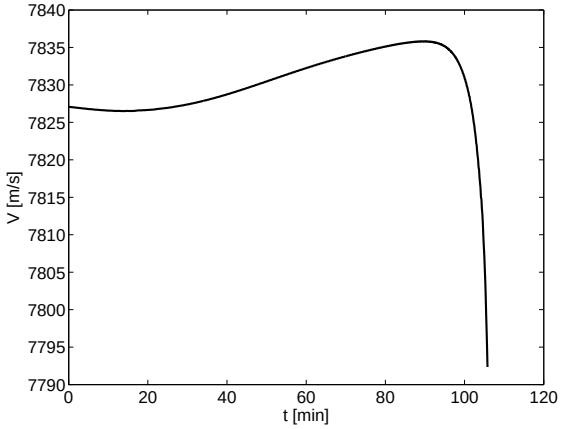
(a) Initial state: 1 of 8. (b) 2 of 8.

(c) 3 of 8. (d) 4 of 8.

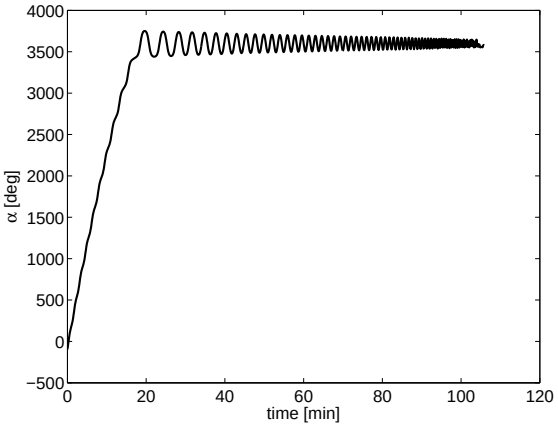
(e) 5 of 8. (f) 6 of 8.

(g) 7 of 8. (h) h_{min} reached: 8 of 8.

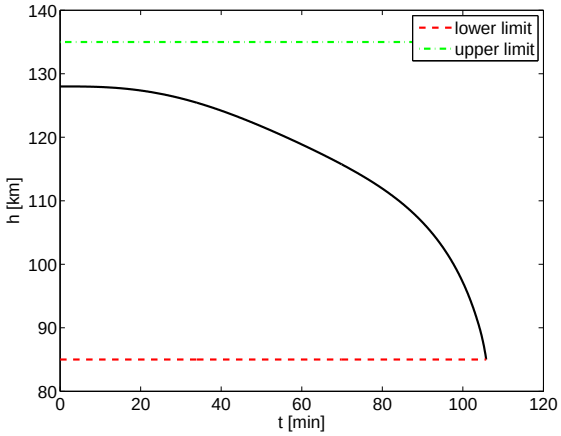
Figure 6.16: Case 1: Visualization of total power distribution over the surface for the airfoil in flight (at the same times as in Figure 6.15).



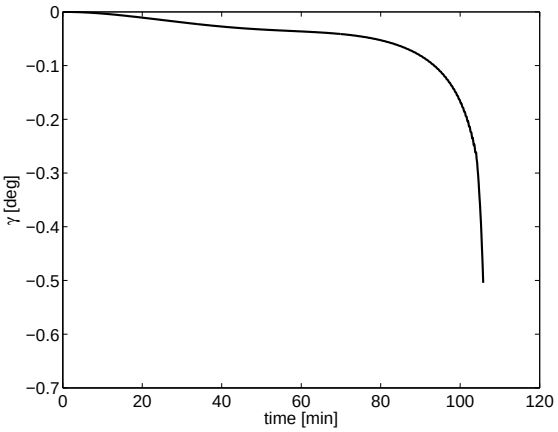
(a) Orbital speed V.



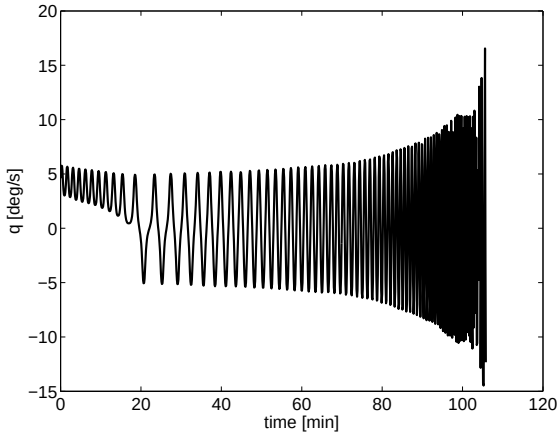
(b) Angle of attack α.



(c) Altitude h.

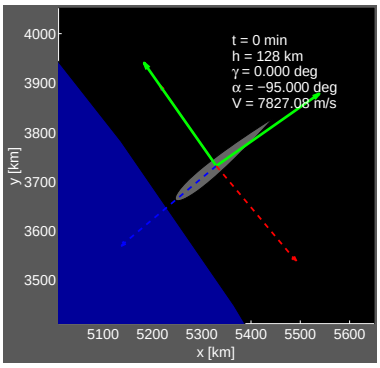


(d) Flight-path angle γ.

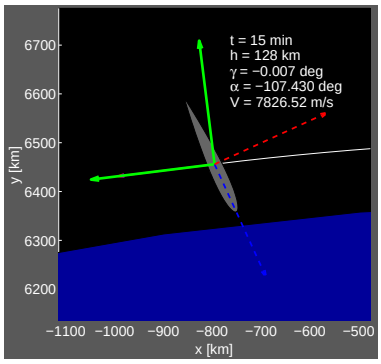


(e) Pitch rate q.

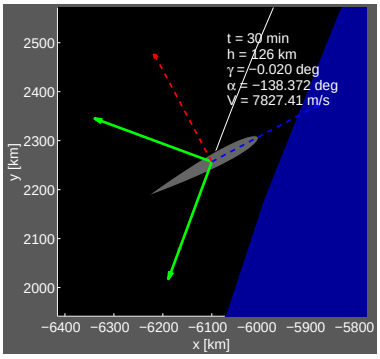
Figure 6.17: Case 1: Time histories of dynamic variables for the airfoil in flight until h_{min} is reached.



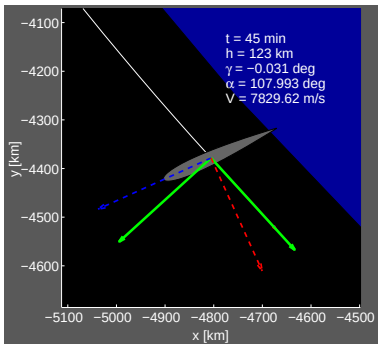
(a) Initial state: 1 of 8.



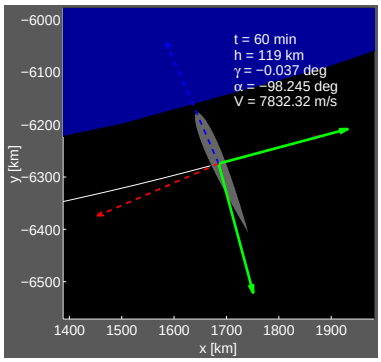
(b) 2 of 8.



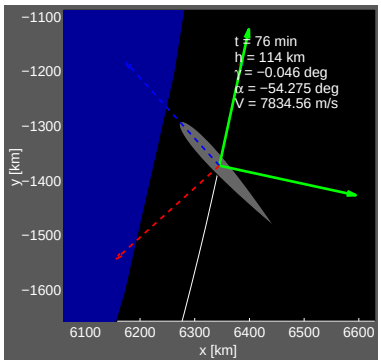
(c) 3 of 8.



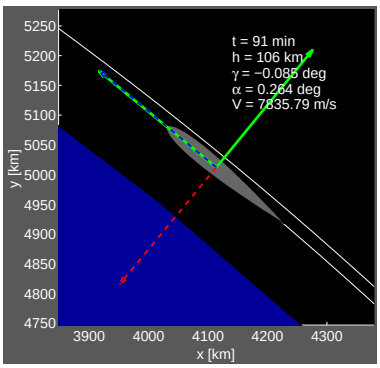
(d) 4 of 8.



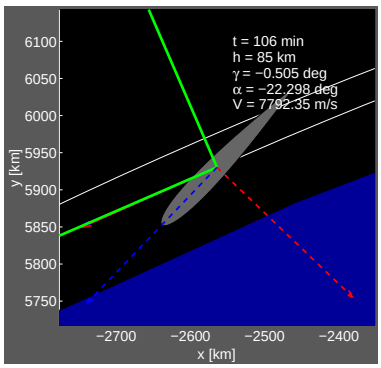
(e) 5 of 8.



(f) 6 of 8.

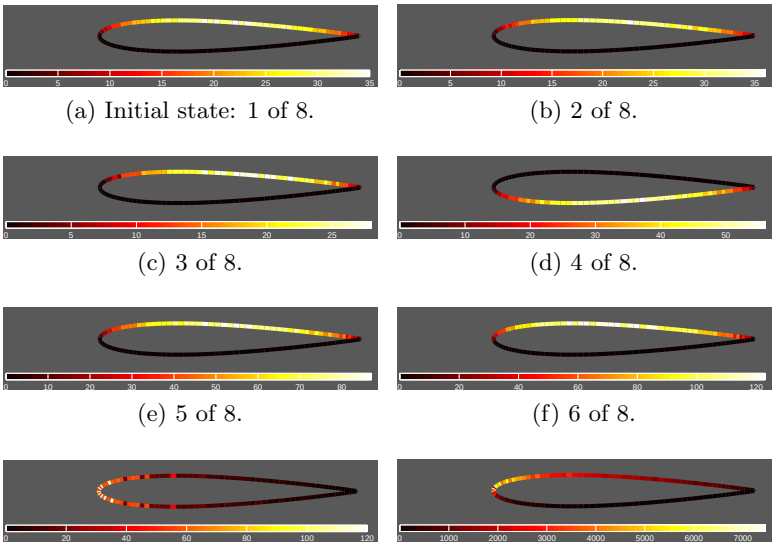


(g) 7 of 8.



(h) h_{min} reached: 8 of 8.

Figure 6.18: Case 2: Geometric visualization for the airfoil in flight.



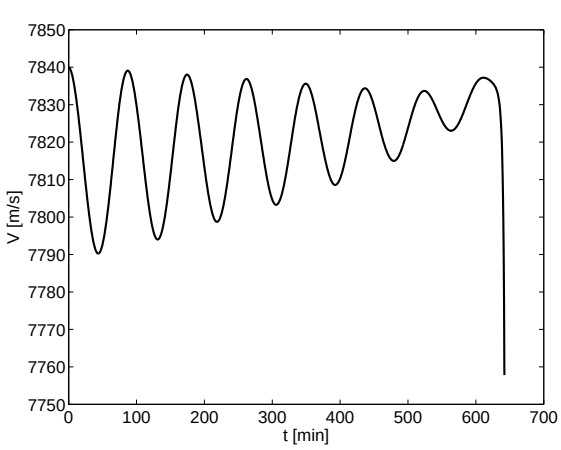
(a) Initial state: 1 of 8. (b) 2 of 8.

(c) 3 of 8. (d) 4 of 8.

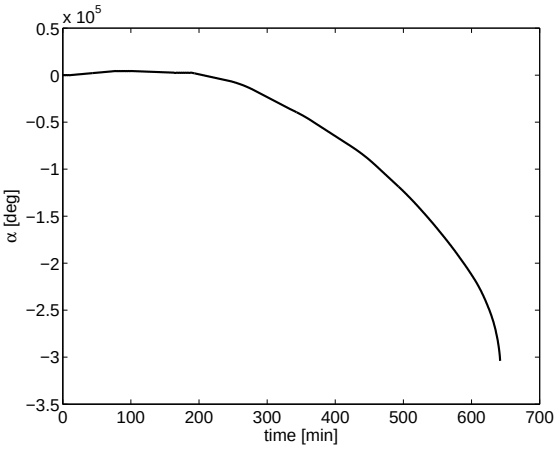
(e) 5 of 8. (f) 6 of 8.

(g) 7 of 8. (h) h_{min} reached: 8 of 8.

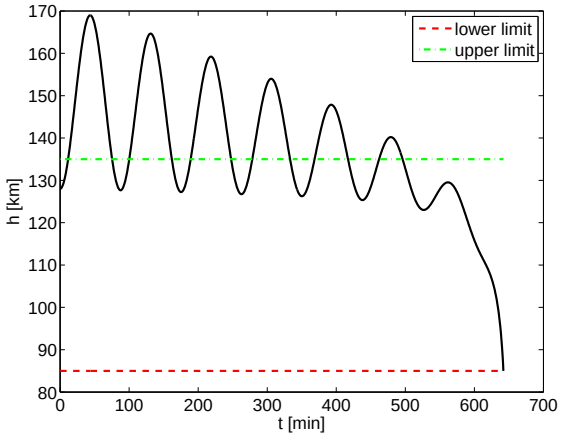
Figure 6.19: Case 2: Visualization of total power distribution over the surface for the airfoil in flight (at the same times as in Figure 6.18).



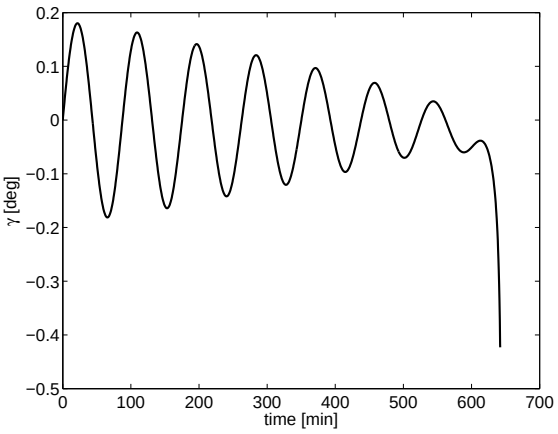
(a) Orbital speed V.



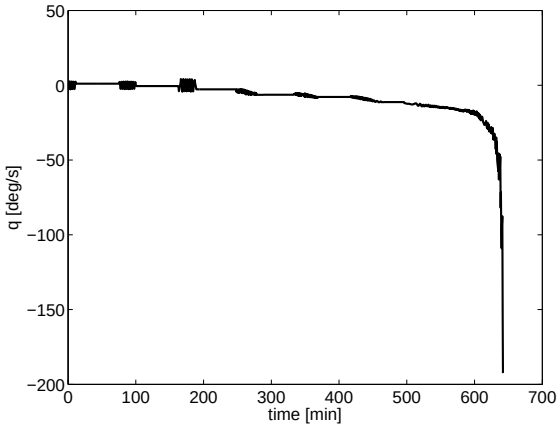
(b) Angle of attack α.



(c) Altitude h.

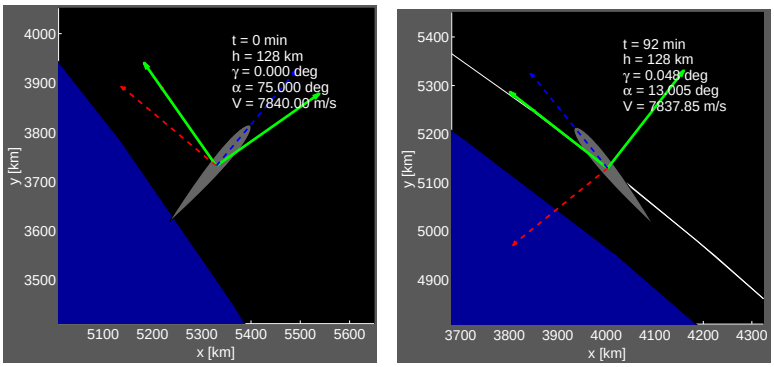


(d) Flight-path angle γ.



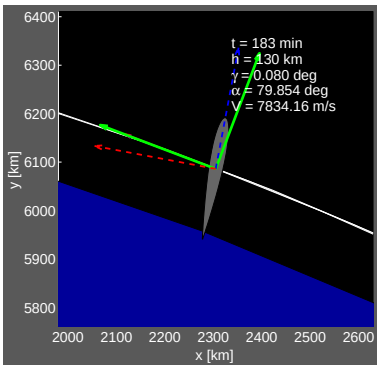
(e) Pitch rate q.

Figure 6.20: Case 3: Time histories of dynamic variables for the airfoil in flight until h_{min} is reached.

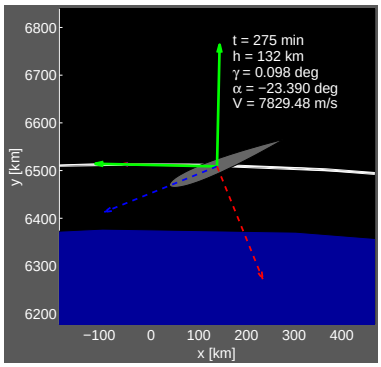


(a) Initial state: 1 of 8.

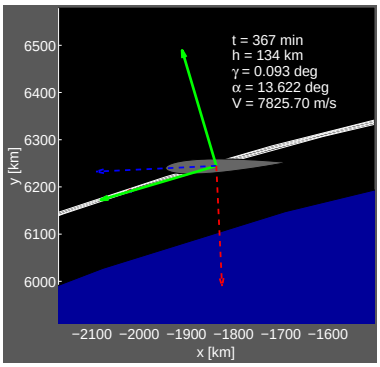
(b) 2 of 8.



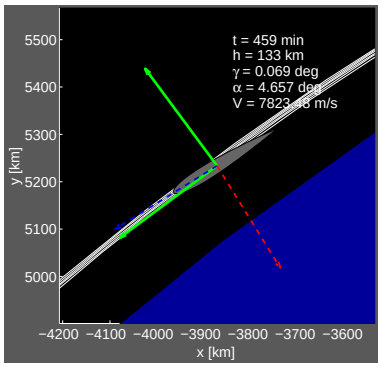
(c) 3 of 8.



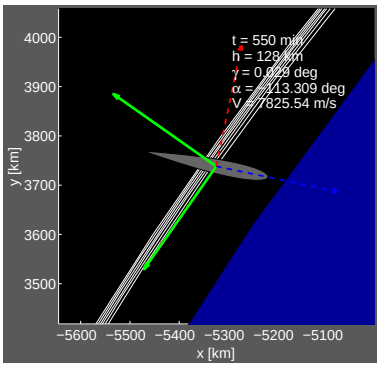
(d) 4 of 8.



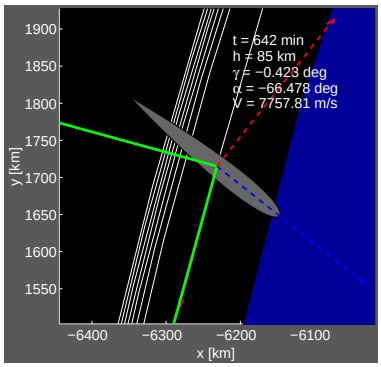
(e) 5 of 8.



(f) 6 of 8.

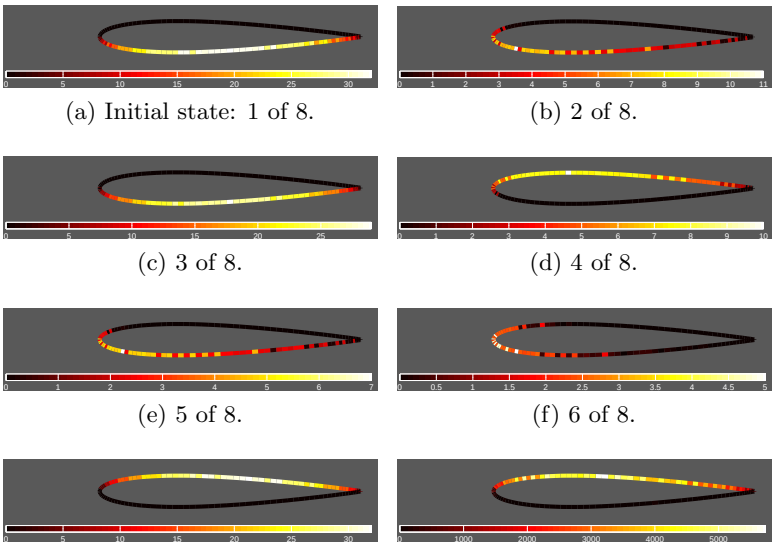


(g) 7 of 8.



(h) h_{min} reached: 8 of 8.

Figure 6.21: Case 3: Geometric visualization for the airfoil in flight.



(a) Initial state: 1 of 8. (b) 2 of 8.

(c) 3 of 8. (d) 4 of 8.

(e) 5 of 8. (f) 6 of 8.

(g) 7 of 8. (h) h_{min} reached: 8 of 8.

Figure 6.22: Case 3: Visualization of total power distribution over the surface for the airfoil in flight (at the same times as in Figure 6.21).

6.5.2 Dynamic Analysis: Airfoil

When the case beginning with $\alpha_0 = 0$ and $q_0 = 0$ is simulated, the airfoil should not rotate at all, due to its symmetry. However, this assumed equilibrium is only quasi-stable, meaning that it is the center of a region within which dynamic trajectories would likely exist as limit sets if the plotted subspace were the entire state space. The presence of h as a state, which tends to pull such possible limit sets out of their apparent location prevents the aforementioned points from being defined as true equilibria. Any slight moment will cause the body to begin rotating and enter an oscillation about $\alpha = 0$. This behavior occurs in the free-molecular version of the airfoil's single degree-of-freedom simulation from Chapter 3. Without extremely precise numerical results, this equilibrium cannot be captured in simulation and must be inferred. Figure 6.23 shows the result of simulating the airfoil starting at this equilibrium for $h_0 = 130$km. The moment source function

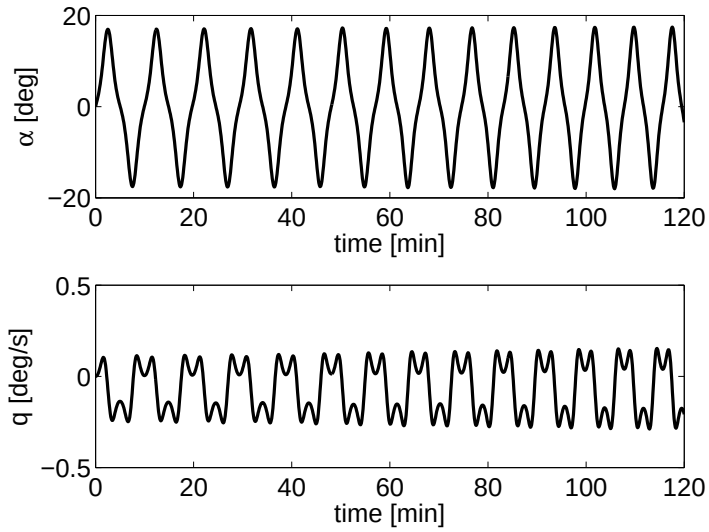
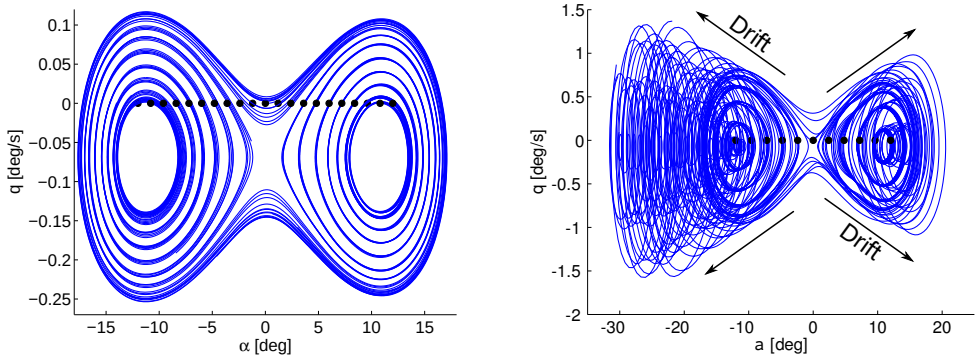


Figure 6.23: Time history of α and q for the airfoil starting at the assumed quasi-stable equilibrium of [$\alpha_0 = 0$deg, $q_0 = 0$deg/s] and $h_0 = 130$km.

M_y is not exactly zero at α_0 for any value of h. Higher resolution DSMC runs near this region would assist in refining M_y but will likely never achieve $M_y(\alpha_0) = 0$. However, in practice, it is impossible to know the attitude of a spacecraft to infinite precision. It is enough to understand the

local behavior of the body in question without capturing its equilibria exactly. A phase diagram consisting of the $[\alpha q]$–subspace such as Figure 6.24a can illustrate the local dynamics of the origin in the airfoil's case. From examination of Figure 6.24a it is clear that the airfoil starting at $\alpha_0 = 0$ will



(a) $h_0 = 130$km. (b) $h_0 = 100$km.

Figure 6.24: Airfoil origin–local phase diagrams.

enter one of two separate temporary limit sets. Neither of these sets remain intact as h decreases, however. Equilibria appear to exist near $|\alpha| \approx 12$deg, again alluding to the presence of α^*, noted in Section 3.3. In a more refined set of data, each of these points would occur at $[\alpha_0 = \alpha^* q_0 = 0]$ rather than at the offset positions shown in Figure 6.24a, which occur for $q_0 \approx -0.06921$deg/s. Figure 6.24b shows the same portrait except with the object starting at a lower altitude. An artificial asymmetry is visible in both the α and q directions. Each temporary limiting region that was clearly defined in the high altitude case seems to drift outward while increasing the oscillation frequency in the low altitude case. This is the slow drift seen in Figure 6.14b which occurs at approximately 150mins. Yet, both of these local phase portraits are expected for a symmetric object, simply offset from the origin. And while it is desirable to increase the accuracy with which these dynamics are simulated, the pertinent characteristics remain visible even with offsets present. It is possible, for example, to begin a simulation close to α^* where the airfoil remains relatively static in α. Figure 6.25 gives an example of the airfoil starting close to this quasi-stable equilibrium at a high altitude. Forces are weak enough to allow α to change very little until the airfoil reaches

lower altitudes, inducing larger oscillations.

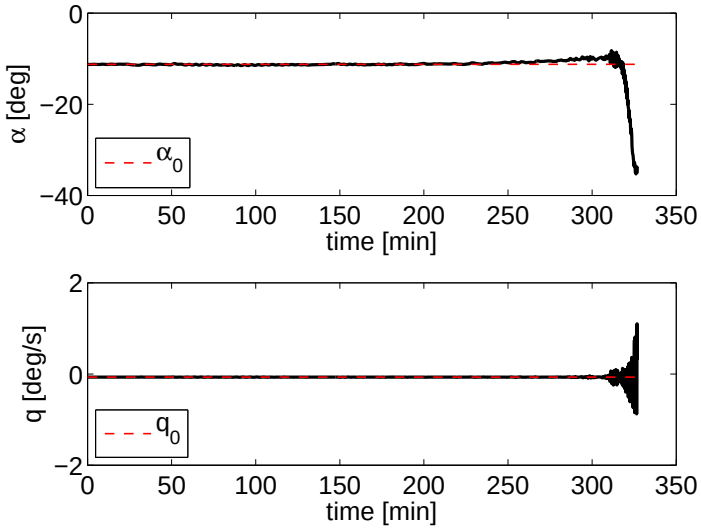


Figure 6.25: Time history of α and q for the airfoil starting at the actual quasi-stable equilibrium of $[\alpha_0 = \alpha^* = -11.25000\text{deg}, q_0 = -0.06921\text{deg/s}]$ and $h_0 = 130\text{km}$.

6.5.3 Arbject

Cases of interest for the Arbject are summarized in Table 6.3. The body is assumed to be 1m deep and composed of a homogeneous material of density $\rho = 2700\,\text{kg/m}^3$. Relevant dynamic variable time histories for Case 1 are given in Figure 6.26. Geometric visualization for eight frames, starting at $t = 0$, is shown in Figure 6.27 and corresponding power distribution diagrams are shown in Figure 6.28. Case 2's results are given in Figures 6.29 to 6.31 and Case 3's in Figures 6.32 to 6.34.

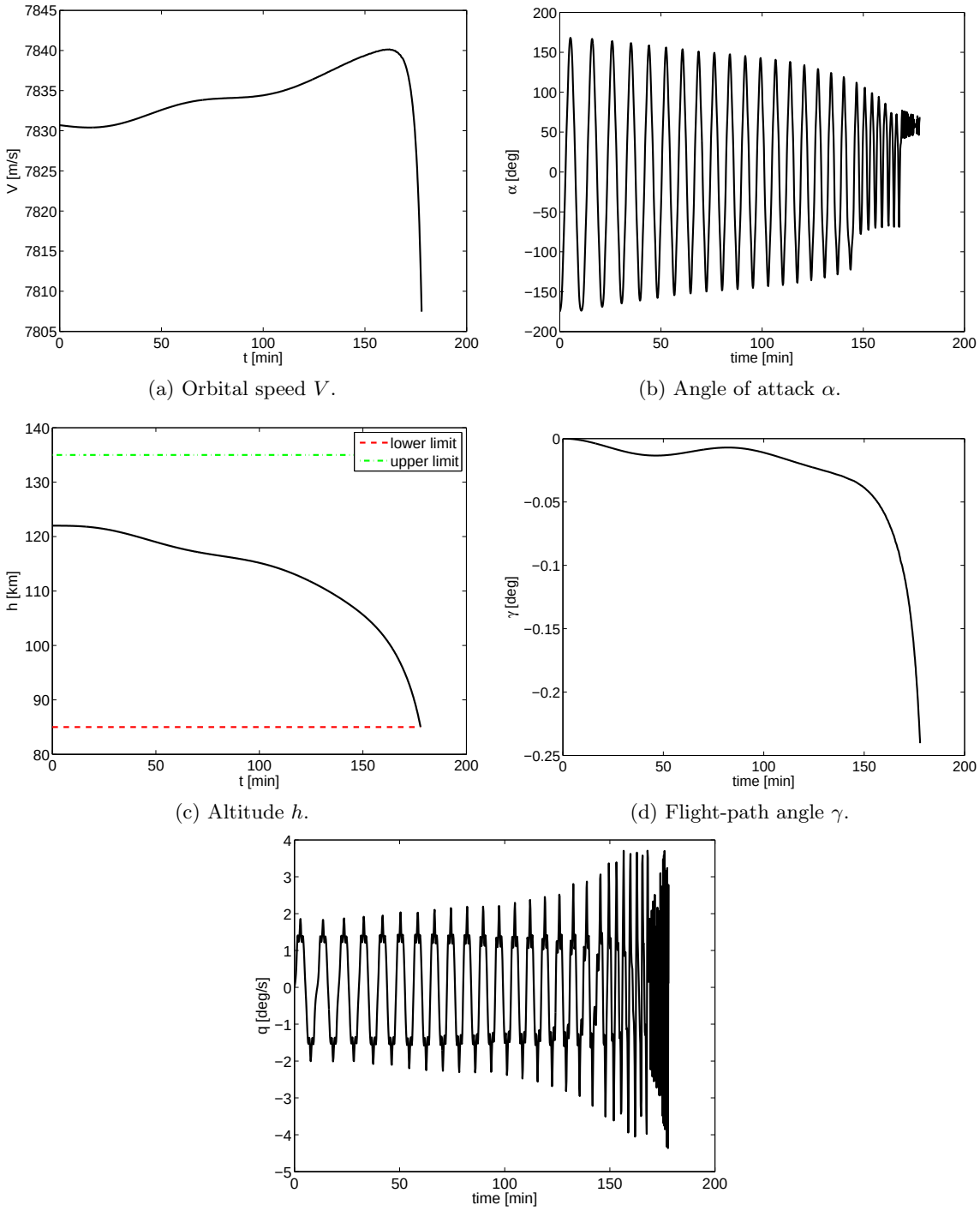
Case 1 starts the Arbject with its concave feature forward, into the flow as shown in Figure 6.27a. Figure 6.26b shows it entering a decaying oscillation of an initial amplitude roughly equal to α_0. This oscillation drops in amplitude, increases in frequency and centers about a positive angle of attack once h falls below the turbopause region. Finally, the Arbject undergoes another dynamic transition to a very high-frequency, low-amplitude oscillation about an even higher angle during the last ten minutes of the simulation.

Table 6.3: Initial conditions for each case of the full orbital simulation of the Arbject (*Circular* speed is the speed calculated for a circular orbit at h_0).

Case	V [m/s]	α_0 [deg]	h_0 [km]	q [deg/s]	θ_0 [deg]	γ [deg]
1	Circular	-175	122	0	35	0
2	Circular	125	122	0	35	0
3	7835	80	125	0	35	0

Case 2 is similar to Case 1 but starts the Arbject at a different angle of attack. Familiar dynamic evolution occurs, with the initial high-altitude stable oscillation transitioning to oscillations about different central angles, with higher frequencies, as the Arbject descends. This case illustrates the expected sensitivity to initial conditions of an asymmetric object.

Case 3 starts the Arbject in a mildly non-circular orbit, ignoring aerodynamic forces above h_{max}, similar to the airfoil's Case 3. The Arbject rotates freely when it is above the viable region, and experiences oscillations when below it. This is evident from the partitioned behavior in Figures 6.32b and 6.32e.



(a) Orbital speed V.

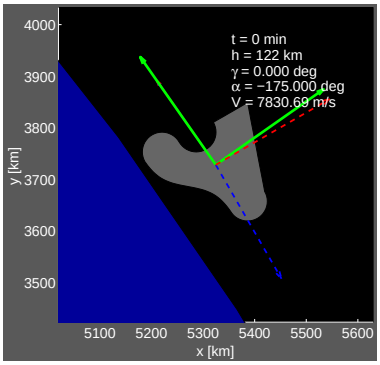
(b) Angle of attack α.

(c) Altitude h.

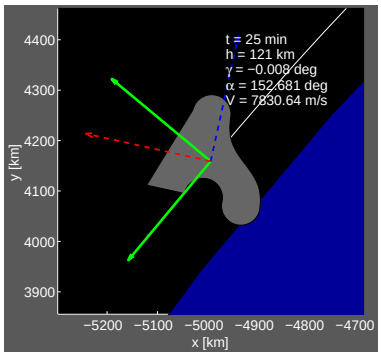
(d) Flight-path angle γ.

(e) Pitch rate q.

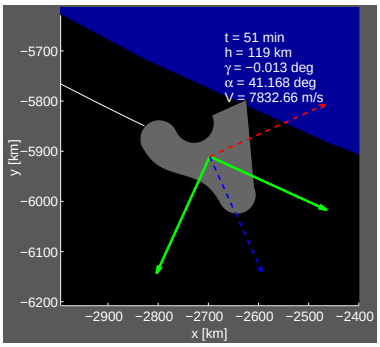
Figure 6.26: Case 1: Time histories of dynamic variables for the Arbject in flight until h_{min} is reached.



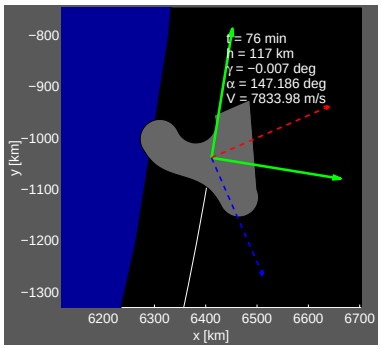
(a) Initial state: 1 of 8.



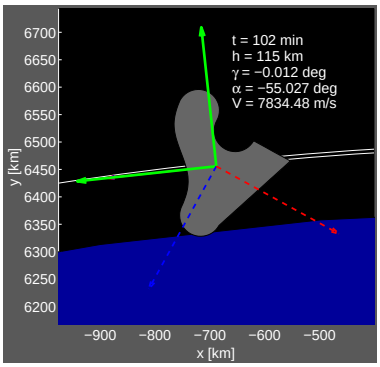
(b) 2 of 8.



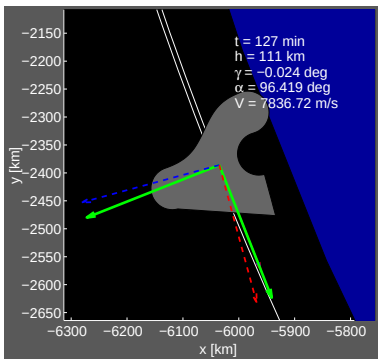
(c) 3 of 8.



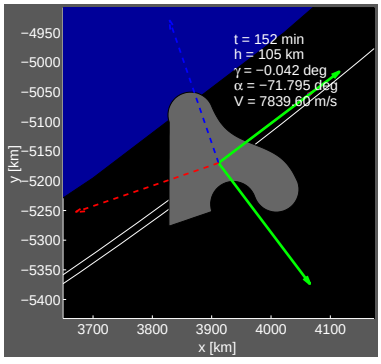
(d) 4 of 8.



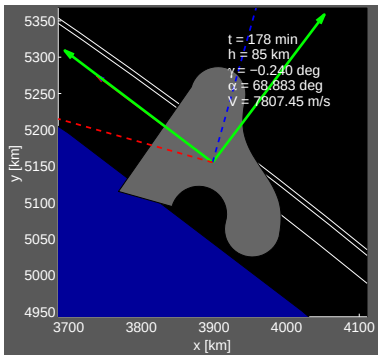
(e) 5 of 8.



(f) 6 of 8.

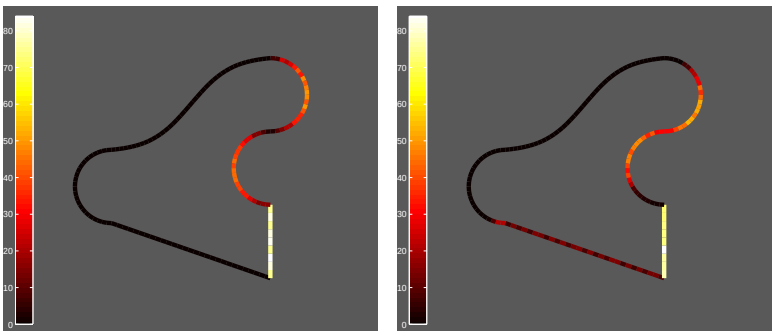


(g) 7 of 8.



(h) h_{min} reached: 8 of 8.

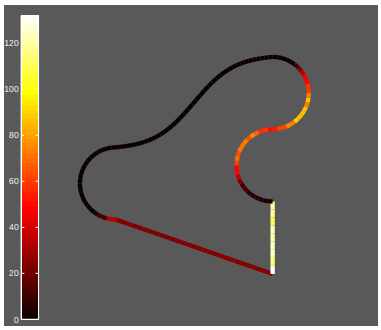
Figure 6.27: Case 1: Geometric visualization for the Arbject in flight.



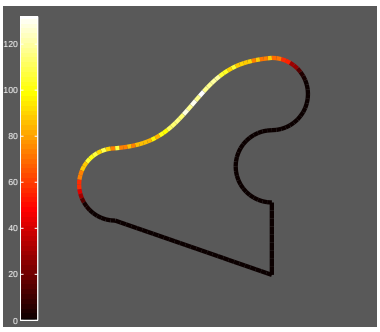
(a) Initial state: 1 of 8.

(b) 2 of 8.

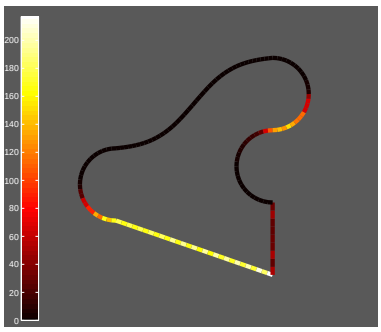
(c) 3 of 8.



(d) 4 of 8.

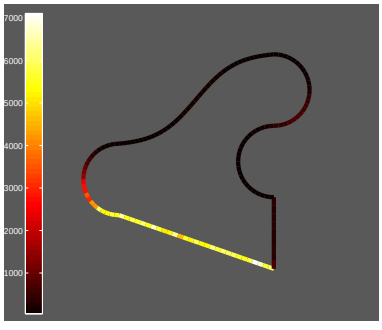


(e) 5 of 8.



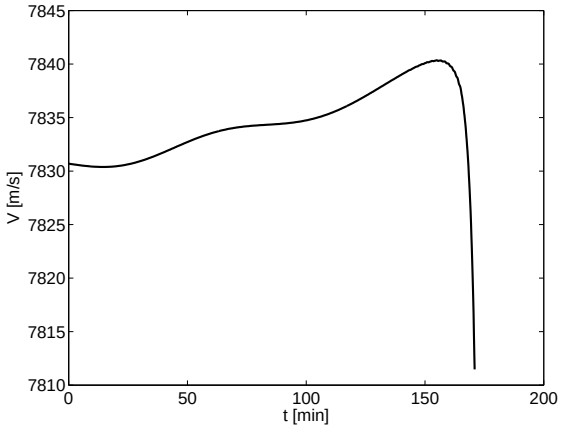
(f) 6 of 8.

(g) 7 of 8.

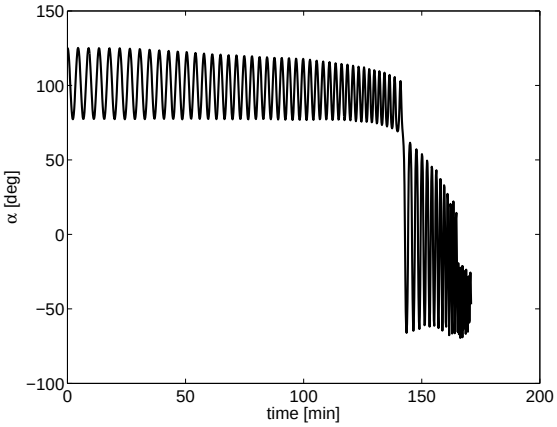


(h) h_{min} reached: 8 of 8.

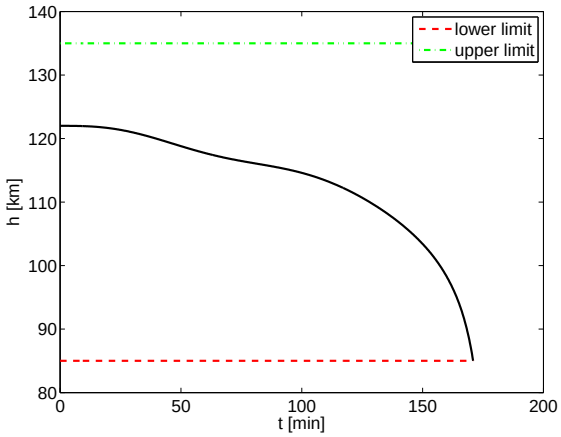
Figure 6.28: Case 1: Visualization of total power distribution over the surface for the Arbject in flight (at the same times as in Figure 6.27).



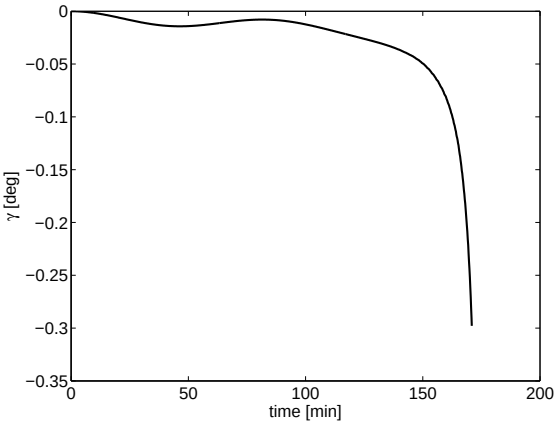
(a) Orbital speed V.



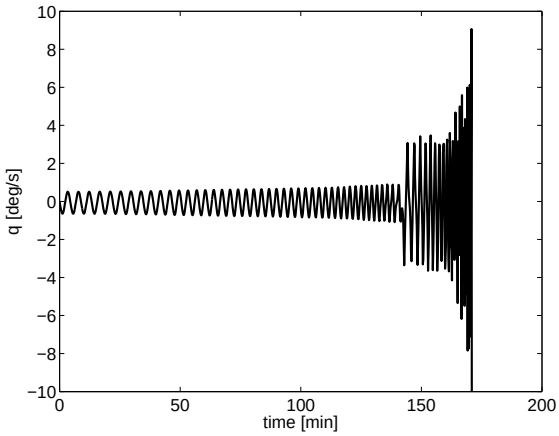
(b) Angle of attack α.



(c) Altitude h.

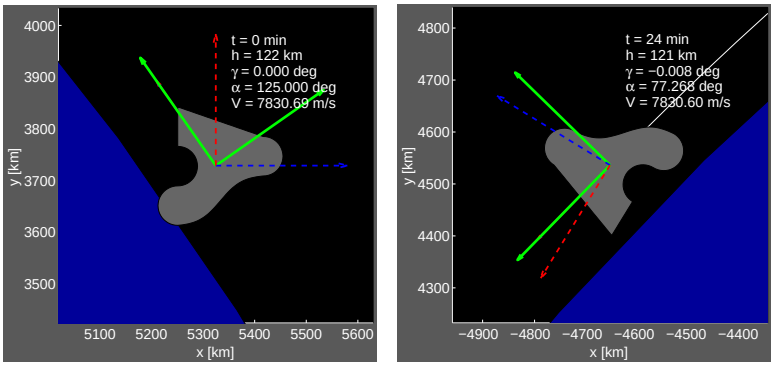


(d) Flight-path angle γ.



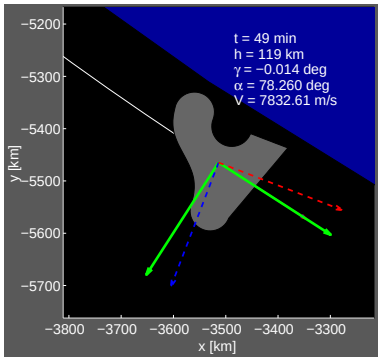
(e) Pitch rate q.

Figure 6.29: Case 2: Time histories of dynamic variables for the Arbject in flight until h_{min} is reached.

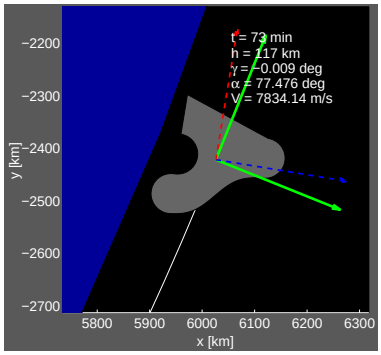


(a) Initial state: 1 of 8.

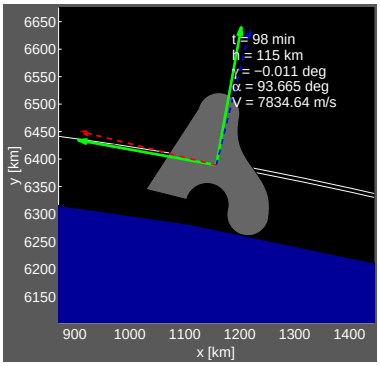
(b) 2 of 8.



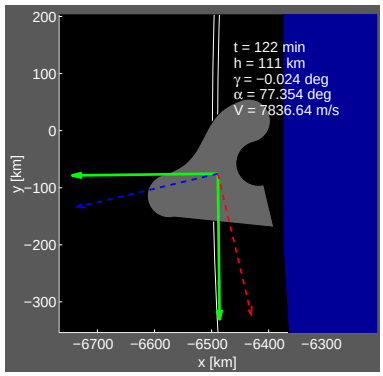
(c) 3 of 8.



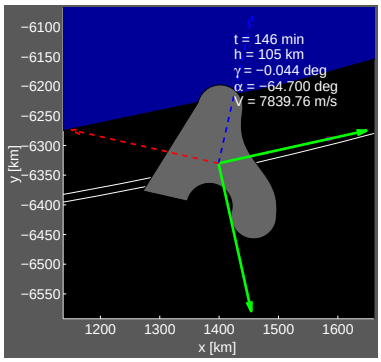
(d) 4 of 8.



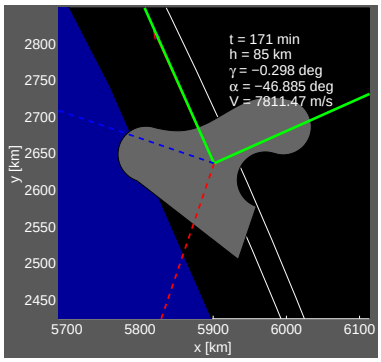
(e) 5 of 8.



(f) 6 of 8.

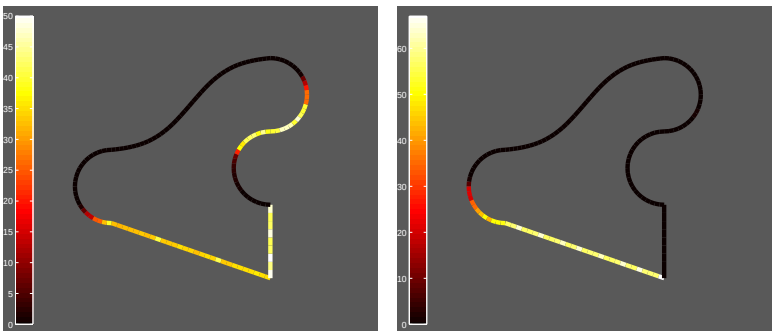


(g) 7 of 8.



(h) h_{min} reached: 8 of 8.

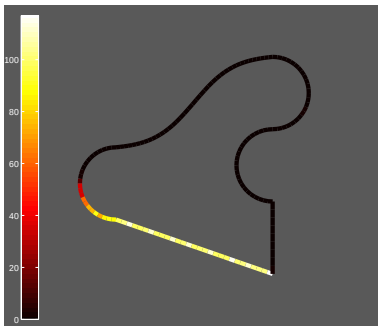
Figure 6.30: Case 2: Geometric visualization for the Arbject in flight.



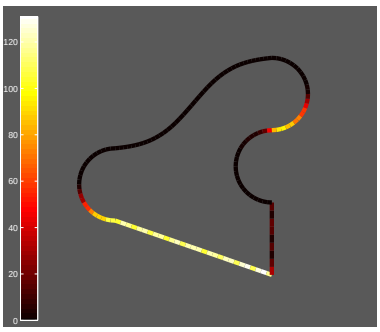
(a) Initial state: 1 of 8.

(b) 2 of 8.

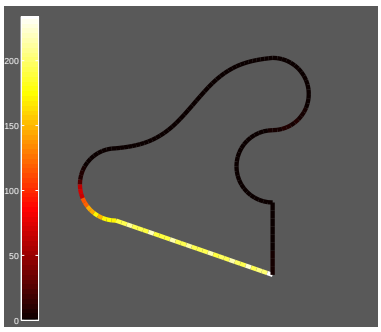
(c) 3 of 8.



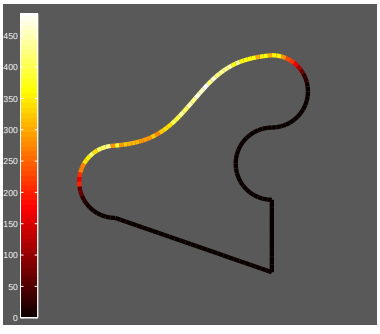
(d) 4 of 8.



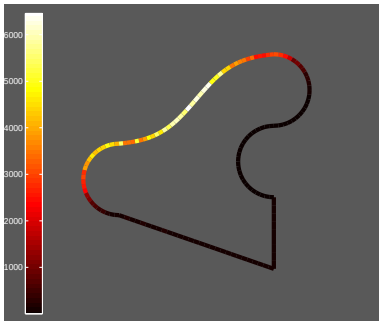
(e) 5 of 8.



(f) 6 of 8.

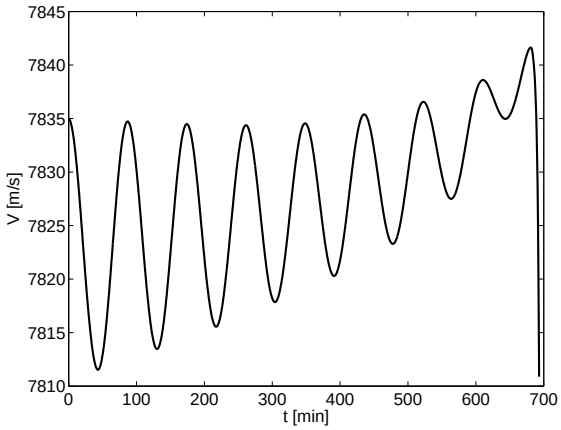


(g) 7 of 8.

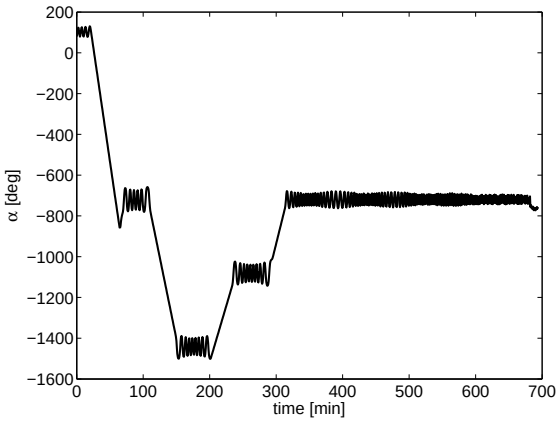


(h) h_{min} reached: 8 of 8.

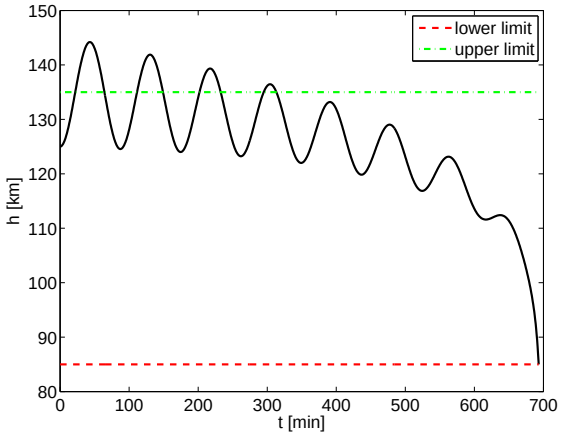
Figure 6.31: Case 2: Visualization of total power distribution over the surface for the Arbject in flight (at the same times as in Figure 6.30).



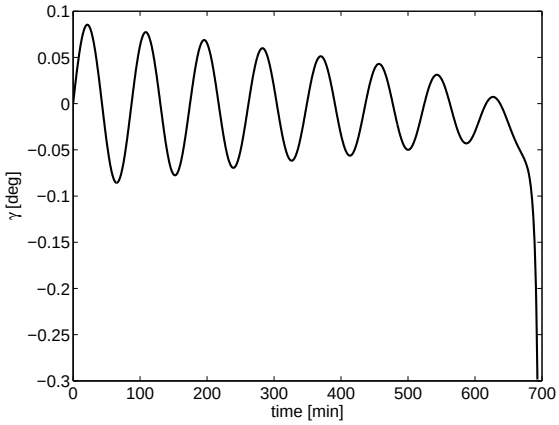
(a) Orbital speed V.



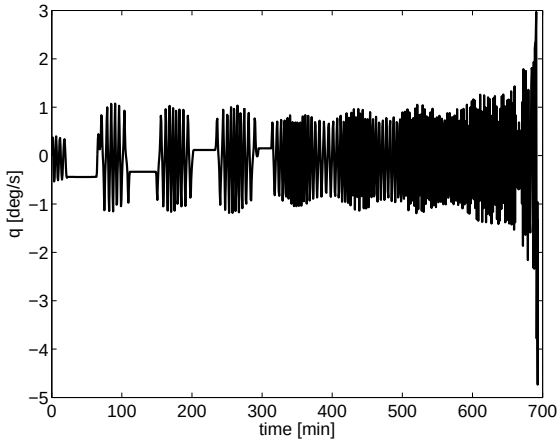
(b) Angle of attack α.



(c) Altitude h.

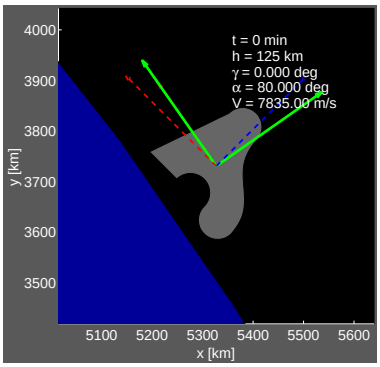


(d) Flight-path angle γ.

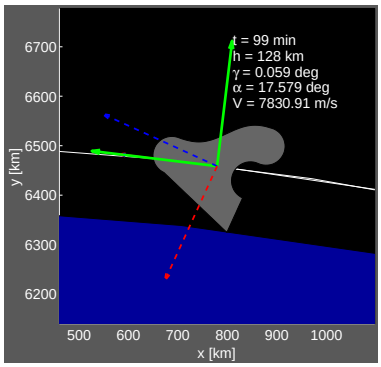


(e) Pitch rate q.

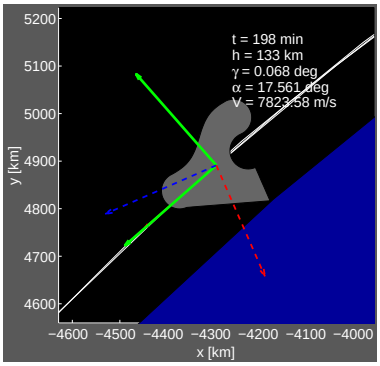
Figure 6.32: Case 3: Time histories of dynamic variables for the Arbject in flight until h_{min} is reached. Aerodynamic forces are set to zero above h_{max}.



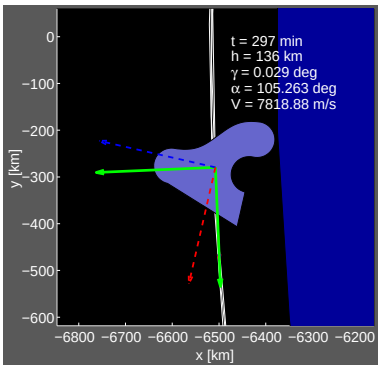
(a) Initial state: 1 of 8.



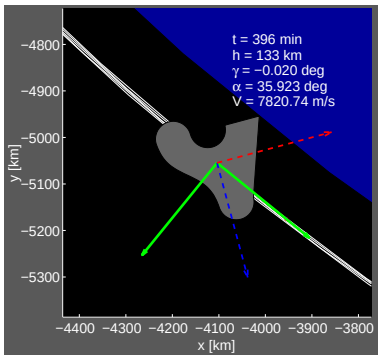
(b) 2 of 8.



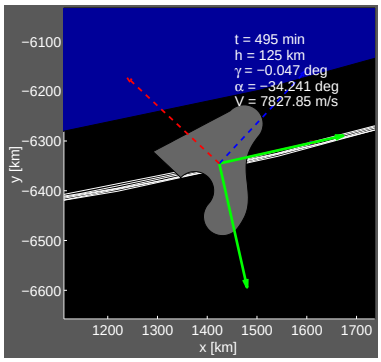
(c) 3 of 8.



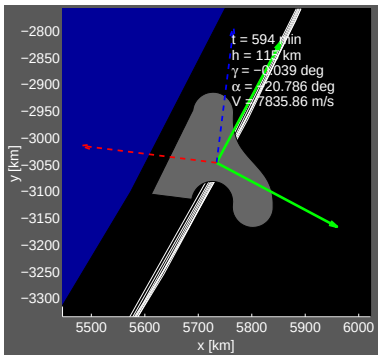
(d) 4 of 8 (aero forces off).



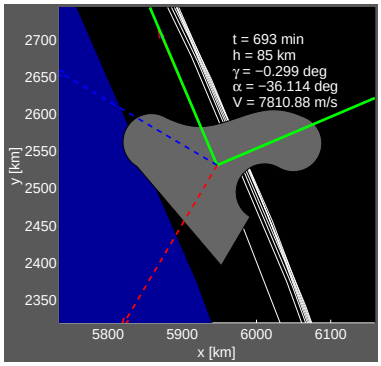
(e) 5 of 8.



(f) 6 of 8.

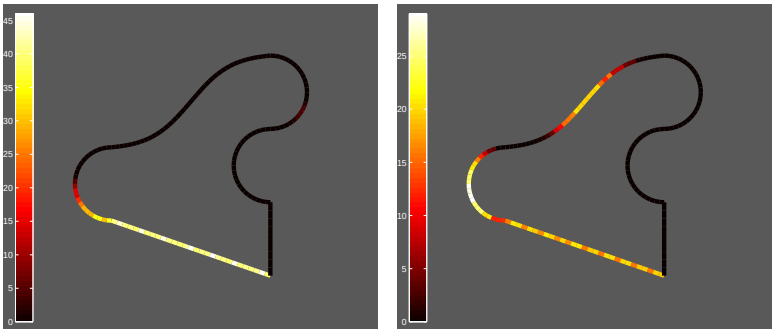


(g) 7 of 8.



(h) h_{min} reached: 8 of 8.

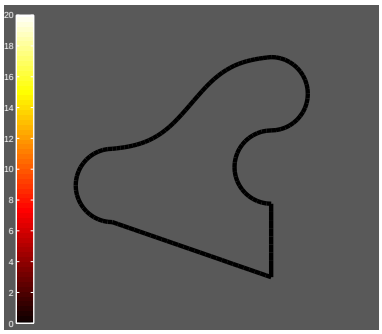
Figure 6.33: Case 3: Geometric visualization for the Arbject in flight.



(a) Initial state: 1 of 8.

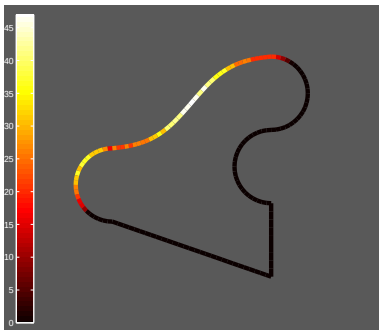
(b) 2 of 8.

(c) 3 of 8.

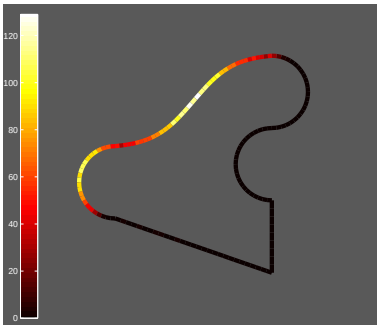


(d) 4 of 8.

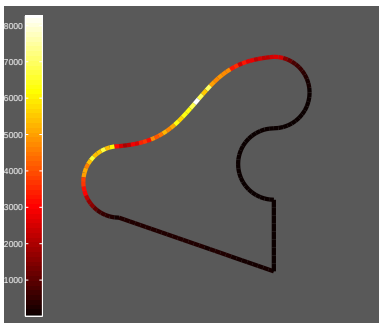
(e) 5 of 8.



(f) 6 of 8.



(g) 7 of 8.

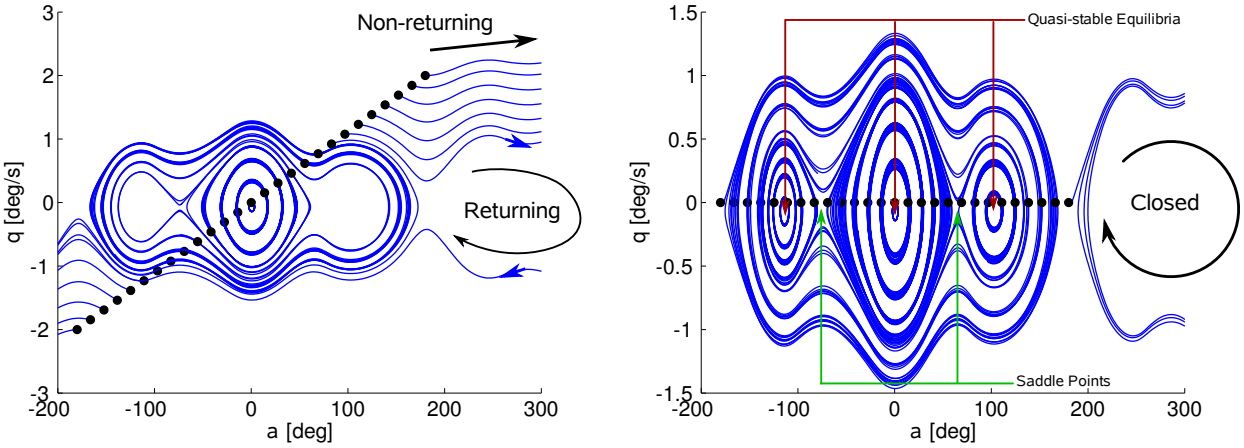


(h) h_{min} reached: 8 of 8.

Figure 6.34: Case 3: Visualization of total power distribution over the surface for the Arbject in flight (at the same times as in Figure 6.33).

6.5.4 Dynamic Analysis: Arbject

Subspace phase portraits may be made for the Arbject in order to more closely examine its dynamic properties. Figure 6.35 presents the $[\alpha q]$–subspace portraits, starting with high-altitude initial conditions ($h_0 = 130$km). Figure 6.35a shows some trajectories which start with high enough



(a) Version 1: larger range in q_0.

(b) Version 2: $q_0 = 0$deg/s.

Figure 6.35: Arbject phase diagrams for high-altitude initial conditions ($h_0 = 130$km). Two versions, each with a different set of initial conditions, are given for visual clarity.

initial positive rotation ($q_0 > 0$) inducing unbounded rotations ("non-returning"). Other lower, positive values of q_0 are slow enough to induce a few complete rotations and eventually reverse direction, returning to the central subspace of $[-\pi, \pi]$ ("Returning"). All values of $q_0 < 0$ appear to be non-returning, however. Figure 6.35b shows a set of three central regions, each separated by saddle points at $[\alpha \approx -73.80\text{deg}, q_0 \approx -0.07\text{deg/s}]$ and $[\alpha \approx 65.17\text{deg}, q_0 \approx -0.07\text{deg/s}]$ and centered about quasi-stable equilibria at $[\alpha \approx -113.90\text{deg}, q_0 \approx -0.07\text{deg/s}]$, $[\alpha \approx 0.62\text{deg}, q_0 \approx -0.07\text{deg/s}]$ and $[\alpha \approx 103.15\text{deg}, q_0 \approx -0.07\text{deg/s}]$. The initial condition of $\alpha_0 = 180$deg escapes the central regions and enters a state-space orbit about a neighboring identical set of regions (which must necessarily exist due to the periodic nature of α).

At lower altitudes, the portrait becomes more complicated. State-space trajectories drift in various directions depending on the initial conditions. Figure 6.36 gives a set of initial conditions

which show this behavior. Most initial conditions tend to result in a decaying amplitude oscillation

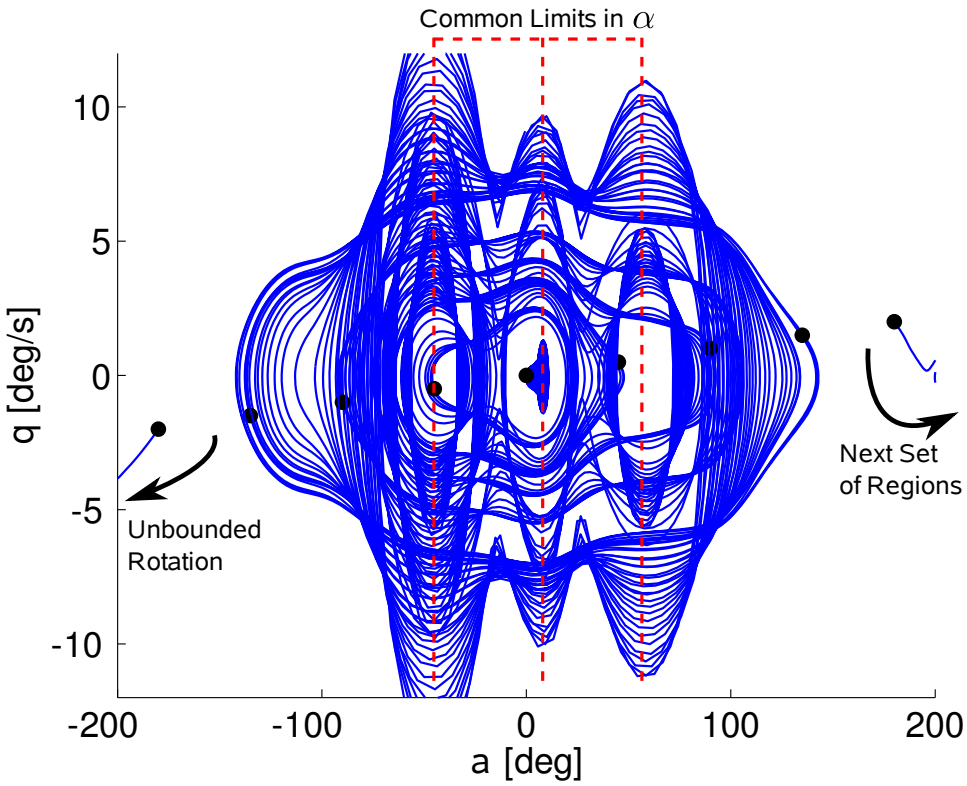


Figure 6.36: Arbject phase diagram for low-altitude initial conditions ($h_0 = 100$km).

about a certain α by the time the object has descended to its minimum altitude. The highly asymmetric geometry of the Arbject makes identifying all possible final orientations difficult, however it appears from the trajectories shown in Figure 6.36 that there are three common limits in α. These limits are the angles about which most of the final oscillations of the Arbject are centered when the simulation begins with low enough $|q_0|$. They are located at $\alpha \approx -45$deg, $\alpha \approx 10$deg and $\alpha \approx 60$deg. Excessive initial rotation in the system causes unrecoverable, unbounded rotation or tumbling, but for low $|q_0|$, it can be expected that the Arbject eventually reaches an angle of attack close to one of these limits. For example, Figure 6.26b shows the Arbject approaching $\alpha \approx 60$deg and Figure 6.29b shows it approaching $\alpha \approx -45$deg.

6.6 Comparisons to Traditional Ballistic Methods

A high-resolution parameter scan over V, α and h allows high-fidelity numerical simulations to be performed which incorporate the effect of body dynamic motion. Traditionally, many spacecraft have been treated as a ballistic body where only a drag coefficient is calculated and assumed sufficient. One of the tenets of this work is the demonstration that these traditional drag problems do not provide enough information to accurately model motion in orbit of spacecraft and debris. A comparison between some traditional "drag-only" ballistic model and the high-fidelity, coupled, body-orbital dynamics simulations would be appropriate. However, there exist no drag models specific to either of the geometries used.

An equivalent averaged-drag or ballistic model may be created using the force data that was computed by assuming the body is only subject to atmospheric drag. This assumption is made by treating the body as a point mass having no body dynamic motion, which removes the dependency on the angle of attack α. This model can be constructed by defining a mean drag force which integrates out dependence on the angle α, which the computed data already possess. If this integration is performed, the drag remains a function of the other two states V and h as $\overline{D}(V,h)$. This form may be retained and used to perform a simplified numerical integration where it is linearly interpolated much the same as the full body-orbital simulation was. However, it is useful to examine the quality of the function's dependence on V, as it is expected to be rather weak. Figure 6.37 shows this functionality for the Arbject. Clearly there is very little change across the range of speeds calculated. This insensitivity to speed compared to that of density (i.e. altitude in atmospheric flight problems) is typical of flows under these conditions. The small envelope of change for drag force defined by the lower and upper curves in Figure 6.37 make a reasonable case for assuming a constant speed. In light of this fact, it seems also reasonable to further refine $\overline{D}(V,h)$ by integrating out dependence on V. However, this time the averaging will only introduce very minor inaccuracies and is being performed to reduce the dimensionality of the drag function for convenience. The very strong sensitivity to altitude should remain, thus giving the final mean

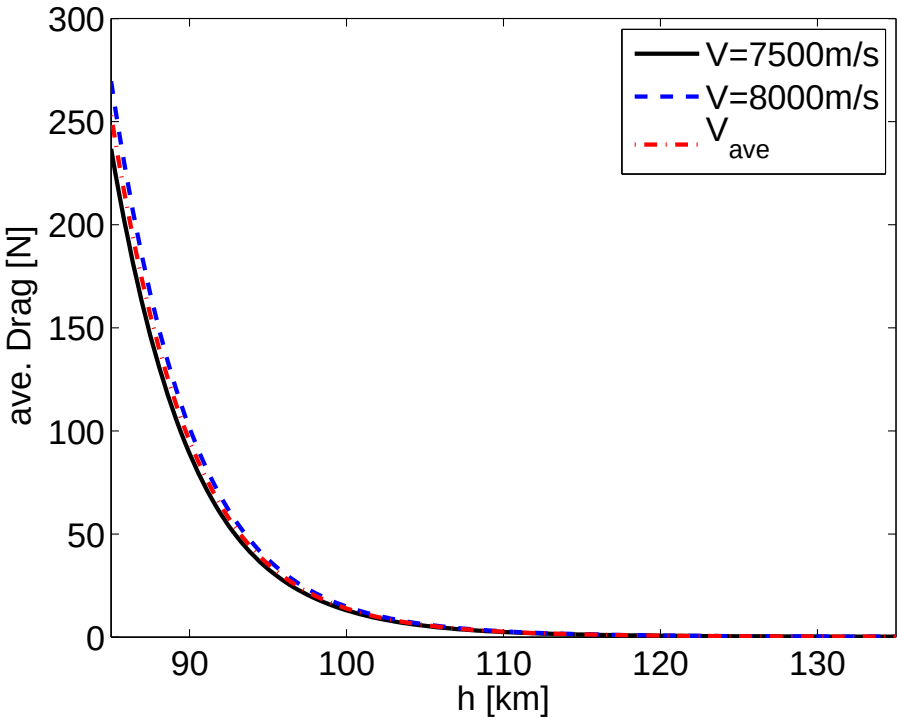


Figure 6.37: Drag force for the Arbject taken as an average over α, calculated from the hi-resolution source data.

drag definition of

$$\overline{D}(h) = \frac{1}{V_{max} - V_{min}} \int_{V_{min}}^{V_{max}} \overline{D}(V,h)dV = \frac{1}{2\pi(V_{max} - V_{min})} \int_{V_{min}}^{V_{max}} \int_{-\pi}^{\pi} D(V,\alpha,h)d\alpha dV \tag{6.34}$$

The limits of integration depend on the range of available computed source data. However, the range of α should cover 2π in order to properly consider the possibility of full rotation. The central curve labeled as V_{ave} in Figure 6.37 shows $\overline{D}(h)$ for the Arbject's data.

As a point of interest (particularly when attempting to compare with classical satellite drag theory and data), a drag coefficient may be defined in a similar manner to the drag force. Unfortunately, using the standard definition of aerodynamic force of

$$\overline{D}(h) \equiv \frac{1}{2} m_g n(h) V(h)^2 A \overline{C}_D(h) \tag{6.35}$$

where A is a characteristic area and m_g is the mass of a gas molecule is somewhat ambiguous. This

is because of the necessity of assuming either a constant speed, or, as is required here, a speed as a function of altitude h. Seeing as how V and h are independent spatial directions in the state $\boldsymbol{s}$, and thus in the computed data set, $V(h)$ cannot be defined. Instead, the form of Equation 6.35 may be used to guide a definition of an average drag coefficient $\overline{C}_D(h)$ by incorporating the desired dependence on V into the integration over V as

$$\overline{C}_D(h) \equiv \frac{1}{\pi m_g n(h) A(V_{max} - V_{min})} \int_{V_{min}}^{V_{max}} \frac{1}{V^2} \left[\int_{-\pi}^{\pi} \overline{D}(V, \alpha, h) d\alpha \right] dV \tag{6.36}$$

This version of $\overline{C}_D(h)$, plotted in Figure 6.38 [3] , produces a function which is of different form than drag force, which may appear awkward in the context of a traditional satellite drag problem. The necessity of $n(h)$ appearing in the definition as a non-constant variable compounds

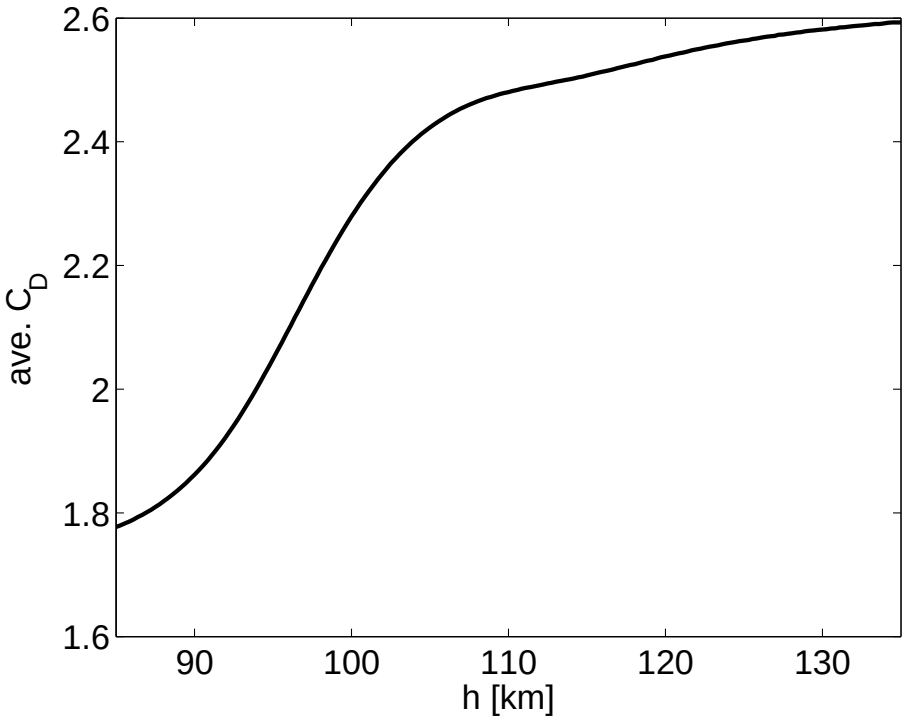


Figure 6.38: Specially-defined mean drag coefficient (Equation 6.36) for the Arbject calculated from the hi-resolution source data.

[3] The value of A used is 0.75m which is a characteristic length of the Arbject used to produce the computed source data; i.e. variable L in Appendix C. Density $n(h)$ is obtained by linearly interpolating from the MSISE[56] model.

this awkwardness, giving lower coefficients where higher drag force is encountered, at low altitudes. Nevertheless, it yields values within the range of typical ballistic drag models. It is important to note that although Equation 6.35 is given as a definition, it is not the definition that is used here. Equation 6.36 is written as a definition purposefully, with the implication that it is defined by taking inspiration from the classic definition of Equation 6.35 and not formally derived from it.

6.6.1 Simplified dynamics without body motion

To remove the effect of body dynamics on the system is to return to a point mass or ballistic description of orbital dynamics. This assumption is equivalent to setting α to zero in Equations 6.27 to 6.32, which aligns the body axes with the velocity vector. In a ballistic model, forces acting on the center of mass due to aerodynamics are typically decomposed into components along the directions described as the drag-lift (DL) frame. This frame is shown in Figure 6.39. Drag force

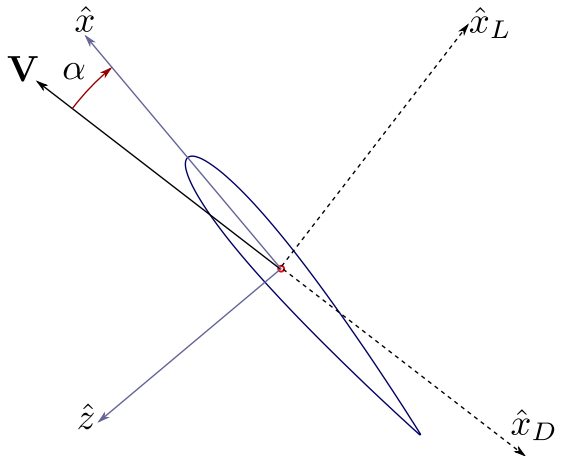


Figure 6.39: Orientation of the drag-lift frame compared to the body frame.

is positive along the $\hat{\boldsymbol{x}}_D$ direction, which is always equal to $-\hat{\boldsymbol{V}}$ and lift force is positive along the $\hat{\boldsymbol{x}}_L$ direction, which is normal to the velocity and forms a right-handed system such that the cross-direction is positive out of the perifocal plane in the same direction as the orbital angular velocity vector. Since $\alpha = 0$, aerodynamic force in the DL frame $\boldsymbol{F}_a^{(DL)} = [D\ L]^{\mathrm{T}}$ is related to the

body force by the transformation

$$\boldsymbol{S}_{DL}^{b} = -\begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} = -\boldsymbol{I} \tag{6.37}$$

which means that body force, as it appears in Equations 6.30 and 6.32, is written in terms of lift and drag as $\boldsymbol{F}_a^{(b)} = [-D \ \ -L]^{\mathrm{T}}$. Setting $\alpha = 0$ in Equations 6.29 to 6.32, they now become the only equations necessary. The averaged drag $\overline{D}(h)$ (and a similarly-defined $\overline{L}(h)$) may be used for D and L to give a set of simplified, average, ballistic equations:

$$\dot{s}_1 = \dot{h} = V_h \tag{6.38}$$

$$\dot{s}_2 = \dot{V}_h = r\Omega^2 - \frac{\mu}{r^2} - \frac{V_h \overline{D}(h)}{mV} + \frac{r\Omega \overline{L}(h)}{mV} \tag{6.39}$$

$$\dot{s}_3 = \dot{\theta} = \Omega \tag{6.40}$$

$$\dot{s}_4 = \dot{\Omega} = -\frac{2V_h\Omega}{r} - \frac{\Omega \overline{D}(h)}{mV} - \frac{V_h \overline{L}(h)}{mVr} \tag{6.41}$$

This system may be used to compare with the full, high-fidelity system by performing integrations of each under the same initial conditions. One relevant metric is the decay of an orbit, which may be examined by plotting the state $h(t)$. Figure 6.40 shows a comparison of this type for the Arbject, which uses a circular starting orbit at an initial altitude of $h_0 = 125$km. Figure 6.40 shows the same type of comparison for the airfoil. A range of initial angles of attack are shown, each giving a different decay and taking a different amount of time. The asymmetry of the Arbject means there is no obvious functionality of decay time (the time to decay from h_0 to $h_{min} = 85$km) on α_0. Table 6.4 lists relative error of total decay time t_d of the average DL ballistic model with respect to each simulation of the proper body dynamics model.

As the orbit decays into the lowest altitudes, the two models diverge more, causing up to almost eight percent of error in decay time in the set of α_0 chosen for the Arbject. For the airfoil, the maximum relative error occurs at $\alpha = 0$deg, with sixty percent. The striking difference between the decay behavior of each geometry using these comparisons is due to the fact that the Arbject is rather blunt regardless of its angle of attack, while the airfoil has a larger disparity in its drag

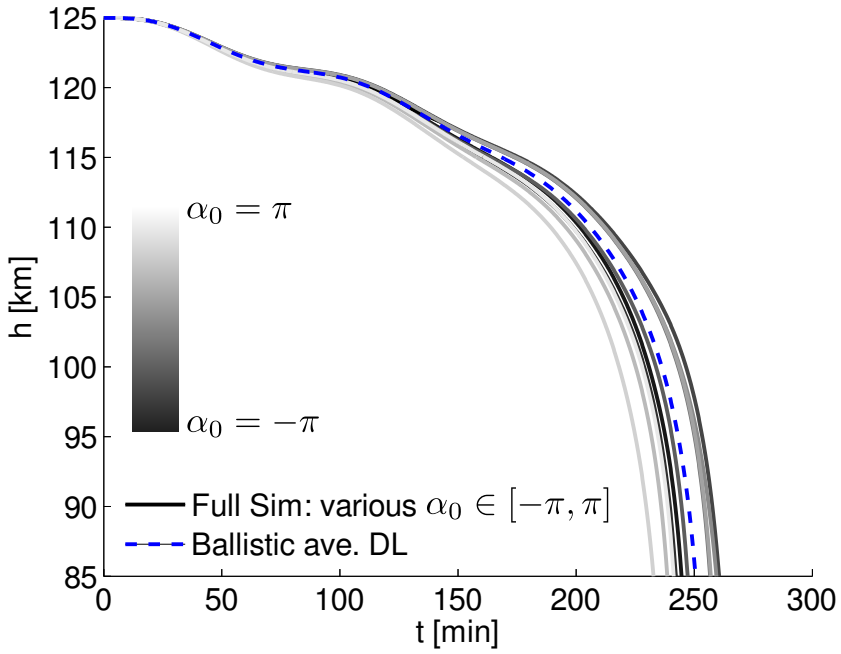


Figure 6.40: Comparison of orbit altitude decay of the Arbject for the full, high-fidelity simulation for a range of initial angles of attack with the equivalent average drag-lift ballistic representation of the same geometry.

between $\alpha = 0$deg where it has a very small stream-normal profile and $\alpha = 90$deg where it is fully blunted to the stream. Therefore the airfoil's drag has much a higher standard deviation from its α-averaged value, $\overline{D}(h)$. The total drag force on the airfoil is also higher than the Arbject when it is at a fully-blunted orientation (e.g. $|\alpha| = 90$deg) due to significantly more pressure drag at lower altitudes. Furthermore, because of the greater magnitude of forces and moments on the airfoil at higher $|\alpha|$, the motion it experiences over the course of its descent differs greatly from what it experiences when it starts in a stable region of state space. For high initial angles of attack, the airfoil is much more likely to enter into an unstable tumbling motion, inducing more drag than if it had remained in a static or low-amplitude stable oscillation.

Additionally, it is important to note is that the averaged drag and lift are still calculated from the full set of data which makes their values more accurate in terms of how well these terms represent the effect of body rotation on average. Without a high-resolution data set which provides source functions of such quality, $\overline{D}(h)$ and $\overline{L}(h)$ may be far less accurate unless they are empirically

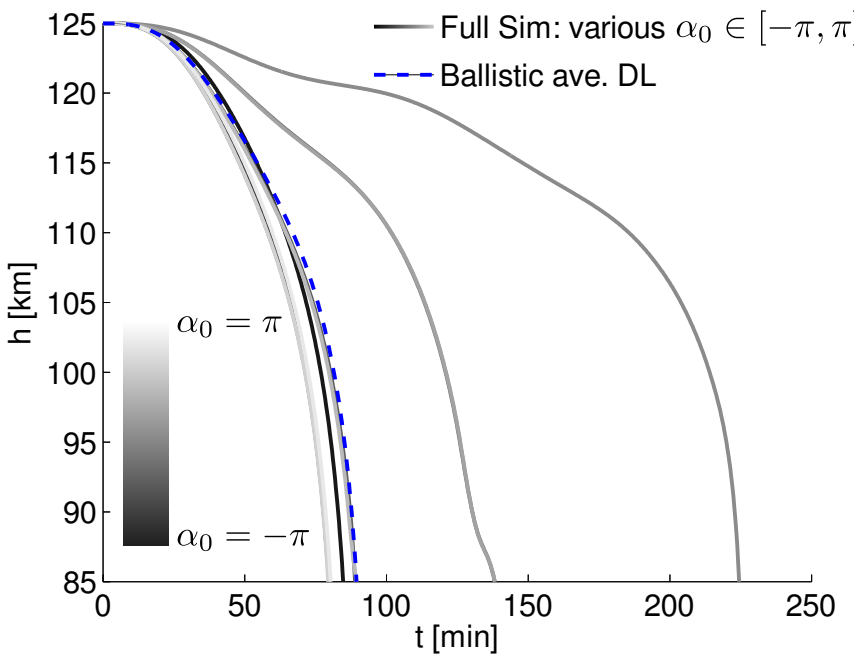


Figure 6.41: Comparison of orbit altitude decay of the NACA-0012 airfoil for the full, high-fidelity simulation for a range of initial angles of attack with the equivalent average drag-lift ballistic representation of the same geometry.

determined (e.g. from actual, recorded flight data). Again, the purpose of this comparison is to demonstrate some of the possible differences between simpler predictive methods (which often only give drag) and what can be achieved with inclusion of body dynamics. An example of these differences is clear when comparing the two distinct geometries examined in this chapter.

Table 6.4: Relative errors in total decay time for the example problems shown in Figures 6.40 and 6.41.

α_0 [deg]	Airfoil ϵ_{t_d} [%]	Arbject ϵ_{t_d} [%]
-180	5.58	2.4
-144	12.06	3.4
-108	12.1	-3.8
-72	0.62	1.3
-36	-35.26	2.41
0	-60.14	-3.4
36	-35.30	-2.5
72	0.02	5.0
108	12.64	7.6
144	11.38	3.8

6.7 Summary of Results

Aerodynamic forces were incorporated into a perifocal description of orbital dynamics to yield a complete description of aerodynamically-induced body dynamics in orbit. The resulting set of equations of motion couple body and orbital dynamics via the angle of attack, which is the primary degree-of-freedom associated with body dynamics in this description. The aerodynamic inputs to these equations are two force and one moment source function. Each source function was shown to be capable of being expressed as a function of the three dynamic variables: V, α and h. This formulation reduced the subsequent parameter scan computation to a tractable level. To generate each of the source functions, a DSMC computation was performed for every combination of V, α and h, forming a parameter scan operation over a 3D subspace of the total six-dimensional state space. Resolutions of this parameter space were chosen, along with DSMC resolutions and configuration parameters such that the computation of these data were possible on the available super cluster computing system. Having generated the source data, the equations of motion were integrated in time to perform simulations of two objects: the NACA-0012 airfoil and an Arbject. At each step in the integration, source functions were obtained by linearly interpolating the raw source data. The resulting system allowed for high-fidelity time-accurate simulations of the objects in orbit under arbitrary initial conditions.

To best illustrate the effectiveness of including body dynamics resulting from rarefied aerodynamic inputs in the transition regime, the source data parameter space was chosen to span a region of orbital flight within the Earth's lower thermosphere, from 85km to 135km in altitude. In this region, both objects were clearly demonstrated to experience a transition from rarefied (near-free-molecular) flow to near-continuum flow over the course of a number of example simulations which simulated an atmospheric entry. The effect of this transition on rigid bodies, which has not been well-studied, was illustrated via the presentation of the evolution of source functions (e.g. Figures 6.8 and 6.13) through the transition region, and by dynamic analysis of phase portraits of relevant subspaces.

Both geometries were observed to share some similar qualitative properties. Each experiences decaying oscillations in α which remain centered about a certain α which is usually close to α_0 for higher altitudes. As the objects descend, the increase in magnitude of aerodynamic forcing due to density increase causes shifts in state space trajectories. Case 1 for the airfoil and Cases 1 and 2 for the Arbject finish their simulations by approaching what appears to be a final limit in α. It may be that as an object descends below the thermosphere, oscillatory motion is eventually damped out, leaving only a final orientation at which the object remains until it slows considerably enough for gas forces in the stream-wise direction to be comparable to other directions. The specific evolution of motion is different for each set of initial conditions, however, which further emphasizes the need to examine transition flow dynamics.

Finally, a comparison was made between the high-fidelity simulations which incorporated body dynamics and an averaged drag-lift or ballistic interpretation of the source data where body dynamics were effectively removed. The airfoil was shown to be poorly modeled by an average ballistic model, showing as much as 60% relative error in orbital decay time with respect to the high-fidelity body-dynamic model, while the Arbject fared better with a maximum relative error of approximately 8%. This analysis also provided some reason to ignore the dependence on V of source data. Source functions were seen to be highly insensitive to V, compared to their sensitivity to altitude (i.e. to density and temperature). This reasoning allows the reduction of the parameter

space of each function with minimal loss in accuracy. For extensions of this work to 3D, this is a very useful conclusion.

Final Summary and Conclusions

Modern spacecraft systems operating in low Earth orbit can no longer ingore or marginalize the effects of their motion within rarefied atmosphere if they are to meet the requirements of industry, governments and militaries. Ballistic or drag-only models of rarefied aerodynamics are insufficient in predicting orbital decay, performing accurate atmospheric science and for designing control systems and hardware to meet the rigorous requirements of these spacecraft. The appearance of modern tools and methods, namely numerical tools such as the direct simulation Monte Carlo method, have created an opportunity to raise the level of fidelity with which spacecraft aerodynamics are treated to a level similar to that seen for atmospheric aircraft.

A framework of approaches, considerations and methods was constructed by identifying the challenges in numerical techniques and dynamic modeling current present in spacecraft systems and contrasting them with aircraft systems. The continuum-rarefied transition flow regime was identified as being the most crucial fluid regime for this class of problems. Most spacecraft which operate in low Earth orbit, and certainly those which enter an atmosphere are subject to the effects of this regime. Lack of availability of appropriate predictive computational methods had prevented transition flow dynamics from being examined in detail. Furthermore, overly-simplistic ballistic descriptions of complicated systems consisting of asymmetric geometry, unknown flight conditions and surface interaction physics have continued to be used. And although DSMC and other numerical methods have matured, spacecraft dynamics remains exclusive of body dynamic motion and its coupling with rarefied gas dynamic forcing.

There exist details of motion of orbiting and entering objects which cannot be known without

inclusion of body dynamics resulting from accurate modeling of aerodynamics. Such characteristics were shown to exist even in free-molecular flow by applying available closed-form methods to generate forces and moments of some example geometries in realistic orbital conditions. The NACA-0012 airfoil was used to demonstrate this by intergrating single degree-of-freedom equations of motion subject to in-loop-evaluated aerodynamic force and moment functions. Characteristics such as regions of stability and locations of equilibria were identified numerically. Subsequently, simulations of this simple system showed the airfoil–generally considered an uninteresting geometry–could oscillate at high enough frequencies to possibly be of concern to designers and analysts. This simulation method was then extended to the transition regime, at Knudsen numbers of 1 where DSMC was used to obtain forces and moments *a priori* which were then used both directly in integration and approximated using a piecewise lienar method. The purpose of the latter approach highlighted the possiblity of reducing the highly non-linear effects of gas flow to a set of tractable, analytical representations which could be used to gain further insight into the system's character and contribute to the design of control systems.

The problem of flow-motion coupling was addressed in the context of rarefied gas flow. This problem was determined to be of possible importance in dynamic simulation and analysis since moving objects naturally affect their surrounding flow and vice versa. The state variables which appear in state-space descriptions of body dynamic motion include rates such as the pitch rate q or, as in the single degree-of-freeom model of Chapters 3 and 4, angle of attack rate $\dot{\alpha}$. In order to generate forces and moments from aerodynamic numerical simulations, functionality on these rates is, in general, required. Unforunately. the feedback of this system, often seen in fluid-structure interaction problems in continuum flow, was shown to be difficult to describe. Simple models of rotational motion are desirable to use in numerical simulations in order to avoid excessive computational cost. The transpiration boundary condition method was chosen to be the method of choice, should DSMC simulations be required to simulate motion without actually moving the geometry. The initial hypothesis was posed that rigid-bodies would be decoupled from the hypersonic flows common to orbital flight, allowing a method such as transpiration to be acceptable. It was de-

termined, however, that in order to properly examine this phenomenon, a true moving-boundary DSMC code would be required. However, some concession was made to use the tools available. Through numerical experiments, it was shown that rates of rotation of two types of geometry could be ignored when the rates were small with respect to the thermal speeds of the near-field gas. Whether transpiration is an acceptable model at high rates was not determined. However, an analysis lead to the conclusion that actual motion and transpirationally-mimicked motion were no different from the prespective of energy and momentum balance of the surface with the gas. Furthermore, a computational concern was identified which prevents accurate DSMC simulation of flows with rotation rates which give certain transpirational velocities. Overall, flow-motion coupling was determined to remain a difficult problem to characterize. Nevertheless, the numerical studies concluded that objects in hypersonic transition flow could ignore coupling of dynamic rotational rates with the flow. This reduced the necessary dimension of the parameter space which defined the final set of simulations of orbital motion, greatly reducing the computational cost of generating the data over this space.

Demonstrations were given of a NACA-0012 airfoil and an arbitrarily asymmetric object ("Arbject") in thermospheric 2D perifocal orbital and entering flight which utilized high-fidelity aerodynamics obtained from a vast number of DSMC code runs. These DSMC runs were made on the CU JANUS super cluster system. Roughly $9e5$ individual code runs were performed per geometry in order to provide forces in two directions, one moment and power transfer. Focusing specifically on the transition region for meter-sized objects, this parameter space covered the altitude range of 85km to 135km, a range of 7500m/s to 8000m/s in orbital speed and all possible angles of attack ($[-pi, pi]$). This gave a range of Kn between 0.01 and 20. Source function data were then linearly interpolated in an integration of 2D orbital-aerodynamic equations of motion to generate high-fidelity, time-accurate simulations of motion of the objects in flight and descent through the lower thermosphere. The effects of transition flow were clearly observed as each object's source functions could be seen to resemble their equivalents obtained from their free-molecular panel method equivalents (first shown in Chapter 3). Functions which contained high frequencies

at the lowest altitudes (below roughly 100km) were smoothed to include only a few major modes at high (approaching free-molecular) altitudes. This "blunting" characteristic was observed to carry over into the dynamic motion of the objects. Both objects experienced rotational motion that was high oscillatory about a few key angles of attack at high altitudes but were subject to drift to other limiting angles as the objects descended.

The details of the dynamic responses of these objects permitted teh identification of likely modes of operation. The airfoil possessed three quasi-equilibria at high altitudes. The airfoil would experience quasi-stable oscillations about these locations (which were locations of angles of attack at effectively zero pitch rate) at high altitudes. As it descended, these three locations would drift away from their initial locations in state space, causing the airfoil to approach a small set of asymtotic angles of attack at the conclusion of the simulations at 85km in atltitude. The Arbject's possessed much more complicated dynamic characteristics, as expected. However, a similar analysis allowed the identification of quasi-equilibria in this system as well. When the arbject was started near one of these quasi-equilibria it maintined its orientation with only minor oscillations for most of its descent, only experiencing large amplifications at very low atltitudes. Furthermore, despite the asymmetry of the object and its sensitivity to small changes in initial conditions, it was observed to also posssess a set of specific, asymptotic limits in its orientation at $\alpha \approx -45$deg, $\alpha \approx 10$deg and $\alpha \approx 60$deg.

These types of dynamic characteristics can be identified for any object. Those used in the examples are merely tools to demonstrate how useful such an analysis may be. Blunting behavior and identification of equilibria are particularly useful. One possible application is the selection of certain attitudes and rates in order to avoid (or induce) certain motion without the imposition of external control actuators. Spacecraft geometry may be designed with dynamic characteristics in mind, similar to who aircraft may be designed with consideration for avoidance or excitation of certain linear eigenmodes. In fact, linear eigenmodes may be also identified for spacecraft in rarefied flow should certain trim flight conditions be prescribed. The wealth of additional information deriving from high-fidelity aerodynamic models has wide-reaching application in control system

and hardware design.

Finally, full-orbital simulations were compared with an average drag-lift ballistic description obtained by taking the mean drag and lift over all angles of attack and all speeds for each altitude. Orbit decay times for objects using the ballistic description differed by as much as eight percent for the Arbject and sixty percent for the airfoil. Ignoring the contribution to ballistic drag from body dynamics created much more error for the airfoil because of the significant difference between flight at low and high angles of attack. When the airfoil is at a low angle, it suffers minimal pressure drag. At high angles, it experiences more pressure drag than the Arbject at an equivalent angle. Although the Arbject is asymmetric and contains a significant concavity, it has less variability in ballistic drag over all angles of attack than a "flat" object such as the airfoil. Even the eight percent maximum relative error in decay time for the Arbject may be significant, however. This error corresponds to roughly twenty minutes of time, which can result in a large uncertainty in the final ground impact location. Stansberry and Johnson confirm this when discussing USSTRATCOM Joint Space Operations Center's re-entry prediction capabilities:

> *"TIP (Tracking and Impact Prediction) messages provide the best estimates of reentry time and location but have large uncertainties. Even at T-2 hours, the uncertainty of reentry time is on average +/- 25 minutes for nearly circular orbits. This equates to +/- 12,000 km on the Earth."* Ref. [68]

Considering that aerodyanic forcing is overwhelmingly dominant during re-entry, it is clear that the introduction of body dynamics via high-fidelity aerodynamic modeling contributes to serious improvements in re-entry prediction.

6.8 Extensions to 3D

Essential for practical application in real spacecraft systems is the treatment of 3D body dynamics in 3D orbits. However, the demonstrations and analyses presented in this work are independent of problem dimension. Had the resources been available, 3D simulations using 3D source function data would've been equally straightforward to implement. The restriction to 2D in this work was made primarily because of the current state of Voldipar's geometry models. Use of

other codes could be substituted should they posses the same level of computational performance, batch-run capability and configuration options. Additionally, source functions must depend on at least two additional angles in 3D; which increases the dimension of the parameter space used for source data generation to intractable levels given the computing resources available.

Each individual DSMC computation for the airfoil and Arbject in 2D required between 3 and 30 minutes (see Figure 6.3) on a single processor. In 3D, one to two orders of magnitude more computation time would be required to perform similarly accurate runs without parallelization. A 3D DSMC code almost certainly requires a parallel implementation in order to make each run complete in reasonable time. Unfortunately, the dependence of 3D source functions on the side-slip angle and roll angle introduce likely impractical or even impossible computational requirements. More methods of parameter space reductions must be investigated in order to reduce this burden. Though at the expense of accuracy, some means of obtaining such methods may lie in the analysis of sensitivity of source funtions to their various parameters. In Section 6.6 it was shown that drag and lift force were high insensitive to orbital speed when compared to altitude (i.e. density). This property may be exploited by ignoring dependence on speed completely with likely only a small penalty in accuracy. Similar analyses might motivate using coarser resolutions in certain parameter space directions. It may also be possible to use data obtained from lower dimension scans along certain directions to correct source functions rather than directly compute them at all locations in the state space. In this formulation, source data might be obtained within a smaller parameter space where some independent variables are kept constant. Additional low-dimensional scans would generate corrective data which would then be applied to adjust the final value of each source function by perturbing it from its nominal value. Such methods relate to the piecewise Taylor approximations shown in Chapter 4 but applied in a different context.

6.9 Other future work

Orbital simulations were done under the assumption of angularly symmetric atmosphere in order to minimize source function dimension. This is perhaps the most egregious affront to

accuracy in the results of Chapter 6. Density varies so significantly within Earth's atmosphere that it is unlikely that treatment of aerodynamic forces can assume no dependence on location or ignore contributions to density changes from a number of sources such as solar activity. This fact also implies that the parameterization of density and temperature by altitude would not be possible. Source functions would have to include dependence on number density n, temperature T and orbital location (one angle, θ in 2D, and two angles in 3D) as separate variables.

Another assumption of convenience is that the objects' surfaces remained adiabatic at all times, with a fixed temperature and GSI model. In reality, some coupling between power transferred to the surface and surface temperature and GSI model adjustment would be required. As an object's surface heats, its GSI model or at least its parameters must be adjusted appropriately. Even a simple diffuse GSI model with full thermal accommodation would require updating of surface temperature. This, in turn, would increase the mean reflected speed which then affects the momentum balance of the object with the gas. For more complicated GSI models such as CL, adjustments in accommodation coefficients also might be required.

Most future work would then focus on increasing the number of parameters upon which the coupled aerodynamic–orbital system depends. Each of the aforementioned improvements would introduce unacceptable expansions to the dimension of source functions. Necessary are more peripheral techniques which can make approximations to source functions by utilizing smaller sets of data. For spacecraft which have trim attitudes, linearization has enormous potential to be capable of incorporating atmospheric variations, surface heating and GSI management without requiring unreasonable computing resources.

Bibliography

[1] G. Beylkin and M. J. Mohlenkamp. Algorithms for numerical analysis in high dimensions. SIAM J. of Sci. Comput., 26(6):2133–2159, 2005.

[2] G.A. Bird. Monte-Carlo Simulation in an Engineering Context. Progress in Astronautics and Aeronautics, AIAA, 74:239–255, 1981.

[3] G.A. Bird. Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Oxford University Press, Oxford, Great Britain, 1994.

[4] G.A. Bird. Recent advances and current challenges for DSMC. Computers Math. Applic., 35(1):1–14, 1998.

[5] G.A. Bird. Sophisticated DSMC. In DSMC07 workshop, Santa Fe, New Mexico, USA, 2007.

[6] B. Bowman. Drag coefficient variability at 175-500 km from the orbit decay analyses of spheres. In AIAA/AAS Astrodynamics Specialist Conference, Paper AAS 05-257, Lake Tahoe, California, USA, 2005.

[7] G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. The Annals of Mathematical Statistics, 29(2):610–611, 1958.

[8] E. Cecil and J.C. McDaniel. Planar Velocity and Temperature Measurements in Rarefied Hypersonic Flow Using Iodine LIF. In AIAA Thermophysics Conference, Toronto, Ontario, Canada, June 2005.

[9] C. Cercignani. Scattering kernels for gas-surface interactions. Transport Theory and Statistical Physics, 2(1):27–53, 1972.

[10] C. Cercignani and M. Lampis. Kinetic models for gas-surface interaction. Transport Theory and Statistical Physics, 1(2):101–114, 1971.

[11] C. Cercignani and M. Lampis. New scattering kernel for gas-surface interaction. AIAA Journal, 35(6):1000–1011, 1997.

[12] S. Chapman and T. G. Cowling. The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction, and diffusion in gases. Cambridge University Press, 1990.

[13] S. Chen and G.D. Doolen. Lattice boltzmann method for fluid flows. Annual Review of Fluid Mechanics, 30:329–364, 1998.

[14] B.K. Ching, D.R. Hickman, and J.M Straus. Effects of atmospheric winds and aerodynamic lift on inclination of orbit of s3-1 satellite. Journal of Geophysical Research-Space Physics, 82(10):1474–1480, 1977.

[15] F.G. Collins and E.C. Knox. Parameters of nocilla gas/surface interaction model from measured accommodation coefficients. AIAA Journal, 32(4):765–773, 1994.

[16] G.E. Cook. Satellite Drag Coefficients. Planetary and Space Science, 13(10):929–946, 1965.

[17] G.E. Cook. Variations in exospheric density during 1967-1968, as revealed by echo 2. Planetary and Space Science, 18(10):387–394, 1970.

[18] R.A. DeCarlo, S.H. Zak, and G.P. Matthews. Variable Structure Control of Nonlinear Multivariable Systems: A Tutorial. In Proceedings of the IEEE, March 1988.

[19] 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.

[20] German Aerospace Center (DLR). DLR - ROSAT. `http://www.dlr.de/dlr/en/desktopdefault.aspx/tabid-10424/`, Oct 2012.

[21] E. Doornbos, M. Forster, T. van Helleputte, J. van den IJssel, G. Koppenwallner, H. Luhr, D. Rees, and P. Visser. Air density models derived from multi-satellite drag observations. Technical Report 21022/07/NL/HE, European Space Agency Study Contract Report, 2009.

[22] C. Ericson. Real-Time Collision Detection. Morgan Kaufmann, 2005.

[23] B. Etkin. Dynamics of Atmospheric Flight. John Wiley & Sons Inc., 1972.

[24] C.C. Fisher and A.S. Arena Jr. On the transpiration method for efficient aeroelastic analysis using an euler solver. In AIAA Atmospheric Flight Mechanics Conference, AIAA Paper 96-3436, San Diego, California, Jul. 1996.

[25] A.L. Garcia and W. Wagner. Generation of the Maxwellian inflow distribution. Journal of Computational Physics, 217(2):693–708, 2006.

[26] F. O. Goodman and H. Wachman. Formula for thermal accommodation coefficients. Journal of Chemical Physics, 46(6):2376–&, 2006.

[27] F.O. Goodman. A three-dimensional hard spheres theory of scattering of gas atoms from a solid surface. Technical Report CR-933, NASA, 1967.

[28] J. Hasselgren, T. Akenine-Möller, and L. Ohlsson. Conservative Rasterization. GPU Gems II, 42:677–690, 2005.

[29] T. He and A. Kaufman. Collision Detection for Volumetric Objects. In IEEE Visualization, New York, USA, 1997.

[30] J. Huang, R. Yagel, V. Filippov, and Y. Kurzion. An accurate method for voxelizing polygon meshes. In 1998 IEEE Symposium on Volume Visualization, 1998.

[31] F.C. Hurlbut and F.S. Sherman. Application of the nocilla wall reflection model to freemolecule kinetic theory. Physics of Fluids, 11(3):486–497, 1968.

[32] D.R. Imbro, M.M. Moe, and K. Moe. On fundamental problems in the deduction of atmospheric densities from satellite drag. Journal of Geophysical Research, 180(22):3077–3086, 1975.

[33] L.G. Jacchia. Static diffusion models of the upper atmosphere with empirical temperature profiles. Technical Report 170, Smithsonian Astrophysical Observatory, 1965.

[34] L.G. Jacchia. Empirical models of the thermosphere and requirements for improvements. Advances in Space Research, 1:81–86, 1981.

[35] A. Kaufman and A. Shimony. 3D Scan-Conversion Algorithms for Voxel-Based Graphics. In ACM Workshop on Interactive 3D Graphics, Chapel Hill, North Carolina, USA, October 1986.

[36] T.S. Kelso. Latest phobos-grunt reentry prediction. `http://celestrak.com/events/reentry/phobos-grunt.asp`, January 2012.

[37] D.G King-Hele. Improved formulae for determinging upper-atmosphere density from the change in a satellite's orbital period. Planetary and Space Science, 11:261–268, 1963.

[38] D.G King-Hele. The upper atmosphere as sensed by satellite orbits. Planetary and Space Science, 40(2/3):223–233, 1965.

[39] D.G King-Hele. The variation of upper-atmosphere density between sunspot maximum (1957-1958) and minimum (1964). Journal of Atmospheric and Terrestrial Physics, 27:197–209, 1965.

[40] K. Koura. Improved null-collision technique in the direct simulation Monte Carlo method: Application to vibrational relaxation of nitrogen. Computers & Mathematics with Applications, 35(1–2):139–154, 1998.

[41] J. Lätt. Hydrodynamic Limit of Lattice Boltzmann Equations. PhD thesis, University of Geneva, 2007.

[42] 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(6–7):595–609, 2001.

[43] C.R. Lilley and M.N. Macrossan. Methods for implementing the stream boundary condition in DSMC computations. International Journal for Numerical Methods in Fluids, 42(12):1363–1371, 2003.

[44] R. G. Lord. Application of the cercignani-lampis scattering kernel direct simulation monte carlo calculations. In 17th International Symposium on Rarefied Gas Dynamics, pages 1427–1433, Weinheim, Germany, July 1990.

[45] M.N. Macrossan. Searching for a near neighbor particle in DSMC cells using pseudo-subcells. Journal of Computational Physics, 229(17):5857–5861, 2010.

[46] J.C. Maxwell. On the dynamical theory of gases. Proceedings of the Royal Society of London, 15:167–171, 1866.

[47] J.C. Maxwell. On stresses in rarefied gases arising from inequalities of temperature. Philosophical Transactions of the Royal Society of London, 170:231–256, 1879.

[48] E. Meiburg. Comparison of the Molecular Dynamics method and the direct simulation technique for flows around simple geometries. Physics of Fluids, 29:3107–3113, 1986.

[49] K. Moe and M.M. Moe. Gas-surface interactions and satellite drag coefficients. Planetary and Space Science, 53(8):793–801, 2005.

[50] NASA. UARS re-entry overview, Sep 2012.

[51] I. Newton. Philosophiæ Naturalis Principia Mathematica. Philosophical Transactions of the Royal Society, London, UK, 1687.

[52] W.D. Niven. The Scientific Papers of James Clerk Maxwell. Dover Publications, New York, USA, 1965.

[53] S. Nocilla. The surface re-emission law in free molecule flow. Rarefied Gas Dynamics, pages 327–346, 1963.

[54] E.S. Oran, C.K. Oh, and B.Z. Cybyk. Direct Simulation Monte Carlo: Recent advances and applications. Annual Review of Fluid Mechanics, 30:403–441, 1998.

[55] J.F. Padilla and I.D. Boyd. Assessment of gas-surface interaction models for computation of rarefied hypersonic flow. Journal of Thermophysics and Heat Transfer, 23(1):96–105, 2009.

[56] J. M. Picone, A. E. Hedin, and D. P. Drob. NRLMSISE-00 empirical model of the atmosphere: Statistical comparisons and scientific issues. Journal of Geophysical Research, 107, 2002b.

[57] M.D. Pilinski, B.M. Argrow, and S.E. Palo. Semiempirical model for satellite energy-accommodation coefficients. Journal of Spacecraft and Rockets, 47(6):951–956, 2010.

[58] M.D. Pilinski, B.M. Argrow, and S.E. Palo. Drag coefficients of satellites with concave geometries: Comparing models and observations. Journal of Spacecraft and Rockets, 48(2):312–325, 2011.

[59] P. Raj and B. Harris. Using surface transpiration with an euler method for cost-effective aerodynamic analysis. In AIAA Paper 93-3506, Aug. 1993.

[60] S. Rjasanow and W. Wagner. On time counting procedures in the DSMC method for rarefied gases. Mathematics and Computers in Simulation, 48(2):151–176, 1998.

[61] L.N. Sankar, S.Y. Ruo, and J.B. Malone. Application of surface transpiration in computational aerodynamics. In AIAA 24th Aerospace Sciences Meeting, AIAA Paper 86-0511, Reno, Nevada, Jan. 1986.

[62] T.J. Scanlon, E. Roohi, C. White, M. Darbandi, and J.M. Reese. An open source, parallel DSMC code for rarefied gas flows in arbitrary geometries. Computers and Fluids, 39(1):2078–2089, 2010.

[63] R. Schamberg. Analytic representation of surface interaction for free molecule flow with application to drag of various bodies. Technical Report R-339, The RAND Corporation, 1959. Available in PROJECT RAND, Aerodynamics of the Upper Atmosphere, Compiled by David J. Masson.

[64] R. Schamberg. On concave bodies in free molecule flow. Technical Report P-3164-1, The RAND Corporation, 1967.

[65] L. V. Schmidt. Introduction to Aircraft Flight Dynamics. AIAA Inc., Reston, Virginia, USA, 1998.

[66] R. Schunk and A. Nagy. Ionospheres: Physics, Plasma Physics, and Chemistry. Cambridge University Press, 2009.

[67] L.H. Sentman. Free molecule flow theory and its application to the determination of aerodynamic forces. Technical Report LSMC-448514, Lockheed Missles and Space Co., 1961.

[68] E. Stansbery and N. Johnson. NASA upper atmosphere research satellite (UARS) re-entry prediction and analysis. Technical Report JSC-CN-25915, NASA, Mar. 2012.

[69] S. Succi. The Lattice Boltzmann Equation for Fluid Dynamics and Beyond. Oxford University Press, 2001.

[70] T. Tokumasu and Y. Matsumoto. Construction of collision model of diatomic molecules (Improvement of energy transfer model and its verification). JSME International Journal Series B–Fluids and Thermal Engineering, 43(2):288–295, 2000.

[71] V. Utkin. Variable Structure Systems with Sliding Modes. IEEE Transactions on Automatic Control, 22(2):212–222, 1977.

[72] D.A. Vallado and D. Finkleman. A critical assessment of satellite drag and atmospheric density modeling. In AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii, USA, Aug. 2008.

[73] W.G. Vincenti and C.H. Kruger. Physical Gas Dynamics. Krieger Publishing Company, Malabar, Florida, USA, 1965.

[74] R. Yagel, D. Cohen, and A. Kaufman. Discrete Ray Tracing. IEEE Computer Graphics and Applications, 12(5):19–28, 1992.

[75] L. Zhang, W. Chen, D.S. Ebert, and Q. Peng. Conservative Voxelization. Visual Computer, 23(9–11):783–792, 2007.

[76] G. Zuppardi. DSMC implementation of the schamberg model: Comparison with the maxwell model. In Rarefied Gas Dynamics: 24th International Symposium, AIP 0-7354-0247-7/05, 2005.

Appendix A

DSMC Resolution Convergence Study

Table A.1 shows the range of values for a resolution convergence study for the NACA-0012 airfoil with the conditions given in Table 4.2 at an angle of attack of $\alpha = 45\,\text{deg}$. A square domain was used with side lengths of $L_x = L_y = 2.0\,\text{m}$ and the leading edge of the airfoil was positioned at $(x_{off}, y_{off}) = (0.5, 1.2)\text{m}$. The final simulation time t_{max} to was set to twice the steady-state time.

Table A.1: DSMC parameter values for convergence study for the 1m chord NACA-0012 airfoil at Kn =1, Ma =24.18.

Description	Variable	Nominal Value	Range for study
Num. cells total	N_c	40,000	[15000,150000]
Ratio of real-to-simulated particles	F_{num}	3×10^{13}	$[5 \times 10^{12}, 1 \times 10^{14}]$
Number of pixels per dir.	n_{vox}	500	[200,900]
Time to steady-state	t_{ss}	0.0015 s	[0.0005,0.0050] s

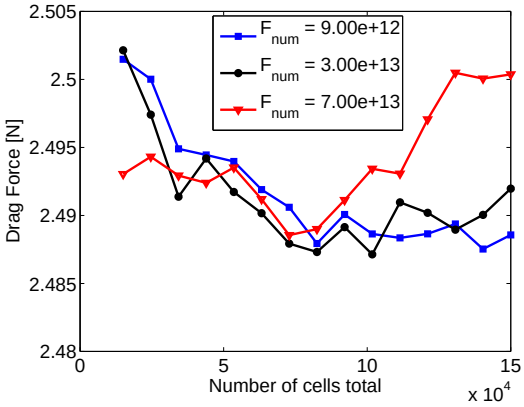
Results for the drag force on the airfoil as a function of each "pseudo-independent" variable, and are shown in Figure A.1. Some of the parameter space was covered with simulations with constant parameter values centered about the given nominal parameters in their respective categories. For example, Figure A.1a plots dimensional drag force as a function of the total number of DSMC collision/sample cells in the domain for three values of F_{num} (which is inversely related to the number of simulation particles).

The most converged result appears to be in time (Figure A.1d), which is expected as more sampling reduces the statistical scatter in the calculation of macroscopic values such as drag force. Particle resolution would be expected to converge as well, however no convergence trend is evident

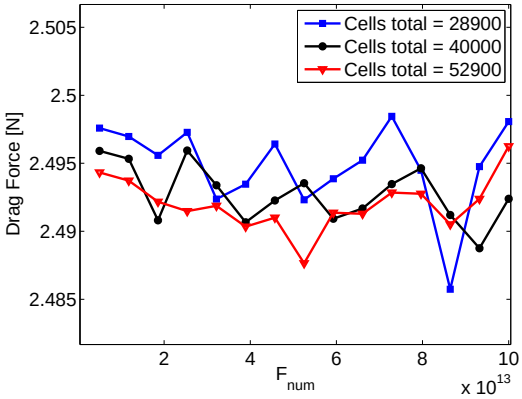
in Figure A.1b. Increasing the number of simulated molecules by reducing F_{num} reduces statistical scatter [3]. It appears that the smallest number of simulated molecules (roughly 120,000) is sufficient to yield a drag result to within one percent of the result for the maximum number (roughly 2.4 million). A more comprehensive study is required to elucidate the convergence behavior of this parameter. Figure A.1c indicates little advantage in pixel resolution greater than $n_{vox} = 400$, thus the nominal value of 500 is assumed sufficient for this case. Unexpectedly, cell resolution results shown in Figure A.1a do not show a clear convergence trend. As cells become smaller, they hold fewer particles and thus experience fewer collisions, which can prove to be statistically undesirable. This particular behavior of DSMC computations is documented by Bird [5]. Voldipar does not currently implement gradient-adapted sample cells. This lack of a sample-refining technique in the cell structure of the code can lead to poorer results for high cell resolutions. Both cell and particle resolution studies show the drag force variation of less than one percent. The parameter to which drag force is the most sensitive, geometry resolution, varies by about two percent. This sensitivity likely results from the calculation of surface forces by sampling particle collisions on surface panels. Sample accuracy increases with higher geometric resolution as more pixels are assigned to each line segment of the boundary. A set of resolutions that resulted in run times of approximately 10-12 minutes each were chosen with the expectation that results would be within five percent of converged values, or on the order of $\pm 0.01\,\mathrm{N}$. These values are given in Table A.2. All runs appear to converge to a value of $2.49\,\mathrm{N} \pm 0.01\,\mathrm{N}$. Assuming a frontal area given by $S \approx L_c \sin(\alpha)$ where $\alpha = 45\,\mathrm{deg}$ and $L_c = 1.0\mathrm{m}$, the converged drag coefficient for all studies appears to be approximately 1.90.

Table A.2: DSMC values for transition-regime source function generation for a 1m chord NACA-0012 airfoil at Kn = 1, Ma = 24.18, and $T_{in} = 252.9\mathrm{K}$.

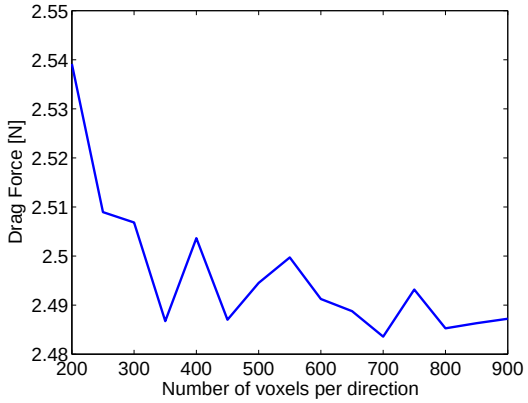
Description	**Variable**	**Value**
Number of cells total	N_c	22500
Ratio of real-to-simulated particles	F_{num}	7×10^{13}
Number of pixels per dir.	n_{vox}	500
Time to steady-state	t_{ss}	0.0015 s



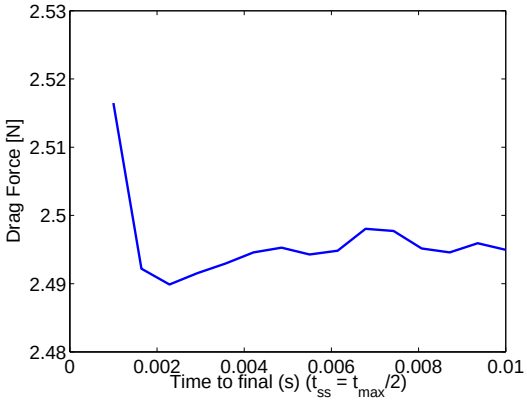
(a) Number of cells (total). Each direction has the same number of cells.



(b) F_{num} (Number of particles ranges approximately from 2.4×10^6 at $F_{num} = 5 \times 10^{12}$ to 1.2×10^5 at $F_{num} = 1 \times 10^{14}$).



(c) Geometry resolution (number of domain pixels per direction).



(d) Time to final completion t_{max}.

Figure A.1: DSMC pseudo-independent variable resolution convergence results for drag force for the airfoil in N_2 at $\alpha = 45$deg, Kn =1, Ma =24.18, and $T_{in} = 252.9$K.

Appendix B

Enforcing DSMC Inflow Equivalency

The energy and momentum balance analyses of Chapter 5 may be facilitated by numerical studies which measure the total energy and momentum crossing the boundaries of the domain of a test problem. The "true-equivalent" method of simulating motion where boundary conditions are adjusted to mimic a flow with changing angle of incidence requires special consideration when implemented in a particle method such as DSMC.

A example of "raw" DSMC (only small interval time sampling) with a moving boundary condition is given here. The rate of characteristic stream direction rotation is $\dot{\alpha} = 18,000$ deg/s, with Kn=0.01, Ma=28.40, N_2. There is an inconsistency with the formulation of this problem as equivalent to a case involving an actual moving body. The equivalent case is usually the imposition of a stream or characteristic velocity that changes in time, rotating the velocity vector by an angle appropriate to the desired angular rate (see Figure 5.2). In continuum methods, this is straightforward. In particle methods like DSMC, this velocity is imposed by setting the external stream boundary condition velocity to the desired value over time. When the angle changes beyond zero, more than one boundary now has a component of bulk flow that enters the domain. Each domain boundary must enter particles through itself. The number of particles that enter per unit time, per unit area is determined as a function of the external stream properties: density, temperature and velocity. When more than one boundary has a bulk flow component that directs inward, a greater number of particles will be calculated, in total, to enter the domain than would be when the stream is aligned with a single coordinate direction (i.e. $\alpha = n\pi/2$, $n = 0, 1, 2...$)

simply due to the triangle inequality. The "true" case (an actual, moving internal body) would naturally maintain a constant stream velocity oriented in this manner. Therefore imposing any stream boundary velocity between intervals bounded by $n\pi/2$ causes the domain to enter more molecules and thus transport more energy and momentum into the domain than the true case, despite the bulk velocity that the entering molecules are given being correct.

In order to correct this discrepancy, the desired stream velocity can be scaled in order to cause the entering number flux to be equivalent to the true case. Consider the case of flow entering from the left at velocity $\boldsymbol{V}$, moving right at a desired stream speed of $V_c = \|\boldsymbol{V}\|$, aligned with the horizontal direction as the true case. When the imposed stream velocity angle is changed to α in order to mimic an internal boundary's rotation by $-\alpha$, the left and bottom boundaries each change their inward components to $V_L = V_c \cos(\alpha)$ and $V_B = V_c \sin(\alpha)$, respectively. Now define V_L' and V_B' as the effective components that should result in a total domain entry flux equivalent to the flux at $\alpha = 0$ (the true case). The following conditions can be imposed:

$$V_L' + V_B' = V_c \tag{B.1}$$

$$\frac{V_B'}{V_L'} = \tan(\alpha) \tag{B.2}$$

Equation B.1 collapses the triangle inequality and Equation B.2 ensures the angle α is preserved. Solving this system gives the simple result

$$V_L' = \frac{V_c \cos(\alpha)}{\sin(\alpha) + \cos(\alpha)} \quad V_B' = \frac{V_c \sin(\alpha)}{\sin(\alpha) + \cos(\alpha)} \tag{B.3}$$

or

$$\boldsymbol{V}_{eff} = [V_L' \; V_B'] = \frac{1}{\sin(\alpha) + \cos(\alpha)} [V_c \cos(\alpha) \; V_c \sin(\alpha)]^T = \frac{1}{\sin(\alpha) + \cos(\alpha)} \boldsymbol{V} \tag{B.4}$$

which states that the desired stream velocity $\boldsymbol{V}$ need only be scaled by the scalar value of $(\sin(\alpha) + \cos(\alpha))^{-1}$. Figure B.1 illustrates how this scaling ensures a constant inward number flux for a specific case. There is one caveat to note about this new effective stream velocity. This is that $\|\boldsymbol{V}_{eff}\| \leq \|\boldsymbol{V}\|$, so if entering molecules use $\boldsymbol{V}_{eff}$ as their bulk velocity, they will be moving slower than the true case, even if the domain's total inward number flux is maintained. This is to say that

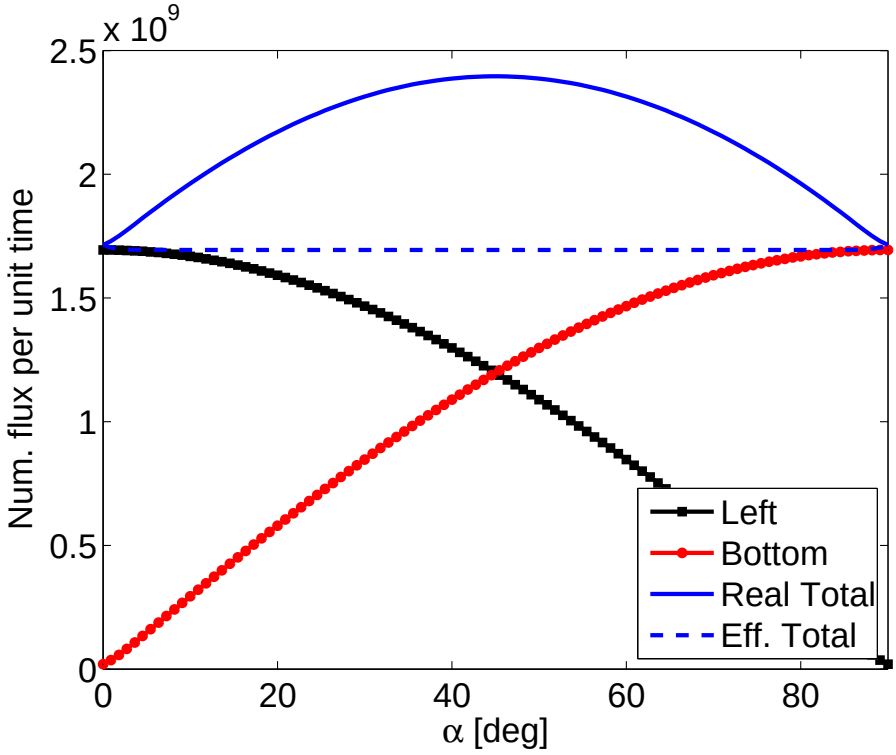
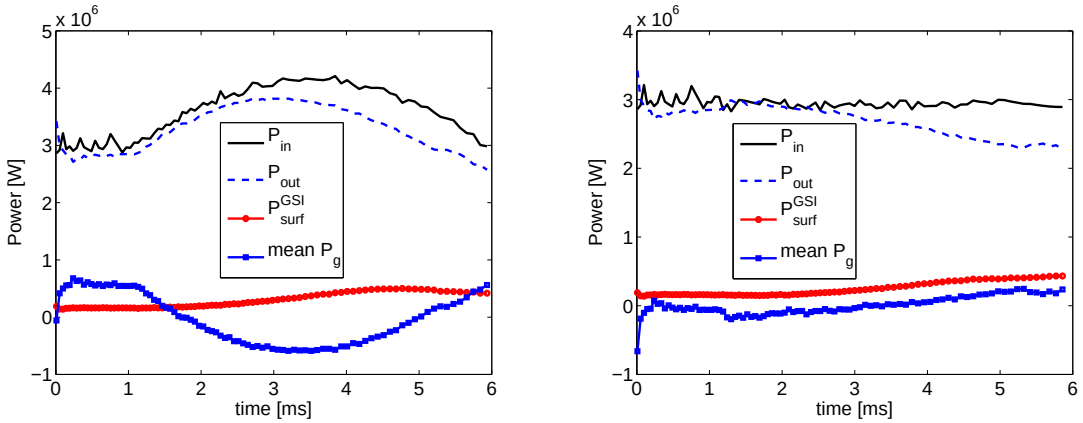


Figure B.1: Entry number flux per unit time for two inward directions (left and bottom), as well as their real sum, and adjusted sum according to Equation B.4.

the quantity to maintain is not number flux or velocity separately, but inward *power* flux. Power flux is enforced as constant when entering molecules must still use the original velocity $\boldsymbol{V}$ when they are created, while the calculation of number flux is performed with the effective velocity $\boldsymbol{V}_{eff}$.

When the simulation is allowed to function without scaling, the inflow power to the domain would be expected to follow the "real" curve in Figure B.1. When the adjustment method is applied, the inflow power should remain constant for all time (and angle). These expectations are verified in Figure B.2.

High frequency fluctuations in P_{in} occur due to the molecule entry algorithm in DSMC. Entry naturally can only occur for an integral number of molecules. The number of entering molecules is a real number and thus the remaining fraction of molecules not capable of being created on one iteration of the procedure are carried over and summed with the next iteration's calculated entry



(a) Unscaled.

(b) Scaled to match power flux at $\alpha = 0$.

Figure B.2: Power budget over the entire domain for Kn=0.01, Ma=28.40 with a moving boundary condition at rate $\dot{\alpha} = 18,000$ deg/s over a range of $\alpha = [0..90]$ deg. Motion begins at 1ms.

number, causing a regular oscillation of entry number. The large sinusoidal deviation (from the rough initial value) seen in Figure B.2a is absent in Figure B.2b. Removal of this artificial increase allows an accurate inspection of the power budget of the equivalent case. This budget indicates a steady increase of power transmitted to the gas itself (P_g) as well as to the body (P_{surf}^{GSI}) as the flow angle increases over time. This implies more energy storage inside the domain (either in the gas itself, as an increase in translational and rotational energy) due to the increasing bluntness of the body geometry to the flow direction.

Appendix C

Arbject Description

The 2D arbitrary object (termed the *Arbject*) is a figure defined by the five independently-selectable parameters: $\boldsymbol{p}_1, \boldsymbol{p}_2, R, d, \omega$. It may also be given in terms of one of its characteristic lengths L and an angle θ as: $\boldsymbol{p}_1, L, \theta, R, d, \omega$ in which case $\boldsymbol{p}_2 = L[\cos(\theta)\ \ \sin(\theta)]$. It is entirely asymmetric, contains concavity, various differing curvatures, sharp and shallow angles and straight lines. These features make it a reasonable choice for benchmarking purposes. Figure C.1 shows the arbject's geometry and visualizes these parameters.

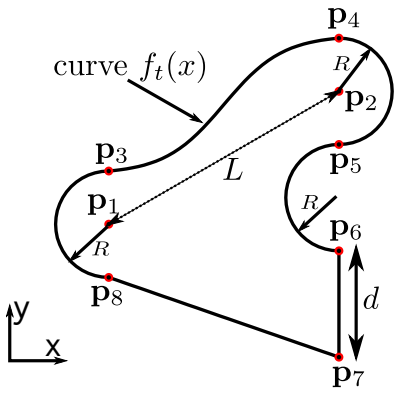


Figure C.1: Arbject geometry and illustration of parameters.

Eight points are defined that give key locations. The first two are the specified parameters

$\boldsymbol{p}_1, \boldsymbol{p}_2$, the remaining six are related as

$$\begin{aligned}
\boldsymbol{p}_3 &= \boldsymbol{p}_1 + [0\ R] \\
\boldsymbol{p}_4 &= \boldsymbol{p}_2 + [0\ R] \\
\boldsymbol{p}_5 &= \boldsymbol{p}_2 - [0\ R] \\
\boldsymbol{p}_6 &= \boldsymbol{p}_2 - [0\ 3R] \\
\boldsymbol{p}_7 &= \boldsymbol{p}_2 + [0\ 3R] - [0\ d] \\
\boldsymbol{p}_8 &= \boldsymbol{p}_1 - [0\ R]
\end{aligned} \tag{C.1}$$

The regions inscribed with R in Figure C.1 are circles. The top curve is determined by the following:

$$x_0 = \frac{1}{2}(p_{3x} + p_{4x}) \tag{C.2}$$

$$B = [\tanh(\omega(p_{3x} - x_0)) - \tanh(\omega(p_{4x} - x_0))]^{-1} \tag{C.3}$$

$$A_x = B(p_{3y} - p_{4y}) \tag{C.4}$$

$$A_y = -B\left[p_{3y}\tanh(\omega(p_{4x} - x_0)) + p_{4y}\tanh(\omega(p_{3x} - x_0))\right] \tag{C.5}$$

$$f_t(x) = A_x\tanh(\omega(x - x_0)) + A_y \tag{C.6}$$

To create the Arbject, each component should be created separately by evaluating respective equations of circles, lines and the curve $f_t(x)$ with as high a resolution as is desired. To use these components, it is often necessary to ensure that the points at which each component is evaluated do not overlap at their endpoints (the eight key points given as Equations C.1) and that the winding/plotting order of each component is the same. The latter requirement implies to create a set of ordered points that, when plotted, form a strongly simple (topologically) polygon.

The characteristic length of the Arbject may be specified in a number of ways. The length L, shown in Figure C.1 is one obvious option: $L = \|\boldsymbol{p}_2 - \boldsymbol{p}_1\|$. Another choice to consider may be $\max(L, R, d)$ or simply the maximum dimension of the entire figure.