On the Accuracy and Efficiency of the Direct Simulation Monte Carlo Method

by

Cyril Galitzine

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in The University of Michigan
2014

Doctoral Committee:

Professor Iain D. Boyd, Chair
Professor Edward W. Larsen
Professor Kenneth G. Powell
Assistant Professor Alexander G.R. Thomas

Edvard Munch (Norwegian, 1863-1944)

Ved Ruletten i Monte Carlo

At the Roulette Table in Monte Carlo, 1892

And if I have the gift of prophecy and comprehend all mysteries and all knowledge; if I have all faith so as to move mountains but do not have love, I am nothing.

1 Cor. 13:2

To my parents

ACKNOWLEDGEMENTS

I would like first and foremost to thank Professor Iain Boyd for his support and guidance throughout my time as a PhD student. Thank you for giving me the freedom to pursue my own research ideas and for all your time and help throughout the years. I am very thankful for having had the opportunity to use some of tools (such as MONACO) that you and previous students have worked hard to develop.

I would like to extend my gratitude to the other members of my doctoral committee, Professor Edward Larsen, Professor Kenneth Powell and Professor Alec Thomas for taking the time and making the effort to serve on my committee.

I would like to thank Professor Mark Kushner for launching and leading the DOE Plasma Science Center through which this work was funded. I am particularly grateful for the large computational resources that Professor Kushner put at my disposal, which have greatly facilitated my work.

I also thank current and past students and postdocs of the Nonequilibrium Gas and Plasma Dynamics Laboratory for the support I have received from them. I would like to particularly acknowledge all the help throughout the years I have received from Erin Farbar, Jon Burt and Ashley Verhoff.

I am extremely grateful for the unparalleled resources I have had access to during my time as a student at the University of Michigan. I would, in particular, like to thank Ms. Denise Phelps for all her help through the year. I would also like to acknowledge the countless other people from the cleaning staff to the librarians that make the University a great place to learn and work.

I am thankful for all the great teachers that have shaped me through the years at EPFL, Purdue and the University of Michigan. I am particularly grateful to Professor Alina Alexeenko at Purdue for making me discover the field of rarefied gas dynamics which is the subject of this thesis.

My time in Ann Arbor would not have been the same without many great friends. I would like to give a special mention to my former officemate Andrew Crow, Roland Florenz and Paul Giuliano. Outside of school, I am grateful for all my friends from St. Mary's.

Most importantly, I would like to thank my parents, Georges and Catherine Galitzine for their love, encouragement and support through my (many!) years as a student.

This work was carried out under the sponsorship of the Department of Energy grant DE-SC0001939 and I also benefited from a University of Michigan fellowship. I would like to thank the Department of Energy, the University and American and Michigan taxpayers for funding this research and my graduate education.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF NOTATIONS

Nomenclature

Roman Symbols

$\mathbf{1}_{(\cdot)}(\cdot)$	Indicator function
$\mathcal{A}$	Infinitesimal generator of generic Markov jump process
$\mathcal{A}_B$	Infinitesimal generator associated with the Kac equation
$\mathcal{B}$	Boltzmann equation collision operator
B	Boltzmann collision kernel
b	Impact parameter
k_B	Boltzmann constant
$\mathrm{Cov}(X, Y)$	Covariance of random variable X and Y
$\mathbb{E}[X]$	Expected value of random variable X
Exp	Exponential distribution
$\vec{E}$	Electric field
$\vec{e}$	Vector on the unit sphere
$\hat{f}$	Single particle velocity distribution function normalized by number density
f	Single particle velocity distribution function
F^N	Joint density probability for a system of N particles
f_{12}	Two-particle velocity distribution function
f_1	Single particle velocity distribution function
$f_{\mathrm{post,coll}}(\cdot)$	Distribution function of $\vec{e}$ after collision
$f_{\mathrm{sel,coll}}(\cdot, \cdot)$	Distribution function of colliding particle indices
F_{ij}	Force exerted by particle i on particle j
F_j	Field force on particle j
$J(\cdot, \cdot, \cdot, \cdot)$	Jump transformation operator

k	Number of time step or samples
m	Particle mass
$\mathbb{N}$	Set of all natural numbers
$\mathcal{N}$	Normal distribution
$\overline{N}$	Average number of particles
$\tilde{n}$	Number density of the numerical gas
N	Number of particles
n	Number density
N_{coll}	Number of collisions
N_{spec}	Number of different species in the simulation
N_c	Total number of cells in the simulation
N_l	Total number of nodes in the simulation
$N_{p,\text{want}}$	Desired number of particles
N_p	Total number of particles in the simulation
P	Probability
p	Pressure
P_{coll}	Probability of collision
P_{ab}	Probability that a particle of species a changes velocity during a collision with one of species b
$\vec{q}$	Heat flux
q	Electric charge
$Q\left(\cdot\right)$	Boltzmann equation collision operator
q_e	Elementary charge
$\mathbb{R}$	Set of real numbers
$\vec{r}$	Relative position vector
$\vec{r}_m$	Center of mass location
R	Diameter of interaction sphere, Relaxation factor
rand	Random number in $[0, 1]$
$\mathbb{S}$	Set of all vectors on the unit sphere
S	PIC shape factor

t	Time
T^k	Time of transition (jump) of Markov process
t_c	Time counter value
T_e	Plasma temperature
$\mathcal{U}$	Uniform distribution
$\vec{u}'$	Peculiar velocity
$\vec{u}$	Bulk velocity
$\overline{v}$	Current state of Markov process
$\overline{v}'$	State of Markov process after transition
$\vec{v}'$	Post-collision velocity
$\mathrm{Var}\left[X\right]$	Variance of random variable X
$\vec{v}$	Particle velocity
$\vec{v}^*$	Velocity determined by application of collision operator C_{coll}
$\vec{v}_m$	Center of mass velocity
$\vec{v}_r$	Relative velocity
V	Cell volume, State space of all particle velocities in one cell
v_r	Relative velocity magnitude
$\widetilde{W}_p$	Normalized cell weight
$W_{\mathrm{rel},j}$	Relative weight of species j
W_p	Cell weight

Abbreviations

ACF	Autocorrelation function
BC	Boundary condition
CFD	Computational fluid dynamics
CFL	Courant Friedrichs Lewy number
CIC	Cloud in cell
CLT	Central limit theorem
DSMC	Direct simulation Monte Carlo
iid	Independent and identically distributed
KE	Kinetic energy

Kn	Knudsen number
M	Mach number, Million
NGP	Nearest grid point
NTC	No time counter
PIC	Particle in cell
RHS	Right hand side
VHS	Variable hard sphere

Greek Symbols

$\Gamma\left(\cdot\right)$	Gamma function
γ_X	Autocovariance of variable X
$\delta\left(\cdot\right)$	Dirac delta function
Δt	Time step
Δt_{coll}	Time to the next collision
$\Delta\left(\cdot\right)$	Laplace operator
δ_{ab}	Kronecker delta
$\widetilde{\Delta} t$	Normalized time step
$\bar{\varepsilon}\left[\theta\right]$	Numerical error for property θ
$\tilde{\varepsilon}\left[\hat{\theta}\right]$	Convergence error for estimator $\hat{\theta}$
ε	Polar angle
$\hat{\theta}$	Statistical estimator of θ
θ	Statistical property
κ	Thermal conductivity
Λ	Plasma parameter
λ	Mean free path, Parameter of exponential distribution
λ_{ab}	Mean free path of collisions between species a and b
λ_D	Debye length
$\hat{\mu}\left(X\right)$	Statistical estimator of mean $\mu\left(X\right)$
$\mu\left(\cdot,\cdot\right)$	Markov transition function
μ_{ab}	Reduced mass of species a and b
$\mu_X, \mu\left(X\right)$	Mean of X

$\nu^{(1)}$	Single particle collision rate of the physical gas
ν_{ab}	Collision frequency between species a and b
ν_a	Total collision frequency of species a
$\tilde{\nu}^{(1)}$	Single particle collision rate of the numerical gas
ρ	Autocorrelation function, total neutral number density
$\hat{\sigma}^2\left(X\right)$	Statistical estimator of variance $\sigma^2\left(X\right)$
$\hat{\sigma}_{ab}$	Virtual cross section
σ	Differential cross section, Plasma conductivity, Standard deviation
$\sigma^2_X, \sigma^2\left(X\right)$	Variance of X
σ_μ	Viscosity cross section
σ_T	Total cross section
$\tilde{\sigma}_{ab}$	Cross section of numerical particles
$d\Sigma$	Sphere differential surface element
$d\Sigma_0$	Unit sphere differential surface element
$\overline{\overline{\tau}}$	Deviatoric Stress tensor
τ	Characteristic time
τ_{ab}	Mean collision time between species a and b
Φ	Potential of interaction force between two particles, flux of particles
ϕ	Plasma potential, Test function
χ	Collision deflection angle
ψ	Stream function
Ω	Solid angle, Physical domain of Boltzmann equation, Cell volume
ω_μ	Viscosity-temperature exponent
ω_{VHS}	variable-hard-sphere viscosity-temperature exponent

Supercripts/Subscripts

$\left(\cdot\right)^k$	At time step or sampling step k
$\left(\cdot\right)_0$	Reference value, or value at $t = 0$
$\left(\cdot\right)_{\max}$	Maximum value
$\left(\cdot\right)_{\min}$	Minimum value
$\left(\cdot\right)_{\mathrm{REF}}$	Reference value

$(\cdot)_e$	Related to electrons
$(\cdot)_n$	Related to neutrals
$(\cdot)_i$	Quantity related to cell i, quantity related to ions
$(\cdot)_{j,i}$	Quantity related to species j in cell i
$(\cdot)_j$	Quantity related to particle j or species j
$(\cdot)_l$	Quantity related to node l

Mathematical Symbols

$\langle\cdot\rangle$	Ensemble average or average
$\bar{\cdot}$	Average
$\sim$	proportional to, governed by

ABSTRACT

On the Accuracy and Efficiency of the Direct Simulation Monte Carlo Method

by

Cyril Galitzine

Chair: Iain D. Boyd

Rarefied gas flows can be found in many areas of science and engineering, such as around spacecraft reentering the earth's atmosphere or inside micro-electro-mechanical-systems (MEMS) devices. Such flows, which are either characterized by low densities or very small scales, are governed by the Boltzmann equation. The direct simulation Monte Carlo (DSMC) method is today by far the most popular simulation technique to solve the Boltzmann equation for rarefied flows. It is based on the use of computational particles that each represent a large number of physical particles. That number represents the weight of each individual particle. The computational particles are allowed to move and collide, much like gas particles. The simulation is then averaged over time to extract flow properties such as concentrations or velocities.

Because of the physical basis of the DSMC procedure, its accuracy has received relatively little attention compared to other simulation techniques. The first part of the thesis thus aims to characterize both the convergence and the accuracy of the DSMC method. The particular test case used for this investigation consists of an axisymmetric argon jet. Sampled flow quantities are found to be highly time-correlated, but the rate of convergence of their sampled averages is found to be well-predicted by a Central Limit Theorem taking that correlation into account. The influence of the number of particles and the time step on the accuracy of the simulation is then assessed. Their influence on the convergence and time correlation of sampled flow quantities is also studied.

The second part of the thesis addresses one of the major shortcomings inherent to the DSMC method, the fact that the number of computational particles in any one cell is directly proportional to its density. For flows with large density disparities or multiple species, this often results in some cells containing excessive numbers of particles and others too few. A possible solution is thus to use both cell weights and species weights that vary in space. As each particular flow is different, optimum weight fields also differ between simulations. In this thesis, a novel adaptive procedure for the weights and time step is proposed. Both are automatically varied as the simulation progresses to steady state. The performance of the adaptive procedure is then assessed for a test case consisting of two counterflowing jets. Significant performance improvements are observed. These are mostly due to an improved distribution of computational particles throughout the domain for each individual species.

In the last part of this thesis, a hybrid DSMC/Fluid method is applied to the simulation of a weakly ionized rarefied flow. Electrons are modeled via the use of an electron fluid model while DSMC is used to model the dynamics of all non-electron particles. The fluid model provides a partial differential equation for the plasma potential. The latter is in turn used to accelerate the heavy charged particles of the flow. Satisfactory results are obtained for both plasma parameters and gas properties.

CHAPTER I

Introduction

1.1 Flow regimes

All gas flows are characterized by the value of a non-dimensional parameter, the Knudsen number, denoted as Kn. It is defined as the ratio of the gas mean free path, λ, to a characteristic length scale of the flow L, i.e., $\mathrm{Kn} = \lambda/L$. The mean free path is, by definition, the average distance that each particle of the gas travels between collisions, while a basic geometric dimension of the system under consideration can be used as a characteristic length (e.g., the radius of a reentry capsule or the length of an aircraft). The value of Kn for a particular gas flow dictates the set of governing equations that can accurately be used to describe it. All dilute gases, i.e. gases where three-particle collisions and higher can be neglected, can be described by the Boltzmann equation. This encompasses all flows typically encountered within an aerospace context. The numerical solution of the Boltzmann equation, the subject of the present thesis, is, however, particularly challenging, so that, one seeks, when possible, to solve alternate equations. The different flow regimes and the corresponding equations that can be used to study them are shown in Fig.1.1. The continuum regime, described by the Navier-Stokes equation, is by far the most common gas flow regime due to its ubiquity in many important applications such as flows around aircraft [129] or inside jet engines [121]. In this flow regime, collisions between particles

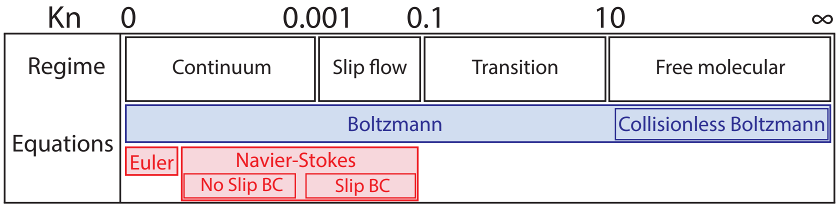


Figure 1.1: Flow regimes and governing equations

are so prevalent that particles only travel very short distances between collisions so that $\lambda \ll L$. Under those conditions, the distribution of the velocities of particles located in any small control volume is naturally a normal distribution, which is referred to as a *Maxwellian* distribution, centered about the mean velocity and with a variance related to the temperature of the flow. This Maxwellian characteristic for the velocity distribution functions becomes increasingly marked as the Knudsen number is reduced. The Navier-Stokes equations are obtained by taking statistical moments of the Boltzmann equation. This, however, results in a system of equations with more variables than equations that thus cannot be solved. The system is closed, i.e. made solvable, by using the fact that particles follow Chapman-Enskog distribution functions. This makes the system of equation substantially easier to solve, although a substantial amount of modeling is required whenever they are applied to turbulent flows [42]. Due to the Maxwellian assumption used to derive such equations, they, however, become increasingly inaccurate as the Knudsen number is increased. At the opposite end of the Knudsen number spectrum, when $\mathrm{Kn} \gtrapprox 10$, collisions between particles are relatively rare, so that $\frac{\lambda}{L} \gg 1$. This implies that the effect of collisions in the Boltzmann equation can be neglected, so that its collision term can be omitted, which renders it substantially easier to solve. Free molecular flow regimes are commonly encountered around satellites [47] as well as in many industrial processes [34]. The subject of the present thesis is the *transitional* regime characterized by

$0.1 \lessapprox \mathrm{Kn} \lessapprox 10$, in which collisions are prevalent enough, so that the free molecular assumption is not valid, while still not frequent enough, so as to force particles to adopt Maxwellian velocity distribution functions. As previously remarked, this makes the Navier-Stokes system of equations that was derived under that Chapman-Enskog assumption inaccurate, so that the Boltzmann equation has to be solved to obtain physical results.

This transition regime is commonly found in very different areas of science and engineering involving low densities and/or small scales. It occurs around spacecraft reentering earth between altitudes of approximately 80 and 130 Km due to the rarefied atmosphere [88] [84]. Below that zone, where the density of the atmosphere is greater, the flow around a vehicle can be considered in the continuum regime while above it, it can be assumed free molecular. This application field has, by far, received the greatest attention to date. The subject of this thesis, the direct simulation Monte Carlo technique, was initially developed [12] with that particular application in mind. Because of the large amount of research conducted in that area during the 1970s and 1980s and reduced interest in space exploration [32], this particular application is now receiving less research attention than in prior years. Another important category of flows in the transition regime is that encountered inside and in the plume of electric propulsion engines for satellites [69] [25]. They are noteworthy in that charged particles (ions and electrons) are present, albeit with small relative number densities, as the degree of ionization is at most a few percent, so that they are said to be *weakly* ionized. The presence of non-zero charge in the flow induces an electric and magnetic field, which in turn affects the motion of individual particles through the Lorentz force. Outside of the aerospace sector, weakly ionized flows in the transition regime are commonly encountered in materials processing [52] [130]. An increasingly important application of the transition regime is microscale flows [131] [93] such as those encountered in Micro-Electro-Mechanical-Systems (MEMS) which operate at relatively high Knudsen numbers because of their small length scales $L \ll 10^{-3}$ m. Less commonly studied flows in the transition regime for instance include that of many industrial processes such as freeze drying [4], astrophysical flows around comets [46] or in divertors of fusion devices [68].

The optimum engineering of any device, such as a capsule, a plasma etch reactor, or a micro-pump, requires that its behavior or performance be known during its design process. Before the advent of digital computers, this was accomplished through the building and testing of prototypes and by simplified analytical models. The conduct of such experiments is, however, costly and detailed non-intrusive measurements are difficult to obtain. Operating conditions are often impossible to replicate in an experimental setting, such as in the case of reentry flows, for which wind tunnels are limited to scaled models and short duration tests [11]. Simplified models are furthermore difficult to apply to complex geometries and/or to multiphysics problems and thus often lead to inaccurate results. The advent of digital computers combined with the development of numerical discretization or simulation techniques for the equations governing such flows and their effectiveness [102] [53] has led to an ever increasing use of numerical simulations during the design process. Simulations can be used to improve the design, as a large number of design iterations can be tested [86], be utilized to better understand the physics and to replace and/or aid in the design of experiments. This importance of simulations in the design process creates the need for accurate numerical methods, i.e. numerical schemes which introduce as few numerical artifacts as possible in the solution that they yield so that the latter may actually be the "true" solution of the physical equations being solved. Efficient numerical methods are also required, so that an accurate solution might be obtained with the least computational effort possible. This is because, despite increased computing power, many simulations still remain impossible to conduct today. Multiple simulations are also often required, especially as part of a design process, which creates an additional need for efficient simulation methods.

The most popular simulation technique for transition flows is the direct simulation Monte Carlo (DSMC) technique, which is the subject of this thesis. Since its invention in 1963 [12] it has been applied, amongst others, to the many different types of flows that were previously cited. Despite being widely used, many challenges remain to be addressed [16]. This thesis proposes to focus on two in particular. The first is that the accuracy of DSMC and its convergence are very dependent on the values of its numerical parameters. The effect of these parameters on the solution can only be quantified with great effort, which makes it difficult to determine whether results obtained with a particular simulation are accurate. The second major issue that is addressed is that, because of its formulation, DSMC is inefficient for flows with large density variations or large density differences between species. Those two specific issues are discussed in more specific terms in the following section.

1.2 Objectives of present work

The DSMC technique is built on the idea of using a relatively small number of computational particles $N_p \sim 10^6 - 10^9$, also called macro-particles, to represent all the actual physical particles that are present in a particular flow. The geometry of the flow that is simulated is divided into cells, with characteristic length Δx, while time is discretized with a time step Δt. At each time step, every computational particle of the domain, thus representing $W_p \sim 10^5 - 10^{10}$ actual particles, is first moved according to Newton's equation of motion and then allowed to collide with other particles that are located in the same cell in a manner consistent with the Boltzmann equation. The result of particles colliding and moving between cells is that the macroscopic properties, such as the number density or average particle velocity, evaluated in any single cell, fluctuate with time. The number density and velocity (or any other statistical property of the flow) averaged over many steps, can be shown to converge to that predicted by the steady state Boltzmann equation [148]. This averaging is accomplished by sampling their respective values at each time step. As relatively small time step values are usually employed, the values of macroscopic properties at consecutive time steps of each individual cell are correlated, i.e. they are not statistically independent. This has the consequence that the convergence rate of sampled averages with the number of time steps is relatively slow. The required number of time steps necessary to achieve the desired level of convergence for sampled quantities can only be determined a posteriori, i.e. by stopping the simulation when the sampled mean remains within a prescribed tolerance for a sustained period of time. That number of steps is, however, not known a priori, i.e. before the simulation, or during most of it when the sampled value changes slowly before only really converging for very large ($\sim 10^6$) numbers of samples. The first objective of this thesis is thus to quantify the level of correlation between sampled quantities and to determine what affects the rate of convergence of sampled quantities in a DSMC simulation. This is a rather fundamental question that has not previously been addressed. The aim of such an investigation is to find possible strategies to increase the efficiency of DSMC by decreasing the correlation between sampled quantities. The second goal is to determine whether the time correlation of sampled quantities can be used, either a priori, or during the early stages of the simulation, to determine how many steps are required to achieve a prescribed level of convergence. The sampled quantities of a DSMC simulation "converge" to those obtainable from the exact solution of the Boltzmann equation in the limit of an infinite number of particles $N_{\text{tot}} \to \infty \iff W_p \to 0$ and infinitesimally small time step $\Delta t \to 0$ [148]. The DSMC method will later be shown, in its collision phase, to solve a Boltzmann-like equation, the Kac equation, as opposed to the Boltzmann equation itself (this issue is discussed in Section 2.3.4). Throughout this thesis, the DSMC method will be considered to provide a solution of the Boltzmann equation, as this constitutes the stated intent of the method although that is born out by its formulation. The use of a finite number of particles and a non-zero time step in turn introduce an error, which will later be called *numerical error*, and denoted by $\overline{\varepsilon}$, in sampled quantities. Although this issue is well known by users of DSMC, it has received relatively little attention because typical error values are either fairly small when enough particles are present (although "enough" has yet to be quantified) or very large for cells with an insufficient number of particles. This is typically evidenced by jagged or non-physical contour plots. Many users of the method, however, content themselves with such results, knowing that better results could be obtained with more particles at greater computational expense. The second thrust of the thesis is thus to quantify the influence of the numerical parameters of the DSMC procedure on the accuracy of the solution, particularly the number of particles (or equivalently the particle weight $\frac{1}{W_p}$) and the time step Δt. This quantification of the error will consist, if possible, in establishing the order of the leading error term as a function of the number of particles and time step, i.e. to determine α and β in $\overline{\varepsilon} = C\left(\frac{1}{W_p}\right)^{\alpha}\Delta t^{\beta}$. The first two objectives of the thesis, which consist in respectively studying the convergence of samples in a DSMC simulation and the accuracy of those samples in a converged state, are reflected in its title.

The third goal of the thesis is to try to accelerate the convergence of samples and to improve their accuracy for the same computational cost, i.e. the same total number of particles. This is accomplished by varying both the value of the particle weight W_p and that of the time step Δt throughout the domain. Because each computational particle represents W_p actual particles, the number of particles N in a cell of volume V of number density n is given by $N = \frac{nV}{W_p}$. In a flow with large number density differences, this results in a few cells containing many particles while others very few, thereby resulting in a waste of computational resources. This issue can be addressed by letting the weight field $W_p\left(\vec{x}\right)$ conform to the number density of the flow such that $W_p\left(\vec{x}\right) = \frac{n(\vec{x})V(\vec{x})}{N_0}$ so as to obtain approximately the same number of particles, N_0, in all cells of the domain for all species. The value of the time step Δt in a particular cell, because of the formulation of DSMC, has to be less than the average time between collisions, called the mean collision time, τ_{mct}, in the physical gas. As $\tau_{\mathrm{mct}} \sim n^{-1}$, the densest regions of the flow dictate an inordinately small time step to the rest of the flow. The time step furthermore has to be small enough so that particles do not cross more than one cell on average without being allowed to collide. The time step can thus be maximized by tailoring it to each region of the flow, while ensuring that all requirements are met, by letting it vary through space by defining it as $\Delta t\left(\vec{x}\right)$. The fourth and last objective of this work is the development and application of a hybrid DSMC/fluid method to partially ionized rarefied flows in the transition regime. These flows are "standard" rarefied flows where electrons and ions (although, here, only one type, i.e., Ar^{+} will be considered) are present. The simulation method is then applied to the computation of a partially ionized rarefied flow between two counter flowing jets. In that simulation formalism, the movement and collision of particles are handled via the DSMC method with the added presence of an electric field. The electric field $\vec{E}$ accelerates charged particles via the Lorentz Force, while its value is determined by solving electron fluid equations. The fluid equations, which govern the motion of the electrons in the flow, are obtained by assuming that electrons are in equilibrium, i.e. that they follow a Maxwellian distribution.

1.3 Outline of thesis

In Chapter II, a rigorous derivation of the Boltzmann equation, the governing equation of all gas flow studied in this thesis, is presented with particular emphasis on all assumptions that have to be made. A brief overview of the formulation of the numerical method used to solve the Boltzmann equation, direct simulation Monte Carlo, is then presented. Finally, a derivation of the collision phase of the DSMC procedure from a master Boltzmann equation is detailed. This will provide a justification as to the form of the DSMC procedure. In Chapter III, a rigorous analysis framework for the study of the convergence of DSMC simulations is presented. This framework is then used to analyze the convergence of the sampled number density and velocity values for a test case consisting of a single axisymmetric jet. In Chapter IV, following the convergence analysis of the previous chapter, a formalism to study the accuracy of DSMC simulations is introduced. This formalism is then used to investigate the effect of the number of particles and time step for the same test case that was previously considered. Following the numerical studies of the two previous chapters, which used a standard DSMC implementation, in Chapter V, a new DSMC procedure to improve simulation efficiency is introduced. It is based on the use of an automated formula, which determines the optimum spatial distribution of cell weights, species relative weights and time step. The procedure is presented in detail and applied to the simulation of the flow between two counter flowing jets that contains multiple species. Finally, in Chapter VI, a hybrid fluid electron/DSMC approach is presented. All the additional steps in the DSMC algorithm made necessary by the presence of charged particles are presented, while the electron fluid equations are rigorously derived from the Boltzmann equation. The adaptive procedure of Chapter V is then used in conjunction with the hybrid method for the simulation of a flow between a weakly ionized plasma jet and a jet of neutrals. Finally in Chapter VII, key findings as well as the notable contributions of the present thesis are summarized.

CHAPTER II

The Boltzmann equation and its numerical solution

In this chapter, the Boltzmann equation is introduced. It is rigorously derived from the Liouville equation for a system of N particles and all underlying assumptions are clearly stated. The DSMC method is then introduced and its basic formulation detailed. The collision part of the DSMC technique is then rigorously derived from the Boltzmann equation and interpreted as a Markov process.

2.1 The Boltzmann equation

2.1.1 A derivation of the Boltzmann equation

The Boltzmann equation can be derived following many different approaches. A rigorous derivation, based on the BBGKY[1] equation hierarchy will be outlined in this section following the presentation given in [49] and [79]. Historically, however, the Boltzmann equation was derived following a more physical approach [22] which is well summarized in [103]. The starting point of the derivation is the Liouville equation

---

[1] Born-Bogoliubov-Gren-Kirkwood-Yvon

for the density probability[2]

$$F^N\left(\vec{x}_1, \vec{x}_2, ..., \vec{x}_N, \vec{v}_1, \vec{v}_2, ..., \vec{v}_N, t\right)$$

of a system of N different particles of mass m located at $\{\vec{x}_j\}_{j=1}^N$ with velocities $\{\vec{v}_j\}_{j=1}^N$

$$\frac{\partial F^N}{\partial t} + \sum_{j=1}^N \vec{v}_j \frac{\partial F^N}{\partial \vec{x}_j} + \sum_{j=1}^N \frac{F_j + \sum_{k \neq j}^N \vec{F}_{jk}}{m} \cdot \frac{\partial F^N}{\partial \vec{v}_j} = 0 \tag{2.1}$$

where $\vec{F}_j + \sum_{k \neq j}^N \vec{F}_{jk}$ is the total force exerted on particle j by an external field as well as all other $N-1$ particles. Equation (2.1) can in turn be used to derive the following evolution equation for the single particle distribution function f_1:

$$\frac{\partial f_1\left(\vec{x}_1, \vec{v}_1, t\right)}{\partial t} + \vec{v}_1 \cdot \frac{\partial f_1}{\partial \vec{x}_1} + \frac{\vec{F}_1}{m} \cdot \frac{\partial f_1}{\partial \vec{v}_1} + \iint\limits_{\mathbb{R}^3\ \mathbb{R}^3} \frac{\vec{F}_{12}}{m} \cdot \frac{\partial f_{12}\left(\vec{x}_1, \vec{x}_2, \vec{v}_1, \vec{v}_2, t\right)}{\partial \vec{v}_1} d\vec{x}_2\, d\vec{v}_2 = 0. \tag{2.2}$$

This leads to an expression for the probable number $\langle dN_1 \rangle$ of particles inside both volume $d\vec{x}_1$ of physical space and volume $d\vec{v}_1$ of velocity space:

$$\langle dN_1 \rangle = f_1\left(\vec{x}_1, \vec{v}_1, t\right) d\vec{v}_1 d\vec{x}_1. \tag{2.3}$$

The equation for f_1, however, depends on the two particle distribution functions f_{12} which gives the probability $\langle dN_{12} \rangle$ of simultaneously finding particle 1 inside volume $d\vec{x}_1$ about $\vec{x}_1$ of physical space with velocity vector inside volume $d\vec{v}_1$ about $\vec{v}_1$ of velocity space and particle 2 inside $d\vec{x}_2$, $d\vec{v}_2$ about $\vec{x}_2$, $\vec{v}_2$ respectively:

$$\langle dN_{12} \rangle = f_{12}\left(\vec{x}_1, \vec{x}_2, \vec{v}_1, \vec{v}_2, t\right) d\vec{v}_1 d\vec{x}_1 d\vec{v}_2 d\vec{x}_2. \tag{2.4}$$

---

[2] By definition, $F^N\left(\vec{x}_1, \vec{x}_2, ..., \vec{x}_N, \vec{v}_1, \vec{v}_2, ..., \vec{v}_N, t\right)\ d\vec{x}_1\ d\vec{x}_2\ ...\ d\vec{x}_N\ d\vec{v}_1\ d\vec{v}_2\ ...\ d\vec{v}_N$ is the simultaneous probability that particle 1 is located in volume $d\vec{x}_1$ about $\vec{x}_1$ of physical space and volume $d\vec{v}_1$ about $\vec{v}_1$ of velocity space and that particle 2 is in $d\vec{x}_2$ about $\vec{x}_2$ and in $d\vec{v}_2$ about $\vec{v}_2$ and so on...

In the particular case of an absence of correlation between particles, their statistical properties are independent so that:

$$f_{12}\left(\vec{x}_1,\vec{x}_2,\vec{v}_1,\vec{v}_2,t\right)=f_1\left(\vec{x}_1,\vec{v}_1,t\right)\ f_2\left(\vec{x}_2,\vec{v}_2,t\right), \tag{2.5}$$

a result which will later be used in the derivation of the Boltzmann equation. The evolution equation for f_{12} can be shown to take the following form

$$\frac{\partial}{\partial t}f_{12}\left(\vec{x}_1,\vec{x}_2,\vec{v}_1,\vec{v}_2,t\right)+\vec{v}_1\cdot\frac{\partial f_{12}}{\partial\vec{x}_1}+\vec{v}_2\cdot\frac{\partial f_{12}}{\partial\vec{x}_2}+\frac{\vec{F}_1+\vec{F}_{12}}{m}\cdot\frac{\partial f_{12}}{\partial\vec{v}_1}+\frac{\vec{F}_2+\vec{F}_{21}}{m}\cdot\frac{\partial f_{12}}{\partial\vec{v}_2}+ \\ \iint\limits_{\mathbb{R}^3\ \mathbb{R}^3}\frac{\vec{F}_{13}}{m}\cdot\frac{\partial f_{123}}{\partial\vec{v}_1}d\vec{x}_3\ d\vec{v}_3+\iint\limits_{\mathbb{R}^3\ \mathbb{R}^3}\frac{\vec{F}_{23}}{m}\cdot\frac{\partial f_{123}}{\partial\vec{v}_2}d\vec{x}_3\ d\vec{v}_3=0 \tag{2.6}$$

which depends on f_{123}, the three particle distribution function. An evolution equation for f_{123} in turn depends on f_{1234}, etc. . The system of N equations that is generated by considering evolution equation for f_s with $1\leq s\leq N$ is referred to as the BBGKY system of equations (or hierarchy). In order to be of any practical use, the system has to be truncated at a low order such as at $s=2$ which only leaves an evolution equation for f_1, Eq. (2.2) which is, however, *open*, i.e. not solvable as is, because of the f_{12} term. The equation is *closed*, i.e. made solvable, by invoking a series of simplifying assumptions which in turn dictate the range of applicability of any equation derived from such a system. We follow the rigorous approach presented in [152] [49] where greater detail can be found.

In order to obtain the Boltzmann equation, a number of assumptions are made. Hypothesis A: The density is assumed to be low enough so that triple body interactions can be neglected. This was referred to as the dilute gas assumption in the Introduction. This has the consequence that all terms involving f_{123} can be neglected in Eq. (2.6) which thus becomes:

$$\frac{\partial}{\partial t} f_{12}\left(\vec{x}_1, \vec{x}_1, \vec{v}_1, \vec{v}_2, t\right) + \vec{v}_1 \cdot \frac{\partial f_{12}}{\partial \vec{x}_1} + \vec{v}_2 \cdot \frac{\partial f_{12}}{\partial \vec{x}_2} + \frac{\vec{F}_1 + \vec{F}_{12}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_1} + \frac{\vec{F}_2 + \vec{F}_{21}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_2} = 0 \tag{2.7}$$

Equations (2.2) and (2.7) thus form a fully determined equation system for f_1 and f_{12} which will further be simplified.

Hypothesis B: Forces between particles are assumed to have a short range of interaction and to be negligible for distances greater than the radius of interaction R. So that with $\vec{r} \triangleq \vec{x}_2 - \vec{x}_1$, $\vec{X}_{12}\left(\vec{r}\right) = \vec{0} \quad \forall \left\|\vec{r}\right\| < R$. Rewriting Eq. (2.7), in terms of $\vec{r}$, the following is obtained

$$\frac{\partial}{\partial t} f_{12}\left(\vec{x}_1, \vec{x}_1, \vec{v}_1, \vec{v}_2, t\right) + \vec{v}_1 \cdot \frac{\partial f_{12}}{\partial \vec{x}_1} + \left(\vec{v}_2 - \vec{v}_1\right) \cdot \frac{\partial f_{12}}{\partial \vec{r}} + \frac{\vec{F}_1 + \vec{F}_{12}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_1} + \frac{\vec{F}_2 + \vec{F}_{21}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_2} = 0 \tag{2.8}$$

It will later be useful to consider this equation in terms of relative coordinates, i.e.

$$\vec{r}_m \triangleq \frac{\vec{x}_1 + \vec{x}_2}{2}, \qquad \vec{r} = \vec{x}_1 - \vec{x}_2, \qquad \vec{v}_m \triangleq \frac{\vec{v}_1 + \vec{v}_2}{2}, \qquad \vec{v}_r \triangleq \vec{v}_1 - \vec{v}_2$$

where $\vec{r}_m$ and $\vec{v}_m$ designate the position and velocity of the center of mass of the two particles, respectively, while $\vec{v}_r$ is the relative velocity between the two particles. In those new coordinates, Eq. (2.8) becomes:

$$\begin{aligned} &\frac{\partial}{\partial t} f_{12}\left(\vec{r}_m, \vec{r}, \vec{v}_m, \vec{v}_r, t\right) + \vec{v}_m \cdot \frac{f_{12}}{\partial \vec{r}_m} + \vec{v}_r \cdot \frac{f_{12}}{\partial \vec{r}} + \\ &\qquad \frac{\vec{F}_1 + \vec{F}_{12}}{m} \cdot \left(\frac{1}{2}\frac{\partial f_{12}}{\partial \vec{v}_m} - \frac{\partial f_{12}}{\partial \vec{v}_r}\right) + \frac{\vec{F}_2 + \vec{F}_{21}}{m} \cdot \left(\frac{1}{2}\frac{\partial f_{12}}{\partial \vec{v}_m} + \frac{\partial f_{12}}{\partial \vec{v}_r}\right) = 0 \end{aligned} \tag{2.9}$$

Momentarily putting the last equation aside, we proceed to integrate Eq. (2.8) over the sphere of interaction, S, $S = \{\vec{r} \in \mathbb{R}^3 \mid \left\|\vec{r}\right\| \leq R\}$ and over all possible $\vec{v}_2$ velocities, i.e. $\vec{v} \in \mathbb{R}^3$ which yields

$$\underbrace{\iint\limits_{\mathbb{R}^3\ S} \frac{\partial f_{12}}{\partial t}\, d\vec{r}\, d\vec{v}_2}_{\text{(I)}} + \underbrace{\iint\limits_{\mathbb{R}^3\ S} \vec{v}_1 \cdot \frac{\partial f_{12}}{\partial \vec{x}_1}\, d\vec{r}\, d\vec{v}_2}_{\text{(II)}} + \underbrace{\iint\limits_{\mathbb{R}^3\ S} (\vec{v}_2 - \vec{v}_1) \cdot \frac{\partial f_{12}}{\partial \vec{r}}\, d\vec{r}\, d\vec{v}_2}_{\text{(III)}} + \underbrace{\iint\limits_{\mathbb{R}^3\ S} \frac{\vec{F}_1 + \vec{F}_{12}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_1}\, d\vec{r}\, d\vec{v}_2}_{\text{(IV)}} + \underbrace{\iint\limits_{\mathbb{R}^3\ S} \frac{\vec{F}_2 + \vec{F}_{21}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_2}\, d\vec{r}\, d\vec{v}_2}_{\text{(V)}} = 0. \quad (2.10)$$

The (V) term can be shown to be null if $\vec{F}_2$ and $\vec{F}_{21}$ satisfy Eq. (6.9), which is indeed the case if $\vec{F}_2$ is the Lorentz force and if $\vec{F}_{21}$ is independent of velocity.

Hypothesis C: Inside the sphere of interaction, field effects are assumed to be negligible compared to interaction forces between particles.

This hypothesis is indeed physically realistic, as, during a collision, particles are subjected to very intense forces far in excess of those produced by the strongest electric or magnetic field. $\vec{F}_1$ can thus be assumed to be negligible compared to $\vec{F}_{12}$, i.e.

$$F_1 \ll F_{12} \quad (2.11)$$

so that

$$\text{(IV)} \approx \iint\limits_{\mathbb{R}^3\ S} \frac{\vec{F}_{12}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_1}\, d\vec{r}\, d\vec{v}_2 \quad (2.12)$$

which thus allows the term to be neglected by the same argument as that used for (V).

Hypothesis D: The spatial variation of f_{12} over the length scale of the interaction is assumed to be small compared to f_{12}.

As f_{12} essentially depends on r, and if $\vec{x}_1$ and $\vec{x}_2$ are inside the sphere of interaction, $|\frac{\partial f_{12}}{\partial r}| \sim \frac{f_{12}}{R}$ this means that

$$|\frac{\partial f_{12}}{\partial r}| R \ll f_{12} \quad (2.13)$$

so that term (II) is negligible compared to term (III)

Hypothesis E: The temporal variation of f_{12} during the course of binary interaction is assumed to be small so that term (I) can be neglected in Eq. (2.10).

Denoting the characteristic time of a binary interaction by τ with $\tau \sim \frac{R}{|\vec{v}_2-\vec{v}_1|}$ (by a dimensional argument), this amounts to

$$|\frac{\partial f_{12}}{\partial t}|\tau \ll f_{12}, \tag{2.14}$$

so that term (I) can be neglected compared to (III). Following hypotheses C, D and E, Eq. (2.10) can thus be rewritten as

$$\iint\limits_{\mathbb{R}^3\ S} (\vec{v}_2 - \vec{v}_1) \cdot \frac{\partial f_{12}}{\partial \vec{r}}\ d\vec{r}\ d\vec{v}_2 + \iint\limits_{\mathbb{R}^3\ S} \frac{\vec{F}_{12}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_1}\ d\vec{r}\ d\vec{v}_2 = 0, \tag{2.15}$$

so that substituting for the F_{12} term in Eq. (2.2) using Eq. (2.15) yields

$$\frac{\partial f_1}{\partial t} + \vec{v}_1 \cdot \frac{\partial f_1}{\partial \vec{x}_1} + \frac{\vec{F}_1}{m} \cdot \frac{\partial f_1}{\partial \vec{v}_1} = \mathcal{B}\left(f_{12}\right) \tag{2.16}$$

with

$$\mathcal{B}\left(f_{12}\right) \triangleq \iint\limits_{\mathbb{R}^3\ S} (\vec{v}_2 - \vec{v}_1) \cdot \frac{\partial f_{12}}{\partial \vec{r}}\ d\vec{r}\ d\vec{v}_2. \tag{2.17}$$

We now proceed to simplify this integral by assuming that all collisions are microreversible (Hypothesis F). This means that collisions are deterministic and time reversible. Using the divergence theorem, Eq. (2.17) becomes

$$\mathcal{B}\left(f_{12}\right) = \iint\limits_{\mathbb{R}^3\ \Sigma} (\vec{v}_2 - \vec{v}_1) \cdot \vec{n}\ f_{12}\ d\Sigma\ d\vec{v}_2 \tag{2.18}$$

where $d\Sigma$ designates the surface element of the sphere of radius R, denoted Σ, $\vec{n}$ being the outward pointing normal of that same sphere. We now proceed to divide sphere Σ into two separate hemi-spheres A and B according to the plane that is perpendicular to $\vec{v}_2 - \vec{v}_1$. This allows the following splitting of the integral in Eq. (2.17):

$$\iint\limits_{\mathbb{R}^3\ \Sigma} f_{12}\left(\vec{x}_1, \vec{x}_2, \vec{v}_1, \vec{v}_2\right)\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 = \\ \iint\limits_{\mathbb{R}^3\ \Sigma_A} f_{12}\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 + \iint\limits_{\mathbb{R}^3\ \Sigma_B} f_{12}\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 \quad (2.19)$$

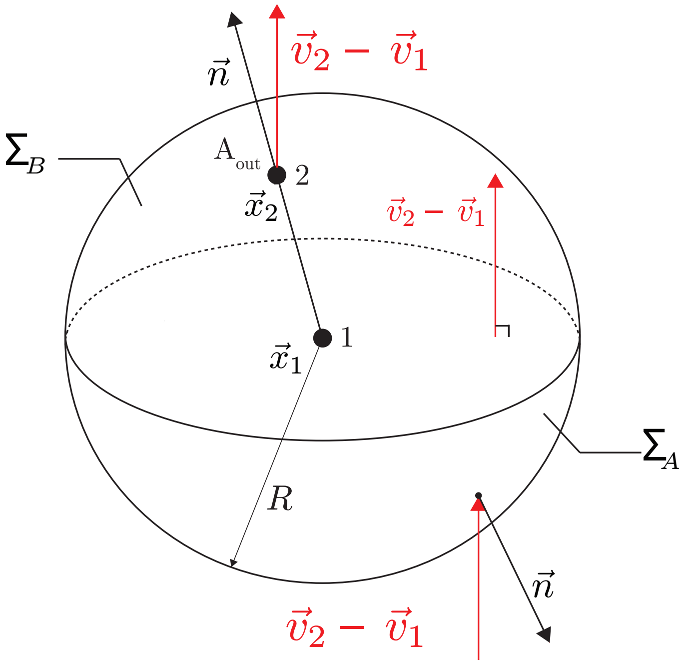


Figure 2.1: Geometrical framework for the calculation of the integrals in Eq. (2.19)

The surface of hemi-sphere A, Σ_A, is characterized by $(\vec{v}_2 - \vec{v}_1)\cdot\vec{n} < 0$, while that of B, Σ_B, is characterized by $(\vec{v}_2 - \vec{v}_1)\cdot\vec{n} > 0$. Both are shown in Fig. 2.1. We can interpret this integral in terms of the relative motion of two particles during their collision as shown in Fig 2.1. Particle 1 is assumed fixed, while particle 2 penetrates into the interaction sphere at point A_{in} before leaving at point A_{out}. The first integral over hemi-sphere A, can be interpreted as the integral over all possible pre-collision states between the two particles. We now make the following hypothesis:

Hypothesis G: There is no correlation between every two particles that participate in a collision. This assumption, originally called *Stosszahlansatz* by Boltzmann in [22] (See [31] for an English translation) is often referred to as the "molecular chaos assumption".

At point A_{in}, the particles have yet to undergo the collision, so that from the hypothesis and following Eq. (2.5),

$$f_{12}\left(\vec{x}_1, \vec{x}_2, \vec{v}_1, \vec{v}_2\right) = f_1\left(\vec{x}_1, \vec{v}_1\right) f_1\left(\vec{x}_2, \vec{v}_2\right) \qquad \forall \vec{x}_2 \in \Sigma_A. \tag{2.20}$$

This allows the first integral in Eq. (2.19), using hypothesis C to neglect spatial variations between $\vec{x}_1$ and $\vec{x}_2$, to be rewritten as:

$$\iint\limits_{\mathbb{R}^3\ \Sigma_A} f_{12}\left(\vec{v}_2 - \vec{v}_1\right) \cdot \vec{n}\ d\Sigma\ d\vec{v}_2 = \iint\limits_{\mathbb{R}^3\ \Sigma_A} \left(\vec{v}_2 - \vec{v}_1\right) f_1\left(\vec{x}, \vec{v}_1\right) f_1\left(\vec{x}, \vec{v}_2\right) d\Sigma\ d\vec{v}_2 \tag{2.21}$$

We note in passing that the assumption of molecular chaos is not valid after the collision, since the latter causes the velocities of the two particles to become correlated. The value of f_{12} after the collision thus has to be related to f_1 in some other manner. This can be accomplished by considering the trajectory of the two particles during their interaction. We notice that the evolution equation for f_{12}, Eq. (2.9), can be considered as a Liouville equation for the two particle system under consideration composed of particles 1 and 2. Placing ourselves in the center of mass reference frame, so that $\frac{\partial}{\partial \vec{r}_m}\left(\cdot\right) = \vec{0}$, and $\frac{\partial}{\partial \vec{v}_m}\left(\cdot\right) = \vec{0}$ and using hypothesis C:

$$\frac{\partial}{\partial t} f_{12}\left(\vec{r}_m, \vec{r}, \vec{v}_m, \vec{v}_r, t\right) + \vec{v}_r \cdot \frac{f_{12}}{\partial \vec{r}} - \frac{\vec{F}_{12}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_r} + \frac{\vec{F}_{21}}{m} \cdot \frac{\partial f_{12}}{\partial \vec{v}_r} = 0 \tag{2.22}$$

In order to satisfy hypothesis F, we further assume that the force $\vec{F}$ acting between particles is a central force (aligned with $\vec{r}$) and that it is derived from a potential Φ, i.e that

$$\vec{F}_{ij} = -\frac{\partial \Phi\left(\vec{r}\right)}{\partial\left(\vec{x}_i - \vec{x}_j\right)} \tag{2.23}$$

which, dropping the time derivative following hypothesis E, allows Eq. (2.22) to be rewritten as

$$\vec{v}_r \cdot \frac{f_{12}}{\partial \vec{r}} - \frac{2}{m}\frac{\partial \Phi\left(\vec{r}\right)}{\partial \vec{r}} \cdot \frac{\partial f_{12}}{\partial \vec{v}_r} = 0 \tag{2.24}$$

We recognize that f_{12} is constant along phase space trajectories that are defined as follows:

$$\begin{cases} \frac{d\vec{r}(t)}{dt} = \vec{v}\left(t\right) \\ \frac{d\vec{v}_r(t)}{dt} = -\frac{1}{m}\frac{\partial \Phi}{\partial \vec{r}}\left(\vec{r}\left(t\right)\right) \\ \vec{r}\left(t=0\right) = \vec{r}_0, \quad \vec{v}_r\left(t=0\right) = \vec{v}_0 \end{cases} \tag{2.25}$$

which corresponds exactly to the equations of motion of the particles during the collision in the center of mass reference frame. This means that the value of f_{12} is constant along the trajectories of particle 2 during a collision with particle 1 in the center of mass reference frame which is depicted in Fig. 2.1. Thus, to find the value of f_{12} at a given A_{out} point on hemi-sphere Σ_B, we only need to find the value of f_{12} at the corresponding A_{in} point. From hypothesis D, the variation of f_2 on the surface of spheres Σ_A and Σ_B is neglected, which means that we only need to find the initial value of $\vec{v}_r$ that yields the specified $\vec{v}_r$ at point A_{out}. We first examine some symmetry properties of trajectories, as described by Eq. (2.25). We designate a solution of system Eq. (2.25) as follows:

$$\vec{r}\left(t\right) = \vec{\Gamma}\left(\vec{r}_0, \vec{v}_{r,0}, t\right) \tag{2.26}$$

where $\vec{\Gamma}$ describes a particular trajectory with initial conditions $(\vec{r}_0, \vec{v}_{r,0})$. In the frame of reference depicted in Fig. 2.1, the P parity transformation is considered

$P : (\vec{r}, \vec{v}_r) \rightarrow \left(\tilde{\vec{r}}, \tilde{\vec{v}}_r\right)$ which is defined as

$$\tilde{\vec{r}} = -\vec{r} \tag{2.27}$$

$$\tilde{\vec{v}}_r = -\vec{v}_r \tag{2.28}$$

Rewriting Eqs. (2.25) in terms of the new variables, the following is obtained:

$$\begin{cases} \frac{d\tilde{\vec{r}}}{dt} = \tilde{\vec{v}}_r(t) \\ \frac{d\tilde{\vec{v}}_r}{dt} = -\frac{1}{m}\frac{\partial \Phi}{\partial \tilde{\vec{r}}}\left(\tilde{\vec{r}}\right) \\ \tilde{\vec{r}}(0) = -\vec{r}_0 \qquad \tilde{\vec{v}}_r(0) = -\vec{v}_{r,0} \end{cases} \tag{2.29}$$

which, by comparison with the original system, Eq. (2.25), and using Eq. (2.26), admits the following trajectories:

$$\tilde{\vec{r}}(t) = \vec{\Gamma}\left(-\vec{r}_0, -\vec{v}_{r,0}, t\right) \tag{2.30}$$

This signifies that the transformation of any trajectory by parity is also a trajectory (characterized by opposite values for the initial position and velocity vectors). We now examine the symmetry of trajectories with respect to time. The time at which particle 2 leaves the interaction sphere is denoted by t_e. The time reversal transformation $T : t \rightarrow \tilde{t}$ is next considered which defines $\tilde{t}$ as follows

$$\tilde{t} = t_e - t. \tag{2.31}$$

By a property of the time reversal symmetry, $\tilde{\vec{v}} = -\vec{v}_r$ while position is unaffected, i.e. $\tilde{\vec{r}} = \vec{r}$, and $\frac{d}{d\tilde{t}}(\cdot) = -\frac{d}{dt}(\cdot)$. Equations (2.25), expressed in terms of these new variables thus becomes:

$$\begin{cases} \frac{d\vec{\tilde{r}}\left(\tilde{t}\right)}{d\tilde{t}} = \vec{\tilde{v}}_r\left(\tilde{t}\right) \\ \frac{d\vec{\tilde{v}}_r\left(\tilde{t}\right)}{d\tilde{t}} = \frac{1}{m}\frac{\partial\Phi}{\partial\vec{\tilde{r}}}\left(\vec{\tilde{r}}\right) \\ \vec{\tilde{r}}\left(\tilde{t}=0\right) = \vec{r}\left(t=t_e\right) = \vec{r}_0^* \qquad \vec{v}_r\left(\tilde{t}=0\right) = -\vec{v}_r\left(t=t_e\right) = -\vec{v}_{r,0}^* \end{cases} \tag{2.32}$$

where $\vec{r}_{r,0}^*$ and $\vec{v}_0^*$ denote the post-collision position and velocity (i.e. at $t = t_e$ when particle 2 leaves the interaction sphere) for Eqs. (2.25). By comparison with Eqs. (2.25) and using Eq. (2.26), the new equation system, Eqs. (2.32), admits the following trajectories

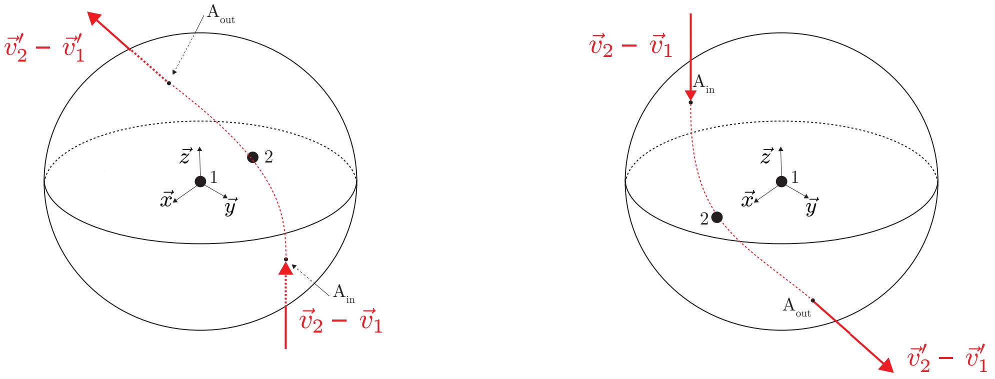
$$\vec{\tilde{r}}\left(t\right) = \vec{\Gamma}\left(\vec{r}_0^*, -\vec{v}_{r,0}^*, t\right) \tag{2.33}$$

parametrized by time t. This shows that $\vec{\Gamma}\left(\vec{r}_0^*, -\vec{v}_{r,0}^*, t\right)$, which is nothing more than the original trajectory $\vec{\Gamma}\left(\vec{r}_0, \vec{v}_{r,0}, t\right)$ in the reverse direction, is also a trajectory. This in turn means that any trajectory in the center of mass reference frame with initial position $\vec{x}_0$ and velocity $\vec{v}_{r,0}$ in the reverse direction is also a trajectory (with initial position $\vec{r}_0^*$ and velocity $-\vec{v}_{r,0}^*$). We now proceed to use the two symmetry properties of trajectories that were just derived to obtain the value of f_{12} at point A_{out} by using the fact that f_{12} is constant along trajectories.

We first consider trajectory (A), shown in Fig. 2.2(a) during which the relative velocity between particles is transformed from its pre-collision value $\vec{v}_r = \vec{v}_2 - \vec{v}_1$ to its post-collision value $\vec{v}_r' = \vec{v}_2' - \vec{v}_1'$[3]. By transforming it using a parity transformation following Eq. (2.28), another acceptable trajectory (shown in Fig. 2.2(b)) is obtained. The new trajectory is then transformed, using the time reversal transformation of Eq. (2.31) into the acceptable trajectory shown in Fig. 2.2(c). This final trajectory describes a collision in which the two particles have a final relative velocity of $\vec{v}_2 - \vec{v}_1$, which is the situation that occurs at point A_{out} in Fig. 2.1. The pre-collision relative

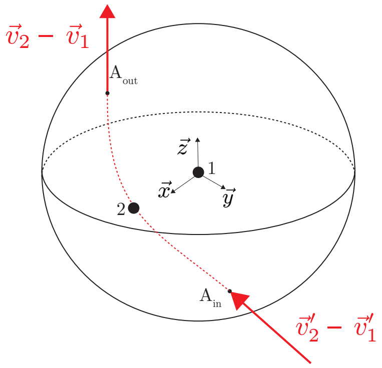
---

[3] Throughout this thesis $'$ denote post-collisional variables.



(a) Trajectory A: Original trajectory

(b) Trajectory B: Trajectory A after parity transformation



(c) Trajectory C: Trajectory B after time reversal

Figure 2.2: Effect of transformation P and T on trajectories

velocity of such a trajectory is $\vec{v}_r' = \vec{v}_2' - \vec{v}_1'$ which means that

$$\vec{v}_1|_{A_{\text{in}}} = \vec{v}_1' \quad \text{and} \quad \vec{v}_2|_{A_{\text{in}}} = \vec{v}_2'. \tag{2.34}$$

Within the framework of Fig 2.1, this means that the relative velocity at point A_{in} is $\vec{v}_r = \vec{v}_2' - \vec{v}_1'$. Because of the conservation of f_{12} along trajectories

$$f_{12}\left(\vec{v}_1|_{A_{\text{in}}}, \vec{v}_2|_{A_{\text{in}}}\right) = f_{12}\left(\vec{v}_1|_{A_{\text{out}}}, \vec{v}_2|_{A_{\text{out}}}\right) \tag{2.35}$$

The molecular chaos hypothesis (hypothesis G) is valid at point A_{in} as particles have not yet collided so that

$$f_{12}\left(\vec{v}_1|_{A_{\text{in}}}, \vec{v}_2|_{A_{\text{in}}}\right) = f_1\left(\vec{v}_1|_{A_{\text{in}}}\right)\ f_1\left(\vec{v}_2|_{A_{\text{in}}}\right) = f_1\left(\vec{v}\,'_1\right)\ f_1\left(\vec{v}\,'_2\right) \tag{2.36}$$

and so from Eqs. (2.34) and (2.36),

$$f_{12}|_{A_{\text{out}}} = f_1\left(\vec{v}\,'_1\right)\ f_1\left(\vec{v}\,'_2\right) \tag{2.37}$$

Returning to the second integral of Eq. (2.19), since point A_{out} belongs to Σ_B:

$$\iint\limits_{\mathbb{R}^3\ \Sigma_B} f_{12}\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 = \iint\limits_{\mathbb{R}^3\ \Sigma_B} f_1\left(\vec{x}, \vec{v}\,'_1\right)\ f_1\left(\vec{x}, \vec{v}\,'_2\right)\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 \tag{2.38}$$

where, because of hypothesis D, spatial variations between $\vec{x}_1$ and $\vec{x}_2$ are neglected. Using the results of Eqs. (2.21) and (2.38), Eq. (2.19) finally becomes

$$\begin{aligned}
\mathcal{B}\left(f_{12}\right) = &\iint\limits_{\mathbb{R}^3\ \Sigma} f_{12}\left(\vec{x}_1, \vec{x}_2, \vec{v}_1, \vec{v}_2\right)\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 \\
&= \iint\limits_{\mathbb{R}^3\ \Sigma_A} f_1\left(\vec{x}, \vec{v}_1\right) f_1\left(\vec{x}, \vec{v}_2\right)\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 + \\
&\quad \iint\limits_{\mathbb{R}^3\ \Sigma_B} f_1\left(\vec{x}, \vec{v}\,'_1\right)\ f_1\left(\vec{x}, \vec{v}\,'_2\right)\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2 \\
= &\iint\limits_{\mathbb{R}^3\ \Sigma_A} \left[f_1\left(\vec{x}, \vec{v}\,'_1\right) f_1\left(\vec{x}, \vec{v}\,'_2\right) - f_1\left(\vec{x}, \vec{v}_1\right) f_1\left(\vec{x}, \vec{v}_2\right)\right]\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}\ d\Sigma\ d\vec{v}_2
\end{aligned} \tag{2.39}$$

since $\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}|_{\Sigma_B} = -\left(\vec{v}_2 - \vec{v}_1\right)\cdot\vec{n}|_{\Sigma_A}$. Using the geometrical framework of Fig. 2.3, the collision term can be rewritten as:

$$\mathcal{B}\left(f_{12}\right) = \int\limits_{\mathbb{R}^3}\int\limits_{\mathbb{R}}\int\limits_{\mathbb{R}^+}\int\limits_0^{2\pi} \left\|\vec{v}_2 - \vec{v}_1\right\|\left[f'_1 f'_2 - f_1 f_2\right] b\ d\varepsilon\ db\ dz\ d\vec{v}_2. \tag{2.40}$$

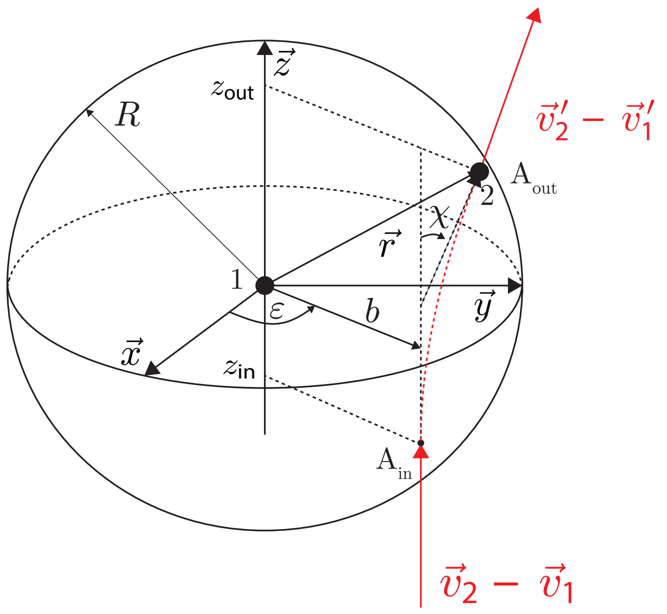


Figure 2.3: View of the interaction sphere during a collision.

b is often referred to as the impact parameter [15], it is the distance of closest approach if the two particles did not interact. In the following, the relative velocity $\vec{v}_2 - \vec{v}_1$, will be systematically designated by $\vec{v}_r$ and its norm by $v_r = \|\vec{v}_2 - \vec{v}_1\|$. The differential surface element $d\Sigma_0 \triangleq b\ db\ d\varepsilon$ can also equivalently be written in terms of solid angle $d\Omega$ [71], using the differential cross section σ:

$$d\Sigma_0 = b\ db\ d\varepsilon = \sigma\left(\chi, \varepsilon\right) d\Omega, \tag{2.41}$$

where χ designates the deflection angle. Using Eqs. (2.40) and (2.41), with $f \triangleq f_1$ and replacing $(\vec{v}_1, \vec{v}_2)$ by $(\vec{v}, \vec{v}_1)$, respectively, Eq. (2.16) can be rewritten as:

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} + \frac{\vec{F}}{m} \cdot \frac{\partial f}{\partial \vec{v}} = \iint\limits_{\mathbb{R}^3\ 4\pi} \sigma v_r \left(f\left(\vec{x}, \vec{v}_1^{\,\prime}, t\right) f\left(\vec{x}, \vec{v}^{\,\prime}, t\right) - f\left(\vec{x}, \vec{v}_1, t\right) f\left(\vec{x}, \vec{v}, t\right) \right) d\Omega\ d\vec{v}_1, \tag{2.42}$$

which is the Boltzmann equation for a one-species, chemically inert gas where individual particles are subjected to force $\vec{F}$ and undergo elastic collisions described by differential cross section σ. It was derived from the Liouville equation by using a number of assumptions which we restate here:

A) The density is assumed to be low enough so that triple body interactions can be neglected.

B) Forces between particles are assumed to have a short range of interaction and to be negligible for distances greater than R.

C) Inside the sphere of interaction, field effects are assumed to be negligible compared to interaction forces between particles.

D) The spatial variation of f_{12} over the length scale of the interaction is assumed to be small.

E) The temporal variation of f_{12} during the course of binary interactions is assumed to be small.

F) All collisions are microreversible.

G) There is no correlation between every two particles that participate in a collision.

For simplicity, in the rest of this chapter, Eq. (2.42) will be considered without external forces $\vec{F} = \vec{0}$, so that it can be rewritten

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} = Q\left(f, f\right) \tag{2.43}$$

where the collision operator Q is abbreviated as:

$$Q\left(f, f\right) = \iint\limits_{\mathbb{R}^3\ 4\pi} \sigma v_r \left(f' f_1' - f f_1\right)\ d\Omega\ d\vec{v}_1 = \iint\limits_{\mathbb{R}^3\ 4\pi} B\left(f' f_1' - f f_1\right)\ d\Omega\ d\vec{v}_1 \tag{2.44}$$

where

$$B\left(v_r, \chi\right) \triangleq \sigma\left(v_r, \chi\right) v_r \tag{2.45}$$

is called the Boltzmann collision kernel. Because of the microreversibility hypothesis that was used in the derivation of the Boltzmann equation, the probability of the collision $(\vec{v}, \vec{v}_1) \rightarrow (\vec{v}', \vec{v}_1')$ has to be identical to that consisting of $(\vec{v}', \vec{v}_1') \rightarrow (\vec{v}, \vec{v}_1)$ (which was shown by using symmetry arguments), so that the cross section (and thus the collision kernel) can only depend on v_r and the deflection angle χ. The effects of the previous assumptions on the form of the collision operator can clearly be seen upon closer inspection. The fact that Q is quadratic in f is a consequence of A), while the fact that variables $\vec{x}$ and t only appear as parameters (not, for example, in a differential operator) is a consequence of D) and E), respectively, which correspond to the fact that collisions are localized in time and space. The appearance of the products $f'f_1'$ and ff_1 is a consequence of G) while F) is reflected in the particular form of the collision kernel (only a function of v_r and χ) and in the expression of $(\vec{v}', \vec{v}_1')$ as a function of $(\vec{v}, \vec{v}_1)$ which will be discussed in the next section. Further details about the properties of the collision operator and the Boltzmann equation, more generally, can be found in [38], [39] or [146].

2.1.2 Binary elastic collisions

The Boltzmann equation, Eq. (2.43), requires some knowledge about the collision dynamics between particles through the cross section σ that appears in the collision term. The differential cross section σ can take a variety of forms depending on the variable of integration chosen for the collision term. When the internal energy of particles is not considered, the most general expression for the cross section is $\sigma\left(\vec{v}_r, \vec{v}_r'\right)$, where $\vec{v}_r'$ designates the post-collision relative velocity between the two particles. By definition, the post-collision distribution function of $\vec{v}'_r$ will be given by

$$f_{\text{post,coll}}\left(\vec{v}'_r|\vec{v}_r\right) = \frac{\sigma\left(\vec{v}_r, \vec{v}'_r\right)}{\int_{\mathbb{R}^3} \sigma\left(\vec{v}_r, \vec{v}'_r\right) d\vec{v}'_r}, \tag{2.46}$$

Following the arguments presented in Chapter 3 of [87], the single particle collision rate is given by

$$k_{\text{coll}}\left(\vec{v}_r\right) = \|\vec{v}_r\| \int\limits_{\mathbb{R}^3} \sigma\left(\vec{v}_r, \vec{v}'_r\right) d\vec{v}'_r, \tag{2.47}$$

so that the probability of a collision can readily be seen to scale as follows:

$$P_{\text{coll}}\left(\vec{v}_r\right) \propto \|\vec{v}_r\| \int\limits_{\mathbb{R}^3} \sigma\left(\vec{v}_r, \vec{v}'_r\right) d\vec{v}'_r. \tag{2.48}$$

Because of the requirement that collisions be microreversible that was previously mentioned, the collision cross section can, however, only be a function of v_r and χ, i.e. only 2 independent variables, instead of the 6 of $\vec{v}_r$ and $\vec{v}'_r$. The integral in the collision operator of Eq. (2.44), can thus be viewed as an integral over all possible collisions between the two particles when one of the particles has velocity $\vec{v}$ and the other $\vec{v}_1$. In the following, we will be using an alternate representation for the collision term, called the σ representation [146], which is more natural to consider when performing particle simulations. As each collision preserves momentum and energy:

$$\begin{cases} \vec{v} + \vec{v}_1 = \vec{v}' + \vec{v}' \\ \|\vec{v}\|^2 + \|\vec{v}_1\|^2 = \|\vec{v}'\|^2 + \|\vec{v}'_1\|^2 \end{cases} \tag{2.49}$$

which provides 3+1=4 constraints for the post collision velocities $\vec{v}'$ and $\vec{v}'$, which have a total of 6 degrees of freedom (as each belongs to $\mathbb{R}^3$). This means that all possible post-collision velocities can be parametrized by 2 degrees of freedom. In the σ representation those two degrees of freedom are provided by vector $\vec{e}$ which belongs to the unit sphere $\mathbb{S}^2$. Using $\vec{e}$, the post-collision velocities can be rewritten as:

$$\begin{cases} \vec{v}' = \frac{\vec{v}+\vec{v}_1}{2} + \frac{\|\vec{v}-\vec{v}_1\|}{2}\vec{e} \\ \vec{v}_1' = \frac{\vec{v}+\vec{v}_1}{2} - \frac{\|\vec{v}-\vec{v}_1\|}{2}\vec{e} \end{cases} \tag{2.50}$$

so that the collision term of Eq. (2.44) can be rewritten as:

$$Q\left(f,f\right) = \iint\limits_{\mathbb{S}^2\ \mathbb{R}^3} \sigma\left(v_r,\vec{e}\right) v_r \left(f'f_1' - ff_1\right)\ d\vec{v}_1\ d\vec{e}. \tag{2.51}$$

noting the fact that, as a consequence of the microreversibility hypothesis, B and σ can only depend on v_r and $\frac{\vec{v}_r}{\|\vec{v}_r\|}\cdot\vec{e}$. We remark that in the previous expression of Eq. (2.44), the two degrees of freedom were provided by angles χ and ε which both combine to form the solid angle $d\Omega = sin\chi\ d\chi\ d\varepsilon$. As the post-collision velocity is parametrized by $\vec{e}$ instead of $\vec{v}_r'$, its distribution function, by analogy with Eq. (2.46) is given by:

$$f_{\text{post,coll}}\left(\vec{e}\,|v_r\right) = \frac{\sigma\left(v_r,\vec{e}\right)}{\int_{\mathbb{S}^2}\sigma\left(v_r,\vec{e}\right)d\vec{e}}. \tag{2.52}$$

In the following, we will make use of the total cross section which is defined as

$$\sigma_T\left(v_r\right) \triangleq \int\limits_{\mathbb{S}^2} \sigma\left(v_r,\vec{e}\right)\ d\vec{e} = \int\limits_0^{2\pi}\int\limits_0^{\pi/2} \sigma\left(v_r,\chi\right)\ sin\chi\ d\chi\ d\varepsilon \tag{2.53}$$

and the viscosity cross section, σ_μ, defined by

$$\sigma_\mu\left(v_r\right) \triangleq \int\limits_0^{2\pi}\int\limits_0^{\pi/2} \sigma\left(v_r,\chi\right)\ sin^3\chi\ d\chi\ d\varepsilon. \tag{2.54}$$

2.2 The direct simulation Monte Carlo technique (DSMC)

2.2.1 Numerical methods for the transition regime

Before turning to numerical methods for the Boltzmann equation, we briefly discuss some alternative fluid models, summarized in Fig. 2.4, that can be used to avoid having to solve the Boltzmann equation in the transition regime. Although not directly based on the Boltzmann equation, they are nonetheless able to capture non-equilibrium effects at low Knudsen numbers. Currently, the two most popular such methods are those based on the Burnett equation or on moments of the Boltzmann equation. The Burnett equations are derived from a Chapman-Enskog expansion [40] of the Boltzmann equation in terms of the Knudsen number Kn. To zeroth order, the Euler equations are obtained, to first order the Navier-Stokes equations, to second order and third order the Burnett and super-Burnett equations, respectively. Unfortunately, these last equations are particularly difficult to solve because they are linearly unstable so that additional stabilizing terms have to be included to stabilize them [1]. Moments equations are based on the solution of transport equations for the moments of the Boltzmann equations. The most widely used model, Grad's *13 moment equations*, consists of a set of transport equations for $n, \vec{u}, \vec{p}, \overline{\overline{\tau}}, \vec{q}$ (for a total of $(1+3+1+5+3 = 13)$ independent unknowns) which are obtained [136] by expanding the distribution function of the Boltzmann equation in terms of Hermite polynomials. In its original form, the 13 moment equation system is hyperbolic and cannot account for the transport of the heat flux or stress tensor [72], which explains its relatively poor performance. A regularized 13 moment equation system has more recently been proposed [137] which has shown good agreement with DSMC for Couette flow [75]. Burnett and moment based equations are, however, seldom used today because of the challenges that were described. They can only represent a viable alternative to DSMC in the lower Kn part of the transition regime (where DSMC is very computationaly expensive) if these are mitigated.

2.2.2 Numerical methods for the Boltzmann equation

The Boltzmann equation, Eq. (2.43), is particularly difficult to solve because of the collision term which is local with respect to $\vec{x}$ but non-local with respect to $\vec{v}$ because of the integral over all $\vec{v}_1$ velocities. The most natural approach, due to the fact that the Boltzmann equation is a partial differential equation, is to attempt to discretize both physical space and velocity space with a grid and to apply any of the methods traditionally used to solve partial differential equations [123]. This approach is often referred to as a *deterministic* simulation approach, in opposition to Monte Carlo based simulation techniques, that will be later introduced. A fairly complete overview of the subject can be found in [6]. Most deterministic simulation methods are based on the discrete velocity method [109] and either use the original collision term or an approximation of it such as the BGK model [3]. Those methods are, however, very expensive due to the need to discretize velocity space (particularly for three dimensional simulations). The problem is exacerbated for flows containing particles with very disparate velocities such as high speed reentry flows. In such flows, both very high speed particles (in the free steam) and low speed particles (behind the shock wave) are present such that adequate resolution in physical space can only be accomplished by employing many grid points. This problem can, however, be somewhat alleviated by using an adaptive grid in velocity space [8]. Deterministic methods have increased in popularity in recent years, both as a result of ever increasing computational resources, and as a response to the complete dominance of Monte-Carlo based simulation techniques for the Boltzmann equation. They are, however, attractive for very low speed flows, which are very difficult to simulate via Monte-Carlo methods due to their inherent noise and are easier to couple to continuum solvers than DSMC, which makes them attractive for hybrid fluid/kinetic simulation approaches [95].

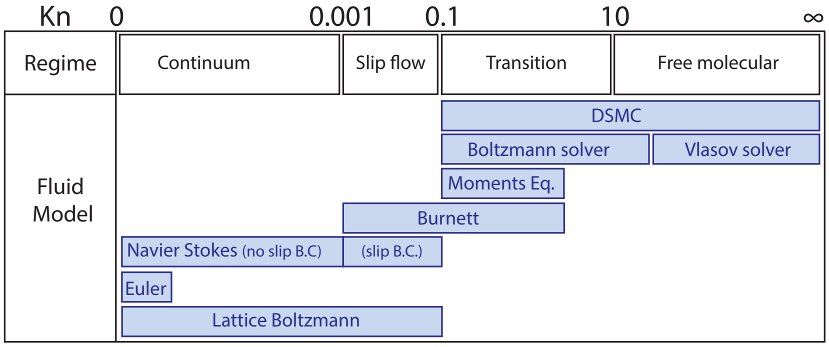


Figure 2.4: Flow Regimes and commonly used fluid models.

The most popular simulation techniques for the Boltzmann equation are those belonging to the class of Monte Carlo methods. They are based on the use of computational particles (also called macroparticles) that each represent a large number of physical particles. The former are in turn moved throughout the simulation domain and allowed to collide. The system of particles is allowed to evolve through time and flow quantities are sampled. The sampled flow quantities are in turn observed to converge to those obtainable by taking moments of the exact velocity distribution function solution of the Boltzmann equation, if it were available. The most popular Monte Carlo simulation method for the Boltzmann equation, today, is the DSMC method. It was originally introduced by Bird in 1963 [12]. It clearly draws its inspiration from the molecular dynamics method that was introduced by Alder in 1957 [2]. The advent of the DSMC method in the early 1960s is really due to the advent of digital computers as the basic principle of the method is fairly obvious. A detailed account of the evolution of DSMC through the years was written by Bird [17] which also recounts the challenges of developing such a method on the computers of that time. DSMC is not the only Monte Carlo simulation method for the Boltzmann equation, others include Nanbu's method [111] which is, however, seldom used today. The DSMC method will be presented in the following within the framework of the simulation of a single species gas governed by Eq. (2.43), i.e., a gas which particles only undergo binary elastic collisions in the absence of any external forces. To not overwhelm the reader, it will furthermore be presented in this chapter in its simplest formulation, that is, for a spatially constant time step Δt and particle weight W_p. The various modifications that have to be brought to the basic method presented below, e.g. in the presence of multiple species with different relative weights or with an electric field, are detailed as needed in each subsequent chapter of this thesis.

2.2.3 Overview of DSMC

The DSMC method aims to solve the Boltzmann Equation, i.e. Eq. (2.43) :

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} = Q\left(f, f\right). \tag{2.55}$$

over the physical domain $\Omega \in \mathbb{R}^3$, for $t > 0$ with suitable boundary conditions on $\partial\Omega$ and given initial conditions $f\left(\vec{x}, \vec{v}, t = 0\right) = f_0\left(\vec{x}, \vec{v}\right)$. We are not concerned in this thesis with the existence or uniqueness of such solutions (See [146] for an overview of the subject). The basic idea of the method is to discretize physical space Ω using N_c cells Ω_i, such that $\cup_{i=1}^{N_c}\Omega_i = \Omega$ and to perform a splitting of Eq. (2.55) by either only considering its advection or collision term. A general outline of the DSMC procedure is shown in pseudo-code 2.1 which will be discussed in the following. More specifically, it consists in sequentially solving the following homogeneous Boltzmann equation for each individual Ω_i cell of the domain

$$\frac{\partial f}{\partial t}\left(\vec{x}, \vec{v}, t\right) = \iint\limits_{\mathbb{R}^3\ \mathbb{S}^2} \sigma v_r \left[f\left(\vec{x}, \vec{v}_1^{\,\prime}, t\right) f\left(\vec{x}, \vec{v}^{\,\prime}, t\right) - f\left(\vec{x}, \vec{v}_1, t\right) f\left(\vec{x}, \vec{v}, t\right)\right]\, d\vec{e}\, d\vec{v}_1. \tag{2.56}$$

and the collisionless Boltzmann (i.e. Vlasov) equation for all cells of the domain:

$$\frac{\partial f}{\partial t}\left(\vec{x}, \vec{v}, t\right) + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} = 0 \tag{2.57}$$

This splitting is accomplished by populating the domain with computational particles that each represent W_p actual particles. They are injected at the open boundaries of $\partial\Omega$ at step iii) of pseudo code 2.1 according to known velocity distribution functions (which are part of the specified boundary conditions). All j particles are moved inside the domain at each time step Δt according to their velocities following:

$$\vec{x}_j\left(t+\Delta t\right) = \vec{x}_j\left(t\right) + \int\limits_{t}^{t+\Delta t} \vec{v}_j\left(s\right)\ ds. \tag{2.58}$$

Eq. (2.58) aims to solve the collisionless Boltzmann equation, i.e. Eq. (2.57). In keeping with the idea of separating the collision of particles from their ballistic motion, the velocity of particles is assumed constant during their movement and is only altered during the collision phase. Using a first order Euler explicit scheme, this allows the following temporal discretization of Eq. (2.58):

$$\vec{x}_j^{k+1} = \vec{x}_j^{k} + \Delta t\ \vec{v}_j^{k}, \qquad \vec{v}_j^{k+1} = \vec{v}_j^{k}\ . \tag{2.59}$$

This movement of particles occurs at step ii) of pseudo code 2.1. While the use of higher order schemes with respect to time is possible, they are not commonly used for DSMC, due to the very small time step values usually employed. This basic particle movement scheme must, however, be altered for axisymmetric geometries (discussed in [15] p. 371) and in the presence of an external force field which alters the velocity of particles between collisions (presented in Chapter VI). At each time step, either before or after the move procedure of Eq. (2.59), and in each cell, particles are allowed to collide between one another according to the DSMC collision process. This part of the algorithm aims to “solve” the homogeneous Boltzmann equation, i.e. Eq. (2.56) in all cells of the domain and occurs at step i) of pseudo code 2.1.

Algorithm 2.1: Pseudo code of a DSMC procedure

```
For time steps, $k = 0, ..., N_{\text{step}}$, do:
    For all cells, $i = 1, ..., N_c$, do:
        i) PerformCollisions()
        ii) MoveParticles
        iii) ApplyBCs()
    endFor all cells
endFor all time step
```

Although, a rigorous derivation of the DSMC collision procedure, such as will be presented in Section 2.3, is fairly involved, its basic premise is to exactly mimic the collisional behavior of the physical gas by suitably rescaling the number of collisions that each particle has to undergo by it weighting factor, W_p. Following the arguments presented in Chapt. 3 of [87], the single particle collision frequency[4] in the physical gas is given by

$$\nu^{(1)} = n\langle \sigma_T v_r \rangle \tag{2.60}$$

where the averaging operator $\langle \cdot \rangle$ is defined as

$$\langle \sigma_T \ v_r \rangle = \int_{\mathbb{R}^3} \sigma_T \left(v_r \right) v_r \ dv_r, \tag{2.61}$$

where the total cross section σ_T is defined by Eq. (2.53). As each computational particle represents W_p actual physical particles, it is natural that the single particle collision frequency $\tilde{\nu}^{(1)}$ in the computational gas be given by:

$$\tilde{\nu}^{(1)} = \frac{n}{W_p} \langle \sigma_T v_r \rangle, \tag{2.62}$$

---

[4] the number of collisions that each particle in a gas with number density n undergoes per unit time.

while an intuitive measure of the probability of a collision for each particle in a cell of volume V during Δt is given ([15] p. 8) by

$$P_{coll} = \frac{\langle \sigma_T v_r \rangle \Delta t}{V} \tag{2.63}$$

As various versions of the DSMC collision process exist, to keep things simple, we will present in this introductory section about DSMC, the version known as the "no time counter scheme" [15] which is the one used throughout this thesis. A few other collision pair selection schemes will, however, be later discussed in Section 2.4.2 to motivate the use of the present scheme. They will all be shown to be particular variants of a general Markov process, which will be discussed in Section 2.3. As the no time counter scheme is essentially a computational procedure, it is presented in the form of a pseudo-code in Algorithm 2.2.

Algorithm 2.2: Pseudo code of the no time counter collision scheme at time t^k

```
for all cells, $i = 1, ..., N_c$, do:
    I) Index all $N_{p,i}$ particles in cell $i$ between $j = 1$ and $j = N_{p,i}$.
    II) Calculate the number of collisions that occur during time $\Delta t$ following:
        $$N_{\text{coll},i}^k = \frac{\Delta t W_p}{2\Omega_i} N_i^k \left(N_i^k - 1\right) \left(\|\vec{v}_r\|\sigma\right)_{i,\max} \qquad (2.64)$$
    III) Randomly select $N_{\text{coll},i}^k$ pairs of molecules for collisions
    IV) Test potential collisions
        For all potential collisions, $l = 1, ..., N_{\text{coll},i}^k$, do:
            i) Randomly select particles $l_1 \sim \mathcal{U}\{1, N_{p,i}\}$ and $l_2 \sim \mathcal{U}\{1, N_{p,i}\}$
            ii) Calculate probability of occurrence of collision
                $$P_{\text{coll}} = \frac{\int_{\mathbb{S}^2} \|\vec{v}_{l_1} - \vec{v}_{l_2}\| \sigma \left(\|\vec{v}_{l_1} - \vec{v}_{l_2}\|, \vec{e}\right) d\vec{e}}{\left(\|\vec{v}_r\|\sigma\right)_{i,\max}} \qquad (2.65)$$
            iii) if(rand() $< P_{\text{coll}}$) collision is fictitious goto (i)
                else
                Collide particles $l_1$ and $l_2$
                1) Generate $\vec{e} \sim f_{\text{post,coll}}\left(\vec{e}\,|v_r\right) = \frac{\sigma(v_r,\vec{e})}{\int_{\mathbb{S}^2} \sigma(v_r,\vec{e}) d\vec{e}}$.
                2) Update velocities
                    $$\begin{aligned} \vec{v}'_{l_1} &= \frac{\vec{v}_{l_1} + \vec{v}_{l_2}}{2} + \frac{\|\vec{v}_{l_1} - \vec{v}_{l_2}\|}{2} \vec{e} \\ \vec{v}'_{l_2} &= \frac{\vec{v}_{l_1} + \vec{v}_{l_2}}{2} - \frac{\|\vec{v}_{l_1} - \vec{v}_{l_2}\|}{2} \vec{e} \end{aligned}$$
                3) If $P_{\text{coll}} > 1$
                    $$\left(\|\vec{v}_r\|\sigma\right)_{i,\max} = \int_{\mathbb{S}^2} \|\vec{v}_{l_1} - \vec{v}_{l_2}\| \, \sigma\left(\vec{v}_{l_1}, \vec{v}_{l_2}, \vec{e}\right) \, d\vec{e} \qquad (2.66)$$
                endif (rand() $< P_{\text{coll}}$)
        endfor all potential collisions
endfor all cells
```

The no time counter (NTC) method calculates at each time step, in a deterministic way, following Eq. (2.64), the number of collisions that have to occur in each cell of the domain, where N_i^k designates the number of particles in cell i at time step t^k, W_p is the particle weight (i.e. the number of physical particles that each actual particle represents) and Ω_i is the volume of cell i. $\left(\|\vec{v}_r\|\sigma\right)_{i,\max}$ designates the maximum value of the product of the relative velocity and the cross section amongst all velocity pairs in the cell during all time steps of the simulation. This quantity is recorded throughout all the simulation and suitably updated following Eq. (2.66). The *collision model* provides a value for the cross section $\sigma\left(v_r, \vec{e}\right)$ and is described following the present discussion of the collision pair selection scheme. Having determined the number of particles pairs to consider for collision, the NTC scheme proceeds to form the required number of particles by randomly selecting particles from the cell in step IV)i). They are in turn tested following Eq. (2.65), to determine if they give rise to a fictitious or real collision which does satisfy the basic scaling of Eq. (2.48). The choice of the proportionality constant $\frac{1}{\left(\|\vec{v}_r\|\sigma\right)_{i,\max}}$ in Eq. (2.65) is a consequence of the number of pairs selected for collision through Eq. (2.64) and the need to reproduce the correct single particle collision rate of Eq. (2.62). A fictitious collision is a collision where the velocity of the two colliding particles are left unaltered. This issue is further discussed in Section 2.4.2.2. In the case where a collision is slated to occur, a new scattering direction for the relative velocity $\vec{e}$ is chosen at step IV)iii)1) following the distribution function for $\vec{e}$ given by Eq. (2.52). The post-collision velocities $\vec{v}_{l_1}{}'$ and $\vec{v}_{l_2}{}'$ of the two particles are in turn calculated following Eqs. (2.50) at step IV)iii)2).

Independently of the pair selection scheme, the *collision model* provides a value for the differential cross section $\sigma\left(\|\vec{v}_{l_1} - \vec{v}_{l_2}\|, \vec{e}\right)$. The “variable hard sphere” (VHS) collision model is used throughout this thesis, first introduced in [14]. It assumes that the differential cross section σ is of the following form:

$$\sigma\left(\vec{v}_{l_1}, \vec{v}_{l_2}, \vec{e}\right) = \frac{C_{\text{VHS}}}{\|\vec{v}_{l_1} - \vec{v}_{l_2}\|^{\omega_{\text{VHS}}}}\ , \quad \omega_{\text{VHS}} \geq 0 \tag{2.67}$$

where constants C_{VHS} and the exponent ω_{VHS} are chosen so as to reproduce both a specified viscosity μ_{ref} when the gas is in equilibrium at temperature T_{ref} and a power law for viscosity, i.e. $\mu\left(T\right) = \mu_{\text{ref}}\left(\frac{T}{T_{\text{ref}}}\right)^{\omega_\mu}$, when the gas is in equilibrium at temperature T. The values of C_{VHS} and ω_{VHS} as a function of T_{ref}, μ_{ref} and ω_μ[5] can be obtained analytically [15] by using the definition of viscosity[6] and the expression for the distribution[7] of relative velocity $\vec{v}_r = \vec{v}_{l_1} - \vec{v}_{l_2 \neq l_1}$ in an equilibrium gas:

$$C_{\text{VHS}} = \frac{15}{64\sqrt{\pi}}\frac{m}{\mu_{\text{ref}}}\left(\frac{4k_B T_{\text{ref}}}{m}\right)^{\omega_\mu} \tag{2.68}$$

and

$$\omega_{\text{VHS}} = 2\ \omega_\mu - 1 \tag{2.69}$$

The differential cross section of Eq. (2.67) is independent of $\vec{e}$. This means that, whenever this model is used, the direction vector $\vec{e}$ at step IV)iii)1) of Algorithm 2.2 should be generated following

$$f_{\text{post,coll}}\left(\vec{e}\ |v_r\right) = \frac{\sigma\left(v_r, \vec{e}\right)}{\int_{\mathbb{S}^2}\sigma\left(v_r, \vec{e}\right) d\vec{e}} = \frac{1}{4\pi}$$

which is the uniform distribution on the unit sphere $\mathbb{S}^2$.

Having presented the basic formulation of the DSMC procedure using the collision-pair selection scheme ("no-time counter" scheme) and the collision model ("variable

---

[5] The following was employed throughout the thesis: for argon, [40] $\mu_{\text{ref}} = 2.117 \times 10^{-5}$ [Pas] at $T_{\text{ref}} = 273\text{K}$ and $\omega_\mu = 0.81$.

[6] The viscosity of a gas in equilibrium can be shown to be given by $\mu = \frac{5/8\ (\pi m k_B T)^{1/2}}{\left(\frac{m}{4k_B T}\right)^4 \int_0^{+\infty} v_r^7 \sigma_\mu(v_r) \exp(-m v_r^2/4k_B T) dv_r}$ p. 404 of [147], where the viscosity cross section σ_μ is defined by Eq. (2.54).

[7] $f\left(v_r = \|\vec{v}_r\|\right) = \frac{\sqrt{2}\ m^{3/2} v_r^2}{\pi^{1/2}(2k_B T)^{3/2}} \exp\left(-m v_r^2/\left(4 k_B T\right)\right)$ [15].

hard sphere" model) that are used throughout the thesis, we now present DSMC from a more general and mathematical viewpoint.

2.3 DSMC as a Markov process

2.3.1 Motivation

The no time counter scheme that was previously presented actually corresponds the implementation of a much more general Markov process. Viewing DSMC as a Markov process has many advantages as it allows *i)* a rigorous presentation and greater understanding of the DSMC procedure, *ii)* the development of new DSMC schemes and *iii)* a convergence proof. In this section, the Markov process corresponding to the DSMC procedure will be used to formulate a general DSMC procedure that is independent of the collision pair selection or collision scheme. A derivation of the no time counter DSMC procedure (introduced in the previous section, and used throughout this work) will be presented and it will be contrasted with the less efficient but more natural "direct simulation scheme". The proof of [148] will only be very briefly discussed at the end.

The DSMC procedure was first developed [12] largely based on physical reasoning by using fairly simple probability concepts. Most of the subsequent improvements brought to the method, such as, for instance, new collision pair selection schemes [14] or species weighting [13], were all based on mostly physical arguments or as the result of numerical experimentation. In parallel to those developments in the West (including Japan), a number of stochastic methods based on stochastic differential equations were introduced in the Soviet Union such as e.g. [7] (See [132] for a good overview) that were similar to Bird's but much more rigorously derived from the Boltzmann equation and with a convergence proof in the case of [133]. Wagner's convergence proof [148] for the "time counter" scheme is partially based on the latter.

That scheme has been mostly superseded by the no time counter scheme which is used in this work and that was introduced in the previous section. A formal definition of the no time counter scheme in terms of a Markov process is given in [125], while the convergence proof is almost identical to that presented in [148]. The presentation and notation employed follow in large part from that introduced in [125] and [126].

2.3.2 A Boltzmann equation for collisions

By decoupling the movement of particles from their collision, the DSMC method de facto assumes that the particles inside each cell cannot be distinguished by their positions during the collision process. This means that the DSMC procedure should thus aim to locally approximate the homogeneous Boltzmann equation in each cell of the domain. We rewrite Eq. (2.56) as follows:

$$\begin{cases} \frac{\partial f}{\partial t}\left(\vec{v},t\right) = \iint\limits_{\mathbb{R}^3\ \mathbb{S}^2} \sigma v_r \left[f\left(\vec{v}_1',t\right) f\left(\vec{v}',t\right) - f\left(\vec{v}_1,t\right) f\left(\vec{v},t\right)\right]\, d\vec{e}\, d\vec{v}_1. \\ f\left(\vec{v},0\right) = f^0\left(\vec{v}\right) \end{cases} \tag{2.70}$$

and define the normalized distribution $\hat{f}$ as:

$$\hat{f}\left(\vec{v},t\right) = \frac{f\left(\vec{v},t\right)}{\int\limits_{\mathbb{R}^3} f\left(\vec{v},t\right) d\vec{v}} \tag{2.71}$$

A discussion of the conditions that the initial solution $f^0\left(\vec{v}\right)$ must satisfy for the existence of a solution can be found, amongst others in [51] [73].

2.3.3 Collisions as Markov jump process

We first start by considering a generalized version of the DSMC collision simulation procedure that was presented in the previous section for the particular case of the no time counter scheme. This general scheme can be used to obtain all current variants of the DSMC procedure and could even be used to define it.

Algorithm 2.3: Pseudo code of a general DSMC collision scheme at time t^k

```
For all cells, $i = 1, ..., N_c$, do:
    I) Index all $N_{p,i}$ particles in cell $i$ between $j = 1$ and $j = N_{p,i}$.
    II) Reset time counter $t_{c,i} = 0$
    III) While $t_{c,i} \leq \Delta t$
        i) Calculate time to next collision $\Delta t_{\text{coll}}$
        ii) Increment the counter $t_{c,i} = t_{c,i} + \Delta t_{\text{coll}}$
        iii) Select indices $l_1, l_2$ of two colliding particles from $f_{\text{sel, coll}}(l_1, l_2)$
        iv) Calculate collision probability $P_{\text{coll}}$
        v) if(rand() < $P_{\text{coll}}$) collision is fictitious goto (i)
           else
           Collide particles $l_1$ and $l_2$:
           1) Generate $\vec{e} \sim f_{\text{post,coll}}(\vec{e}\,|v_r) = \frac{\sigma(v_r,\vec{e})}{\int_{\mathbb{S}^2} \sigma(v_r,\vec{e}) d\vec{e}}$.
           2) Update velocities
              $\vec{v}'_{l_1} = \frac{\vec{v}_{l_1} + \vec{v}_{l_2}}{2} + \frac{\|\vec{v}_{l_1} - \vec{v}_{l_2}\|}{2} \vec{e}$
              $\vec{v}'_{l_2} = \frac{\vec{v}_{l_1} + \vec{v}_{l_2}}{2} - \frac{\|\vec{v}_{l_1} - \vec{v}_{l_2}\|}{2} \vec{e}$
           endif (rand() < $P_{\text{coll}}$)
    EndWhile $t_{c,i} \leq \Delta t$
EndFor all cells
```

An obvious difference between the general procedure and the no time counter scheme that was previously presented is the use of a time counter to determine the number of collisions to perform. The basic physical idea behind the use of such a procedure is to reproduce the correct number of collisions during time step Δt by incrementing the value of a time counter by an estimate Δt_{coll} of the mean collision time (average time between collision) of the cell every time a collision, real or fictitious, is performed.

Such an estimate furthermore need not be deterministic and can be obtained by sampling. We will later show that an implicit time counter is actually present in the no time counter scheme so than it can indeed be rewritten under the form of the general DSMC collision scheme given in algorithm 2.3. A DSMC collision model is needed to obtain an expression for the time counter increment Δt_{coll}, the distribution function of the particles to consider for collision $f_{\text{sel, coll}}\left(l_1, l_2\right)$ and the collision probability P_{coll}. The algorithm described by Algorithm 2.3 can be interpreted as a Markov process.

A Markov process is a random process in which the future is independent of the past, given the present. A random process is characterized at time t by a random variable $\overline{v}\left(t\right)$, which is called the state of the system. A Markov process has the particular property of memorylessness, i.e. that the value of $\left\{\overline{v}\left(t\right), t > t_0\right\}$ only depends of $\overline{v}\left(t_0\right)$, not on $\left\{\overline{v}\left(t\right), t < t_0\right\}$. A more formal definition and a few important properties of such processes, that will later be used, are described in Appendix A. The Markov process that we consider is characterized by the velocity of all particles in cell Ω_i which thus constitute its state variable $\overline{v}\left(t\right)$:

$$\overline{v}\left(t\right) = \left(\vec{v}_1\left(t\right), \vec{v}_2\left(t\right), ..., , \vec{v}_{N_i}\left(t\right)\right) \tag{2.72}$$

which belongs to state space $V = \left(\mathbb{R}^3\right)^{N_i}$. The system changes state every time a collision occurs, at which point it undergoes an instantaneous transition, called a *jump transformation*, and denoted by J, from $\overline{v}$ to its next state $\overline{v}'$. The system remains in the same state between transitions. Transitions occur at random times $T^0, T^1, ..., T^k, T^{k+1}, ...$ etc. and the time between transitions, i.e. $\Delta T^k \triangleq T^{k+1} - T^k$, follows an exponential distribution because of the Markov property, as shown in Appendix A. The parameter[8] λ of the exponential distribution is independent of the

---

[8] An exponential distribution has probability distribution function $f\left(x\right) = \lambda \exp\left(-\lambda x\right)$ which is fully characterized by parameter λ.

time step of the simulation Δt and is related to the mean collision time in the cell. Its value will be derived in Section 2.4.1. When two particles with respective indices l_1 and l_2 collide at time $t = T^k$, the state variable is changed from

$$\overline{v}\left(t\right) = \left(\vec{v}_1, \vec{v}_2, ..., \vec{v}_{l_1}, ..., \vec{v}_{l_2}, ..., \vec{v}_{N_i}\right), \ t < T^k$$

to

$$\overline{v}' = \overline{v}\left(t\right) = \left(\vec{v}_1, \vec{v}_2, ..., \vec{v}'_{l_1}, ..., \vec{v}'_{l_2}, ..., \vec{v}_{N_i}\right), \ t \geq T^k$$

where the post-collision velocities $\vec{v}'_{l_1}$ and $\vec{v}'_{l_2}$ are obtained by Eqs. (2.50) which we restate below:

$$\begin{cases} \vec{v}'_{l_1} = \frac{\vec{v}_{l_1} + \vec{v}_{l_2}}{2} + \frac{\|\vec{v}_{l_2} - \vec{v}_{l_1}\|}{2}\vec{e} \\ \vec{v}'_{l_2} = \frac{\vec{v}_{l_1} + \vec{v}_{l_2}}{2} - \frac{\|\vec{v}_{l_2} - \vec{v}_{l_1}\|}{2}\vec{e} \end{cases}$$

while the position $\vec{x}$ of particles is never altered by collisions. Using the jump transformation $J : V \times \mathbb{N} \times \mathbb{N} \times \mathbb{S}^2 \to V$, such a transition can be formalized as:

$$\overline{v}\,' \triangleq \overline{v}\left(t \geq T^k\right) = J\left(\overline{v}, l_1, l_2, \vec{e}\right). \tag{2.73}$$

where the arguments of J are, respectively, the old state variable $\overline{v} \triangleq \overline{v}\left(t < T^k\right)$, the indices of the two colliding particles l_1 and l_2 and the post-collision relative velocity direction $\vec{e}$ that is obtained from the collision model. In the following, the present state of the system will often be referred to as $\overline{v}$, and $\overline{v}'$ will be used to designate the state after the transition.

2.3.4 A Boltzmann master equation

Equation (2.70), the homogeneous Boltzmann equation, despite being simpler than the full Boltzmann equation, is still very challenging to solve numerically, particularly because of the non-linearity of f in the collision term. For that reason, the

DSMC method seeks to solve an equivalent equation, a particular form of the Kac master equation, based on the fact that N computational particles are present in the cell. This equation aims to mimic the Boltzmann equation while being in a simpler form. To this end, the normalized[9] probability density function F^{N_i} of the N_i particles in cell i is considered, i.e. $F^{N_i}\left(\vec{v}_1, ..., \vec{v}_{N_i}, t\right) = F^{N_i}\left(\overline{v}, t\right)$ using the definition of Eq. (2.72). By definition, $F^{N_i}\left(\vec{w}_1, \vec{w}_2, ..., \vec{w}_{N_i}, t\right) d\vec{w}_1\ d\vec{w}_2\ ...\ d\vec{w}_{N_i}$ is the simultaneous probability (or equivalently the fraction of total particles), at time t, that the velocity of particle 1, $\vec{v}_1$ is inside volume $d\vec{w}_1$ about $\vec{w}_1$ of velocity space, $\vec{v}_2$ is inside volume $d\vec{w}_2$ about $\vec{w}_2$ etc. The master equation is given by

$$\begin{cases} \frac{\partial F^{N_i}(\overline{v},t)}{\partial t} = \frac{W_p}{2\Omega_i} \sum\limits_{1 \leq a \neq b \leq N_i} \int\limits_{\mathbb{S}^2} \sigma_{ab} \|\vec{v}_a - \vec{v}_b\| \left[F^{N_i}\left(J\left(\overline{v}, a, b, \vec{e}\right), t\right) - F^{N_i}\left(\overline{v}, t\right) \right]\ d\vec{e}. \\ F^{N_i}\left(\overline{v}, 0\right) = \prod\limits_{j=1}^{N_i} \hat{f}\left(0, \vec{v}_j\right) \end{cases} \tag{2.74}$$

where J is defined by Eq. (2.73), $\hat{f}$ by Eq. (2.71). This master equation is very similar to those proposed in [89] and [107]. Eq. (2.74) constitutes a Fokker-Planck equation [48] for the density $p\left(\overline{v}, t\right) \triangleq F^{N_i}\left(\overline{v}, t\right)$ of the Markov process $\overline{v}\left(t\right)$ describing the collision between particles. We will make use of that fact in the next section to derive its infinitesimal generator. The realization that the DSMC procedure actually consists in solving the Master equation given by Eq. (2.74) was first advanced in 1975 in [10]. The DSMC method was, however, developed using phenomenological arguments without any recourse to such master equation. Further details about the connection between the Kac master equation and DSMC can be found in [83] [82]. Similar master equations are also commonly used to describe the evolution of physical systems that evolve between a countable set of states as discussed in [143] or [79].

---

[9] $\int\limits_{V=(\mathbb{R}^3)^{N_i}} F^{N_i}\left(\overline{v}\right) d\overline{v} = 1$

2.4 A derivation of the DSMC procedure

2.4.1 Derivation of a general DSMC algorithm

We now aim to derive specific expressions for *i)* the waiting time Δ_{coll} at step III)i), *ii)* the distribution function of particle indices to consider for collision $f_{\text{sel, coll}}(l_1, l_2)$ at step III)iii), and *iii)* the collision probability P_{coll} at step III)iv) in the general DSMC procedure that was presented above. This will be accomplished from basic mathematical considerations following [125]. This will provide a very strong motivation for the formulation of the DSMC algorithm, as the DSMC algorithm will be seen to be obtainable from the Kac Master equation (Eq. 2.74).

Before relating the DSMC procedure to the master equation, we first briefly highlight an important result that will later explain the fact that the DSMC procedure uses fictitious collisions. As the Markov process describing the collision between particles belongs to the class of Markov jump processes, its infinitesimal generator[10] $\mathcal{A}$ can be shown to be of the following form:

$$\mathcal{A}\phi\left(\overline{v}\right) = \lambda\left(\overline{v}\right) \int\limits_{V} \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] \mu\left(\overline{v}, d\overline{v}'\right). \tag{2.75}$$

where μ is the Markov transition function and λ is the waiting time. The transition function μ gives the probability of occurrence of a transition (jump) from state $\overline{v}$ to $\overline{v}'$, while λ corresponds to the parameter of the exponential distribution that describes the time during which the process stays in state $\overline{v}$ (the waiting time), which only depends on the current state of the process, $\overline{v}$. More details about the transition function and waiting times can be found in Appendix A. We now assume that the

---

[10] An infinitesimal generator describes in terms of differential operators the evolution of a test function ϕ which evolution is governed by a Markov Process (see Appendix A for a more rigorous definition)

waiting time is bounded, that is

$$0 < \lambda\left(\overline{v}\right) < \lambda_{\max},\ \forall\ \overline{v} \in V. \tag{2.76}$$

which will later be shown to be a valid assumption for a DSMC cell. Following, [54] p. 163, and using the fact that $\int_V \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] \delta_{\overline{v}}\left(d\overline{v}'\right) = 0$, Eq. (2.75) can equivalently be written as

$$\mathcal{A}\phi\left(\overline{v}\right) = \lambda_{\max} \int\limits_V \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] \left[\frac{\lambda\left(\overline{v}\right)}{\lambda_{\max}} \mu\left(\overline{v}, d\overline{v}'\right) + \left(1 - \frac{\lambda\left(\overline{v}\right)}{\lambda_{\max}}\right) \delta_{\overline{v}}\left(d\overline{v}'\right)\right]. \tag{2.77}$$

which, by comparison with Eq. (2.75), is seen to correspond to a Markov process with a waiting time that follows an exponential distribution of $\lambda_{\max}$ (independent of the present state $\overline{v}$). The second term of the integral in Eq. (2.77) corresponds to the infinitesimal generator of a Markov process whereby a jump occurs with a probability of $\frac{\lambda(\overline{v})}{\lambda_{\max}}$ while no jump (i.e. $\overline{v}' \equiv \overline{v}$) occurs with a probability of $1 - \frac{\lambda(\overline{v})}{\lambda_{\max}}$. The result of Eq. (2.77), which is valid for any Markov jump process, will later be use to derive the DSMC algorithm.

The infinitesimal generator of the stochastic process described by the master equation, Eq. (2.74), can be directly obtained from the fact that it constitutes a Fokker-Planck equation. By definition, a Fokker-Planck equation gives an evolution equation for the probability density function $p\left(\overline{v}, t\right)$ of a stochastic process $\overline{v}\left(t\right)$ [48] [98]. It furthermore can be written as:

$$\begin{cases} \frac{\partial p(\overline{v},t)}{\partial t} = \mathcal{A}_B^* p\left(\overline{v}, t\right) \\ p\left(\overline{v}, 0\right) = p_0\left(\overline{v}\right) \end{cases} \tag{2.78}$$

where $\mathcal{A}_B^*$ is the adjoint of the infinitesimal generator of the stochastic process. A “B”

subscript is used for this operator to highlight its connection with the Boltzmann/Kac master equation. By comparison with Eq. (2.74), with $\phi = F^{N_i}$ and dropping the time dependency for conciseness:

$$\mathcal{A}_B^* \phi\left(\overline{v}\right) = \frac{W_p}{2\Omega_i} \sum_{1 \leq a \neq b \leq N_i} \int_{\mathbb{S}^2} \sigma_{ab} \| \vec{v}_a - \vec{v}_b \| \left[\phi\left(J\left(\overline{v}, a, b, \vec{e}\right)\right) - \phi\left(\overline{v}\right) \right] \, d\vec{e}. \tag{2.79}$$

which shows that $\mathcal{A}_B^* = \mathcal{A}_B$. The infinitesimal generator $\mathcal{A}_B$ can be rewritten as

$$\mathcal{A}_B \phi\left(\overline{v}\right) = \int_V \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] \mathcal{Q}\left(\overline{v}, d\overline{v}'\right). \tag{2.80}$$

where

$$\mathcal{Q}\left(\overline{v}, d\overline{v}'\right) = \frac{1}{2} \sum_{1 \leq a \neq b \leq N_i} \int_{\mathbb{S}^2} \delta_{J(\overline{v}, a, b, e)}\left(d\overline{v}'\right) q\left(\overline{v}, a, b, e\right) d\vec{e}. \tag{2.81}$$

Eq. (2.80) gives an expression of the evolution of test function ϕ in a cell due to collisions. It is obtained by summing the product of the intensity function $q\left(\overline{v}, a, b, e\right)$ and the collision operator $\delta_{J(\overline{v}, a, b, e)}$, corresponding to each collision for all particle pairs in the cells. The intensity function is a quantitative measure of the probability of collision (jump) during a small time interval so that from Eq. (2.63):

$$q\left(\overline{v}, a, b, e\right) = \frac{W_p}{V} \int_{\mathbb{S}^2} B\left(\| \vec{v}_a - \vec{v}_b \|, \vec{e}\right) de. \tag{2.82}$$

where from Eq. (2.45), $B = \sigma_{ab} \| \vec{v}_a - \vec{v}_b \|$. To simplify the presentation in the following, we will assume that the collision kernel $B\left(\vec{v}_i, \vec{v}_j, \vec{e}\right)$ is independent of the scattering angle χ so that $B\left(\vec{v}_i, \vec{v}_j, \vec{e}\right) = B\left(\vec{v}_i, \vec{v}_j\right)$ which is true (cf. Eq. (2.67)) in the case of the variable hard sphere model that is employed throughout this thesis so that

$$\int_{\mathbb{S}^2} B\left(\| \vec{v}_a - \vec{v}_b \|, \vec{e}\right) de = 4\pi \; B\left(\| \vec{v}_a - \vec{v}_b \|\right) \tag{2.83}$$

and

$$\int_{\mathbb{S}^2} q(\overline{v}, a, b, e)\, de = 4\pi\ q(\overline{v}, a, b) \tag{2.84}$$

which allows Eq. (2.81) to be rewritten as:

$$\mathcal{Q}(\overline{v}, d\overline{v}') = 2\pi \sum_{1 \leq a \neq b \leq N_i} \delta_{J(\overline{v},a,b,e)}(d\overline{v}')\, q(\overline{v}, a, b, e)\,. \tag{2.85}$$

so that inserting the new expression into Eq. (2.80),

$$\begin{aligned}\mathcal{A}_B \phi(\overline{v}) = \int_V [\phi(\overline{v}') - \phi(\overline{v})]\, 2\pi \sum_{1 \leq a \neq b \leq N_i} \delta_{J(\overline{v},a,b,e)}(d\overline{v}')\, q(\overline{v}, a, b, e) = \\ 2\pi \sum_{1 \leq a \neq b \leq N_i} \int_V [\phi(\overline{v}') - \phi(\overline{v})]\ \delta_{\overline{v},a,b}(d\overline{v}')\, q(\overline{v}, a, b).\end{aligned} \tag{2.86}$$

We designate by $\hat{q}$ [11] any function such that

$$\hat{q}(\overline{v}, a, b) \geq \int_{\mathbb{S}^2} q(\overline{v}, a, b)\, de = 4\pi q(\overline{v}, a, b) \quad \forall\, \overline{v}, a, b \tag{2.87}$$

and using Eq. (2.77) with $\lambda(\overline{v}) = q(\overline{v}, a, b)$, $\lambda_{\max} = \hat{q}(\overline{v}, a, b)$ and $\mu(\overline{v}, d\overline{v}') = \delta_{\overline{v},a,b}(d\overline{v}')$ for all a, b pairs, the infinitesimal generator of Eq. (2.86) can be equivalently written as:

$$\begin{aligned}\mathcal{A}_B \phi(\overline{v}) = \frac{1}{2} \sum_{1 \leq a \neq b \leq N_i} \int_V [\phi(\overline{v}') - \phi(\overline{v})] \\ \Big[4\pi\ \delta_{J(\overline{v},a,b,e)}(d\overline{v}')\, q(\overline{v}, a, b) + \delta_{\overline{v}}(d\overline{v}')\left(\hat{q}(\overline{v}, a, b) - 4\pi\ q(\overline{v}, a, b)\right)\Big]\,.\end{aligned} \tag{2.88}$$

By defining bilinear form $\hat{\mathcal{Q}}$ as

$$\hat{\mathcal{Q}}(\overline{v}, d\overline{v}') = \frac{1}{2}\left[4\pi\ \delta_{\overline{v},a,b}(d\overline{v}')\, q(\overline{v}, a, b) + \delta_{\overline{v}}(d\overline{v}')\left(\hat{q}(\overline{v}, a, b) - 4\pi\ q(\overline{v}, a, b)\right)\right], \tag{2.89}$$

---

[11] The choice of this function represents a crucial part of the DSMC method. It will later be discussed in Section 2.4.2.

Eq. (2.88) can rewritten as

$$\mathcal{A}_B \phi\left(\overline{v}\right) = \sum_{1 \leq a \neq b \leq N_i} \int_V \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] \hat{\mathcal{Q}}\left(\overline{v}, d\overline{v}'\right). \tag{2.90}$$

We now seek to recast the differential generator of Eq. (2.90) under the form of a jump process with a waiting time and transition function following Eq. (2.75). This can be done in a variety of ways, but the most natural choice is to define the waiting $\hat{\lambda}\left(\overline{v}\right)$ as:

$$\hat{\lambda}\left(\overline{v}\right) = \int_V \hat{\mathcal{Q}}\left(\overline{v}, d\overline{v}'\right) = \frac{1}{2} \sum_{1 \leq a \neq b \leq N_i} \hat{q}\left(\overline{v}, a, b\right) \tag{2.91}$$

such that Eq. (2.90) may be equivalently rewritten as

$$\mathcal{A}_B \phi\left(\overline{v}\right) = \hat{\lambda}\left(\overline{v}\right) \sum_{1 \leq a \neq b \leq N_i} \int_V \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] \frac{\hat{\mathcal{Q}}\left(\overline{v}, d\overline{v}'\right)}{\hat{\lambda}\left(\overline{v}\right)}. \tag{2.92}$$

By analogy with Eq. (2.75), the jump process described by Eq. (2.92), has a waiting time of $\hat{\lambda}\left(\overline{v}\right)$ and a transition function given by $\frac{2\pi \hat{\mathcal{Q}}\left(\overline{v}, d\overline{v}'\right)}{\hat{\lambda}\left(\overline{v}\right)}$. Using the definition of $\hat{\mathcal{Q}}$ given by Eq. (2.89), the complete form of the infinitesimal generator is given by:

$$\begin{aligned} \mathcal{A}_B \phi\left(\overline{v}\right) = \hat{\lambda}\left(\overline{v}\right) \sum_{1 \leq a \neq b \leq N_i} \int_V \left[\phi\left(\overline{v}'\right) - \phi\left(\overline{v}\right)\right] & \\ \left(\frac{\hat{q}\left(\overline{v}, a, b\right)}{2\hat{\lambda}\left(\overline{v}\right)}\right) \left[\delta_{J\left(\overline{v}, a, b\right)}\left(d\overline{v}'\right) \frac{4\pi\ q\left(\overline{v}, a, b\right)}{\hat{q}\left(\overline{v}, a, b\right)} + \delta_{\overline{v}}\left(d\overline{v}'\right) \left(1 - \frac{4\pi\ q\left(\overline{v}, a, b\right)}{\hat{q}\left(\overline{v}, a, b\right)}\right)\right]. & \end{aligned} \tag{2.93}$$

Eq. (2.93) is seen to have an identical form to that of Eq. (2.77) so it can be interpreted in the same manner with the only difference that the jumps will be referred to as *collisions*. That is, the infinitesimal generator describes a jump Markov process whereby a collision occurs with a probability P_{coll}:

$$P_{\text{coll}} = \frac{4\pi\ q\left(\overline{v}, a, b\right)}{\hat{q}\left(\overline{v}, a, b\right)} \tag{2.94}$$

while its state is unchanged with a probability of $(1 - P_{\text{coll}}) = \left(1 - \frac{4\pi q(\overline{v},a,b)}{\hat{q}(\overline{v},a,b)}\right)$. The case when the system does not change state is referred to as a *fictitious jump.* Waiting times, i.e. the amount of times between potential jumps (real and fictitious), follow an exponential distribution and, by construction, its parameter is given by Eq. (2.91). The extra factor $\left(\frac{\hat{q}(\overline{v},a,b)}{2\hat{\lambda}}\right)$ which is not present in Eq. (2.77) governs the overall collision probability as any factor less than 1 and independent of $d\overline{v}'$ in the integral of an infinitesimal generator simply reduces the value of the transition probability. The probability of a collision (real or fictitious) for a particular a, b pair is thus given by $\left(\frac{\hat{q}(\overline{v},a,b)}{2\hat{\lambda}}\right)$. It can be seen that the Markov process that was just described corresponds to point III)iii) of Algorithm 2.3. As collision probabilities are proportional to $\left(\frac{\hat{q}(\overline{v},a,b)}{2\hat{\lambda}}\right)$, this last term provides an expression for the distribution of indices of colliding particles. From the standpoint of implementing the Markov process, this means that l_1 and l_2 (the indices of colliding particles) should be selected according to that distribution, i.e.,

$$f_{\text{sel, coll}}\left(l_1, l_2\right) = \frac{\hat{q}\left(\overline{v}, a, b\right)}{2\hat{\lambda}\left(\overline{v}\right)}. \tag{2.95}$$

which does satisfy the distribution normalization condition because of the definition of $\hat{\lambda}\left(\overline{v}\right)$ following Eq. (2.91):

$$\sum_{\substack{l_2=1 \\ l_2 \neq l_1}}^{N} \sum_{l_1=1}^{N} f_{\text{sel, coll}}\left(l_1, l_2\right) = 1.$$

Eq. (2.95) thus gives the distribution function of particle indices to consider for collision $f_{\text{sel, coll}}\left(l_1, l_2\right)$ at step III)iii) of Algorithm 2.3. The waiting time Δt_{coll}, which is needed at step III)i) of the pseudo code, is obtained by sampling from an exponential distribution, i.e.

$$\Delta t_{\text{coll}} \sim \operatorname{Exp}\left(\hat{\lambda}\left(\overline{v}\right)\right)$$

where the parameter $\hat{\lambda}(\overline{v})$, defined through Eq. (2.91) is given by

$$\hat{\lambda}(\overline{v}) = \frac{1}{2} \sum_{1 \leq a \neq b \leq N_i} \hat{q}(\overline{v}, a, b)$$

It was also shown (cf. Eq (2.94)) that collisions should be performed with a probability of $P_{\text{coll}} = \frac{4\pi q(\overline{v},a,b)}{\hat{q}(\overline{v},a,b)}$ which thus specifies the probability of a fictitious or real collision in points III)iv) of the pseudo code.

We have thus shown that the collision part of a general DSMC procedure can be rigorously derived from a Markov process that seeks to model a Boltzmann master equation for each cell of the computational domain. At this point, the only thing left to do to obtain the full DSMC procedure is to specify function $\hat{q}(\overline{v}, a, b)$, which will be done in the next section and will in turn allow a rigorous derivation of all possible DSMC pair selection schemes.

2.4.2 Pair selection schemes for DSMC

2.4.2.1 Direct simulation scheme

We will first derive the most basic DSMC scheme possible, the direct simulation scheme which will in turn motivate the use of the scheme that is used throughout this thesis, the no time counter scheme. As previously mentioned, the only remaining step to fully characterize, and thus to simulate, the Markov process described by Eq. (2.93) is function $\hat{q}(\overline{v}, a, b)$ for which the only constraint is (cf. Eq. (2.87))

$$4\pi q(\overline{v}, a, b) \leq \hat{q}(\overline{v}, a, b) \quad \forall\, \overline{v}, a, b \tag{2.96}$$

By the definition of Eq. (2.82) and using Eqs. (2.84) and (2.83),

$$q(\overline{v}, a, b) = \frac{W_p}{V} B(\vec{v}_a, \vec{v}_b).$$

and limiting ourselves to a VHS collision kernel, where from Eq. (2.67):

$$B\left(\vec{v}_i, \vec{v}_j\right) = \left\|\vec{v}_a - \vec{v}_b\right\| \sigma_{\mathrm{VHS}}\left(\vec{v}_i, \vec{v}_j\right) = \frac{C_{\mathrm{VHS}}}{\left\|\vec{v}_a - \vec{v}_b\right\|^{\omega_{\mathrm{VHS}}-1}}, \tag{2.97}$$

and hence

$$q\left(\overline{v}, a, b\right) = \frac{W_p}{V} \frac{C_{\mathrm{VHS}}}{\left\|\vec{v}_a - \vec{v}_b\right\|^{\omega_{\mathrm{VHS}}-1}}. \tag{2.98}$$

the most natural choice is to pick the smallest possible bound, which according to Eq. (2.96) is

$$\hat{q}\left(\overline{v}, a, b\right) = 4\pi q\left(\overline{v}, a, b\right) = \frac{4\pi W_p\ C_{\mathrm{VHS}}}{V\left\|\vec{v}_a - \vec{v}_b\right\|^{\omega_{\mathrm{VHS}}-1}}. \tag{2.99}$$

Following Eq. (2.91), this leads to the following parameter for the exponential distribution of the waiting time:

$$\hat{\lambda}\left(\overline{v}\right) = \frac{1}{2} \sum_{1 \leq a \neq b \leq N_i} \hat{q}\left(\overline{v}, a, b\right) = \frac{2\pi W_p C_{\mathrm{VHS}}}{V} \sum_{1 \leq a \neq b \leq N_i} \left\|\vec{v}_a - \vec{v}_b\right\|^{1-\omega_{\mathrm{VHS}}}, \tag{2.100}$$

while the distribution of indices of colliding particles, following Eq. (2.95), has to occur according to the following distribution:

$$\begin{aligned} f_{\text{sel, coll}}\left(l_1, l_2\right) = \frac{\hat{q}\left(\overline{v}, l_1, l_2\right)}{2\hat{\lambda}\left(\overline{v}\right)} &= \frac{4\pi \frac{C_{\mathrm{VHS}}}{W_p\ V\left\|\vec{v}_a - \vec{v}_b\right\|^{\omega_{\mathrm{VHS}}-1}}}{\frac{4\pi C_{\mathrm{VHS}}}{W_p V} \sum_{1 \leq a \neq b \leq N_i} \left\|\vec{v}_a - \vec{v}_b\right\|^{1-\omega_{\mathrm{VHS}}}} \\ &= \frac{\left\|\vec{v}_a - \vec{v}_b\right\|^{1-\omega_{\mathrm{VHS}}}}{\sum_{1 \leq a \neq b \leq N_i} \left\|\vec{v}_a - \vec{v}_b\right\|^{1-\omega_{\mathrm{VHS}}}} \end{aligned} \tag{2.101}$$

where $\hat{\lambda}\left(\overline{v}\right)$ and $q\left(\overline{v}, a, b\right)$ are, respectively, obtained with Eqs. (2.100) and (2.99). The probability of a real collision, given with Eq. (2.94) and using the definition of $\hat{q}\left(\overline{v}, a, b\right)$ is found to be $P_{\mathrm{coll}} = 1$. This means that within the framework of the direct simulation scheme, no fictitious collisions take place. The large drawback of the present scheme lies in the computational effort required for the calculation of the parameter of the exponential distribution of the waiting time and in the calculation of the distribution used to select the indices of colliding particles which both sum over all particle pairs of the cell, i.e. $\sum_{1\leq a\neq b\leq N_i}(\cdot)$. This represent a substantial computational effort, as these have to be calculated multiple times during every time step Δt, that is, until the time counter, t_c, exceeds Δt. This has led to the development of a number of more efficient pair selection schemes for DSMC such as the no time counter which is presented in the next section.

2.4.2.2 No time counter scheme

The no time counter scheme can be obtained by using the following bound for $\hat{q}\left(\overline{v},a,b\right)$:

$$\hat{q}\left(\overline{v},a,b\right)=\frac{4\pi W_p C_{\text{VHS}}}{V\|\vec{v}_{r,\max}\|^{\omega_{\text{VHS}}-1}}, \tag{2.102}$$

where $\|\vec{v}_{r,\max}\|=\max\limits_{a,b}\|\vec{v}_a-\vec{v}_b\|$, which is suitably updated for each individual cell of the simulation whenever a particle pair with greater relative velocity is encountered. Using this expression for $\hat{q}\left(\overline{v},a,b\right)$, the parameter of the exponential distribution for the waiting time is found to be using Eq. (2.91)

$$\hat{\lambda}\left(\overline{v}\right)=\frac{1}{2}\sum_{1\leq a\neq b\leq N_i}\hat{q}\left(\overline{v},a,b\right)=\frac{2\pi W_p}{V}\,C_{\text{VHS}}\|\vec{v}_{r,\max}\|^{1-\omega_{\text{VHS}}}N\left(N-1\right), \tag{2.103}$$

while, from Eq. (2.95), the two indices l_1 and l_2 of the particles to select for potential collisions follow a uniform distribution $f_{\text{sel, coll}}\left(l_1,l_2\right)$, i.e.

$$f_{\text{sel, coll}}\left(l_1,l_2\right)=\frac{1}{N^2} \tag{2.104}$$

so that both particles to consider for a potential collision are randomly selected amongst all those present in the cell. Finally, the collision probability is found using Eq. (2.94)

$$P_{\text{coll}}=\frac{4\pi q\left(\overline{v},a,b\right)}{\hat{q}\left(\overline{v},a,b\right)}=\frac{\|\vec{v}_a-\vec{v}_b\|^{\omega_{\text{VHS}}-1}}{\|\vec{v}_{r,\max}\|^{\omega_{\text{VHS}}-1}} \tag{2.105}$$

This formulation of the scheme at this point can be further simplified which in turn allows the full recovery of the formulation of Algorithm 2.2, where it was initially introduced. The waiting time $\tau\left(\overline{v}\right)$ need not be obtained through the sampling of the exponential distribution if its value is small. As it is obtained from an exponential distribution with parameter λ, $\mathbb{E}\left[\tau\left(\overline{v}\right)\right] = \lambda^{-1}\left(\overline{v}\right)$ and $\mathrm{Var}\left[\tau\left(\overline{v}\right)\right] = \lambda^{-2}\left(\overline{v}\right)$, so that if $\lambda^{-1}\left(\overline{v}\right) \ll 1$, $\tau\left(\overline{v}\right) \approx \lambda^{-1}\left(\overline{v}\right)$. This means that the value of the waiting time can be obtained deterministically from Eq. (2.103) without any sampling. As the expression furthermore only depends on the number of particles in the cell, N, and $\|\vec{v}_{r,\max}\|$ which is constant (except for a few time steps in the ramp up to steady state state), it only has to be evaluated once per time step Δt. The number of collisions (real or fictitious) that occur in a cell N_{coll} during time step Δt is thus known at the onset of the time step. It is obtained by equating the time step Δt to the fixed waiting time between each collision multiplied by the number of collisions, so that

$$N_{coll} = \frac{\Delta t}{\tau\left(\overline{v}\right)} = \Delta t\ \lambda\left(\overline{v}\right) \tag{2.106}$$

so that using the expression of Eq. (2.103) for $\lambda\left(\overline{v}\right)$ and specializing the result to a cell with index i:

$$N_{\mathrm{coll},i} = \frac{\Delta t}{\tau\left(\overline{v}\right)} = \frac{\Delta t W_p}{2\Omega_i}\ 4\pi\ C_{\mathrm{VHS}}\|\vec{v}_{r,i,\max}\|^{1-\omega_{\mathrm{VHS}}} N_i\left(N_i - 1\right) \tag{2.107}$$

By definition of the total collision cross section (cf. Eq. (2.53)) $\sigma_T = 4\pi C_{\mathrm{VHS}}\|\vec{v}_r\|^{-\omega_{\mathrm{VHS}}}$, so that Eq. 2.107 can be rewritten as

$$N_{\mathrm{coll},i} = \frac{\Delta t W_p}{2\Omega_i} N_i\left(N_i - 1\right)\left(\|\vec{v}_r\|\sigma_T\right)_{i,\max} \tag{2.108}$$

which is the exact same expression for the number of pairs to select for potential collision as that of Eq. (2.64) in Algorithm 2.2. The collision probability given by

Eq. (2.105) is furthermore identical to that of the pseudo-code, i.e. (Eq. 2.65). We have thus rigorously derived the no time counter scheme from the master equation that is solved during the collision part of the DSMC procedure. The computational advantages of the present scheme compared to the direct simulation scheme that was previously presented are obvious, as the number of particle pairs to test for collision (or equivalently the waiting time) can be determined without the need to loop over all pairs in the cell as in Eq. (2.103). The absence of a time counter further contributes to the computational efficiency of the scheme.

2.4.3 DSMC and the Boltzmann equation

As previously mentioned in Section 2.3.4, the homogeneous part of the DSMC algorithm consists in simulating the evolution of N_i particles in each individual cell of the domain following a Boltzmann master equation (Eq. 2.74). By construction of the collision algorithm, it is exactly simulated by the DSMC procedure. It, however, of course differs from the homogeneous Boltzmann equation (Eq. 2.70). The Boltzmann and master equations are reproduced below to highlight differences, keeping in mind that the distribution function in the master equation is normalized.

$$\begin{cases} \frac{\partial f}{\partial t}\left(\vec{v},t\right) = \underset{\mathbb{R}^3\ \mathbb{S}^2}{\iint} \sigma v_r \left[f\left(\vec{v}'_1,t\right) f\left(\vec{v}',t\right) - f\left(\vec{v}_1,t\right) f\left(\vec{v},t\right)\right]\ d\vec{e}\ d\vec{v}_1 \\ f\left(\vec{v},0\right) = f^0\left(\vec{v}\right) \end{cases} \tag{2.109}$$

$$\begin{cases} \frac{\partial F^{N_i}\left(\overline{v},t\right)}{\partial t} = \frac{W_p}{2\Omega_i} \sum_{1\leq a\neq b\leq N_i} \int_{\mathbb{S}^2} \sigma_{ab} \|\vec{v}_a - \vec{v}_b\| \left[F^{N_i}\left(J\left(\overline{v},a,b,\vec{e}\right),t\right) - F^{N_i}\left(\overline{v},t\right)\right]\ d\vec{e} \\ F^{N_i}\left(\overline{v},0\right) = \prod_{j=1}^{N_i} \hat{f}\left(0,v_j\right) \end{cases} \tag{2.110}$$

Despite the lack of non-linearity in the distribution function in Eq. (2.110), this equation can be shown, in the limit $N_i \to \infty$, to produce the same measure. For each variable $\overline{v}$, the following empirical measure $\mu_{\overline{v}^{N_i}}$ is defined

$$\mu_{\overline{v}_t^{N_i}} = \frac{1}{N_i} \sum_{j=1}^{N_i} \delta_{\vec{v}_j(t)}, \tag{2.111}$$

which, when $N_i \to \infty$, has the property that

$$\mathbb{E}\left(|\langle \mu_{\overline{v}_t^{N_i}} - \hat{f}\,(t) \rangle| \right) \to 0. \tag{2.112}$$

The proof for the DSMC no time counter scheme was first given in [148] where Eq. 2.112 corresponds to part of Theorem 4.4 therein. It is also discussed at length in [125]. As it is fairly technical and builds on previous work in the area, it is not discussed further here.

CHAPTER III

On the convergence of the direct simulation Monte Carlo method

3.1 Introduction

Error in DSMC simulations can be defined in the broadest possible sense as the difference between the statistical properties of the computational particles and the corresponding moments of the exact solution of the Boltzmann equation. This error originates from multiple sources which either mitigate or reinforce one another. It can come from the use of finite numerical parameters in the application of the DSMC method which introduces error due to the numerical scaling (as each computational particle represents W_p physical particles, and both time and space are discretized). This error is most commonly due to the use of a too small number of particles N_p, a too large time step Δt, or too large cells. This type of error will be termed *numerical error* in the following. Error can also be due to the inability of DSMC, in the absence of numerical error, to reproduce the moments of the Boltzmann equation. This issue will not be addressed in the present work noting that DSMC has been shown to consistently model the homogeneous Boltzmann equation [148]. Finally, error is introduced by the use of a finite number of samples k to estimate the statistical properties of the particles in the simulation. It will be referred to as the *convergence error* in this thesis. Numerical error, under a variety of names, has been previously studied among others in [55], [41], [124] or [134]. Convergence error is referred to as the *statistical error* in [77] and was found in [55] to vary as $A\left(N \times k\right)^{-1/2}$. The same result was later obtained in [41]. The error was related in [77] to the physical fluctuations in the gas and closed form expressions were obtained in terms of non-dimensional flow numbers for A. This thesis will first present a rigorous definition of both types of error before discussing in detail the convergence error. The issue of numerical error will be addressed in Chapter IV.

From an error perspective, the role of sampling in a DSMC simulation is to reduce the convergence error affecting the quantities being sampled for. Despite the previous investigations of the convergence error cited above, there are no clear requirements within the DSMC community as to the number of sampling steps required to obtain accurate or converged results for those sampled properties. Often an arbitrarily large number of samples, e.g. 10^5 as in [24] is used without further considerations or the sampling period is extended [149] "until the statistical error is small enough". This situation is in no small part due to the correlation between samples which prevents the use of the central limit theorem. For instance, the predictions of the statistical error due to finite sampling presented in [77] only hold in the absence of correlation between samples which leads its authors to sample only every 250 time steps in the DSMC simulations conducted to validate them. A large number of authors also try to reduce the correlation between successive samples by only sampling every few time steps such as [151] or [106] to obtain more independent samples. This shows that the time correlation between samples in DSMC simulations is a well known issue although it has not been accounted for in any existing work about the convergence error except recently in [119]. That latter work was however only concerned in obtaining upper bounds (which were subsequently used in [120]) for the convergence error and did not examine in detail the correlation between samples (in particular the autocorrelation function), which we propose to do in the following.

The present work thus aims to derive expressions for the convergence error that take the correlation of samples into account.
In this chapter, a rigorous formal definition of both the convergence and numerical error in DSMC simulations is first presented. Building on that formulation, the behavior of the convergence error is then investigated while that of the numerical error is addressed in Chapter IV. The specific goal of the present chapter is to provide an expression for the value of the convergence error that can be used for its a priori determination during the course of the simulation. As previously mentioned, the convergence error is greatly affected by the time correlation between samples. This makes it necessary to quantify the correlation between samples which is performed in Section 3.5.1. A central limit theorem that takes correlation into account is then introduced. It provides an expression for the aforedefined convergence error in terms of the autocorrelation function of samples. This expression can readily be used ”on the fly” during the simulation to assess the convergence of cellwise sampled quantities. Convergence error predictions are then assessed within the framework of a test case that consists of an axisymmetric jet. Such a relatively complex test case is chosen because it is more representative of the type of complex flows that DSMC practitioners are likely to encounter in practice than, for instance, channel flow, which is often studied, as in e.g. [77] or [139]. The cellwise value of the convergence error will be examined in this work as opposed to a global error value such as was done for instance in [55] of [41]. This is because, as it will later be seen, the pointwise value of the error varies greatly throughout the domain so that a global error measure gives little information on the state of convergence of sampled quantities in discrete cells.

3.2 Framework for error analysis

3.2.1 Implementation of the DSMC procedure

A simple argon gas in axisymmetric domain $\Omega = \{(x, y) \mid 0 \leq x \leq x_1, 0 \leq y \leq y_1\} \subset \mathbb{R}^2$ rotated about $y = 0$ with velocities in $E \subset \mathbb{R}^3$ is considered. $\overline{\Omega}$ is decomposed into N_c quadrangular cells $\{\Omega_i\}_{i=1}^{N_c}$ of respective volume $\{V_i\}_{i=1}^{N_c}$ via quadrangulation Ω_h:

$$\overline{\Omega} = \bigcup_{\Omega_i \in \Omega_h} \Omega_i \tag{3.1}$$

such that $\mathrm{V}_i \leq h,\ 1 \leq i \leq N_c$. The number density n in a cell containing N particles is obtained with

$$n = \frac{W_p N}{V}, \tag{3.2}$$

each computational particle representing W_p physical particles so that the entire domain contains N_p particles. A constant time step Δt and scaling factor W_p are used throughout the domain. The axisymmetric move procedure detailed in [15] p. 371 is employed. The standard NTC (No Time Counter) collision scheme [15] is used to calculate the number of potential collisions in each cell:

$$N_{\text{coll}} = \frac{1}{2\mathrm{V}} W_p \Delta t N \overline{N} \left(\sigma g\right)_{\max}, \tag{3.3}$$

where $\overline{N}$, the average number of particles in the cell, is obtained by an exponential moving average with a relaxation factor of 0.001. Binary elastic collisions are modeled via the variable hard sphere model [15] with a reference diameter of 4.17×10^{-10} m at 273 K and a temperature exponent of 0.81. Further details concerning the parallel implementation of the DSMC procedure can be found in [50].

3.2.2 Cell and particle based variables

Any variable characterizing particle j at time t^k is denoted by $\mathbf{y}_j^k$ with $1 \leq j \leq N_p$ which can for instance represent its instantaneous position $\vec{\mathbf{x}}_j^k$ or velocity $\vec{\mathbf{v}}_j^k$. Cell-based quantities are denoted by y_i^k with $1 \leq i \leq N_c$ which designates any variable specific to cell i at time t^k such as the number of particles contained in the cell, N_i^k. When all particles or all cells of the simulations are simultaneously considered, particle and cell-based quantities can both be represented by a series of vectors, namely $\vec{\mathbf{y}}^k = \left(\mathbf{y}_1^k, ..., \mathbf{y}_{N_p}^k\right)$ and $\vec{y}^k = \left(y_1^k, ..., y_{N_c}^k\right)$. The average value of $\mathbf{y}$ in cell Ω_i at time step k is naturally defined as:

$$y_i^k = \langle \mathbf{y} \rangle_{\Omega_i}^k = \frac{\sum_{j=1}^{N_i^k} \mathbf{y}_j^k}{N_i^k}. \tag{3.4}$$

The transformation of particle-based quantities to cell-based ones can more formally be seen as a linear transformation. To formalize the transformation, an indicator function $\mathbf{1}_{\Omega_i} : \mathbf{R}^2 \to \{0, 1\}$ is defined as

$$\mathbf{1}_{\Omega_i}(\mathbf{p}) := \begin{cases} 1 & \text{if } \mathbf{p} \in \Omega_i \\ 0 & \text{if } \mathbf{p} \notin \Omega_i \ , \end{cases} \tag{3.5}$$

which determines whether or not a given particle located at position $\mathbf{p}$ belongs to Ω_i. This function can be used to give a more rigorous definition of number density than that of Eq. (3.2):

$$n_i^k = \frac{W_p \sum_{j=1}^{N_p} \mathbf{1}_{\Omega_i}(\vec{\mathbf{x}}_j^k)}{\mathrm{V}_i}. \tag{3.6}$$

This clearly indicates that the number density in cell i only depends on the location $\left\{\vec{\mathbf{x}}_j^k\right\}_{j=1}^{N_p}$ of all the particles in the simulation. Thus, errors for the statistical properties of the number density in a simulation can only arise from errors for the location of particles in the simulation, i.e. $\left\{\vec{\mathbf{x}}_j^k\right\}_{j=1}^{N_p}$. The indicator function can also be used to rewrite Eq. (3.4) more rigorously as:

$$y_i^k = \langle \mathbf{y} \rangle_{\Omega_i}^k = \frac{\sum_{j=1}^{N_p} \mathbf{y}_j^k \; \mathbf{1}_{\Omega_i}(\vec{\mathbf{x}}_j^k)}{\sum_{j=1}^{N_p} \mathbf{1}_{\Omega_i}(\vec{\mathbf{x}}_j^k)}. \tag{3.7}$$

which shows that the projection of a particle-based variable on the mesh depends on both the position of all particles $\left\{\vec{\mathbf{x}}_j^k\right\}_{j=1}^{N_p}$ and the value of the particular variable $\left\{\mathbf{y_j^k}\right\}_{j=1}^{N_p}$. This in turn means that the sampled value of velocity in a cell will be affected by errors in both the location $\left\{\vec{\mathbf{x}}_j^k\right\}_{j=1}^{N_p}$ and velocities $\left\{\vec{\mathbf{v}}_j^k\right\}_{j=1}^{N_p}$ of all the particles of the domain.

3.2.3 Statistical estimators

The stochastic particle system engendered by the DSMC procedure is related in practice to the Boltzmann equation by averaging quantities over multiple realizations (i.e., time steps). This is due to the statistical fluctuations inherent to the DSMC formulation which make individual realizations usually too noisy for practical use. The statistical fluctuations of quantities of the DSMC particle system are themselves tied to those of the actual physical system. For instance, the statistical properties of DSMC particles in a given computational cell are identical to those that are present in the same physical cell. One can, in a sense, view DSMC or other particle methods as numerical experiments where measurements are made and collected to estimate statistical properties. Due to the finite number of samples, the estimation of the true statistical properties of the system is imperfect and leads to what is later termed the *convergence error*. It is thus appropriate to distinguish between the true statistical properties of the DSMC gas and their estimation obtained through sampling. Let Y_i be any cell-based variable, e.g. N_i, and let y_i^k designate an observed output of that random variable at time step k. $\theta\left(Y_i\right)$ refers to any statistical property of random variable Y_i such as for example its first order moment μ. The estimator of $\theta\left(Y_i\right)$ at time step k is denoted by $\hat{\theta}^k\left(y_i\right)$. Because all the statistical estimators used are consistent i.e., $\lim_{k\to\infty} \hat{\theta}^k(y_i) = \theta(Y_i)$, in the following the limit of $\hat{\theta}^k(y_i)$ when $k \to \infty$ will be abbreviated as $\theta(Y_i)$. $\hat{\theta}^k(y_i)$ typically depends on all previously observed samples $\left(y_i^1, ..., y_i^k\right)$ if sampling occurs at every time step as is the case in the present work. For instance, the estimator for $\mu(Y_i) = \mathbb{E}[Y_i]$ (often referred to as the *sampled value* of y) at time step k is defined by:

$$\hat{\mu}_{1,i}^k(y) = \frac{\sum_{k'=1}^k y_i^{k'}}{k}, \tag{3.8}$$

whereas the following estimator is used for $\sigma^2(Y_i) = \mathrm{Var}(Y_i)$:

$$\hat{\sigma}_i^{2\ k}(y) = \hat{\mu}_{1,i}^k\left(y^2\right) - \left(\hat{\mu}_{1,i}^k(y)\right)^2 = \frac{1}{k}\sum_{k'=1}^{k}\left(y_i^{k'}\right)^2 - \left(\hat{\mu}_{1,i}^k(y)\right)^2. \tag{3.9}$$

In the case of particle-based variables such as velocities, a different statistical estimator is used. Although the transformation of Eq. (3.4) followed by the estimator defined by Eq. (3.8) could be used, greater variance reduction can be achieved by using estimator $\check{\mu}_i^k$ defined as:

$$\check{\mu}_i^k(\mathbf{y}) = \frac{\sum_{k'=1}^k \sum_{j=1}^{N_i^{k'}} \mathbf{y}_j^{k'}}{\sum_{k'=1}^k N_i^{k'}}. \tag{3.10}$$

The use of this estimator amounts to considering each particle at each time step as a sampled value of the velocity distribution (in the case of velocity). This is indeed consistent with the DSMC formulation that considers particles as indistinguishable and the steady state assumption which considers time steps indistinguishable. Because of the time correlation between time steps, which will be studied in section 3.5.1, the samples are, however, not independent which, as will later be seen, reduces the rate of convergence of statistical estimators. The variance of $\mathbf{y}$ is obtained similarly to that of y with $\check{\sigma}^2(\mathbf{y}) = \check{\mu}_i^k(\mathbf{y}^2) - \check{\mu}_i^k(\mathbf{y})^2$. To simplify notations in the following, $\check{\mu}$ will be denoted by $\hat{\mu}$ with the understanding that $\check{\mu}$ defined by Eq. (3.10) is used for all particle-based variables instead of the estimator defined by Eq. (3.8).

The variance of number density can be calculated from Eq. (3.2):

$$\mathrm{Var}\left(n_i\right) = \left(\frac{W_p}{\mathrm{V}_i}\right)^2 \mathrm{Var}\left(N_i\right). \tag{3.11}$$

Because N_i follows a Poisson distribution for cells with $N_i \gg 1$ [64], $\mathrm{Var}\left(N_i\right) = \mathbb{E}\left[N_i\right]$, so that

$$\mathrm{Var}\left(n_i\right) = \left(\frac{W_p}{\mathrm{V}_i}\right) \mathbb{E}\left[n_i\right]. \tag{3.12}$$

Thus, for a given cell, when W_p is varied $\mathrm{Var}\left(n_i\right) \sim W_p$. This means that the variance of number density is a quantity that directly depends on the number of particles used in the simulation. Its variance will decrease when the value of W_p is reduced (or equivalently when the number of particles is increased). In contrast to that, the variance of a particle-based variable does not depend on the numerical parameters of the simulation. Taking one velocity component as an example, e.g. $\mathbf{y} = V_x$, its variance obtained with Eq. (3.9) is equal to the translational temperature of the particles, both computational and physical, in the cell.

3.2.4 Error types

Multiple sources of error are present in a DSMC simulation. They can be broadly placed into two categories. The first comes from the inadequacy of the DSMC procedure to properly model the Boltzmann equation in the absence of any scaling that is in the limit of $N \to \infty, \Delta t \to 0, k \to \infty$ and $h \to 0$. This can for instance be caused by an inadequate pair selection scheme or erroneous boundary conditions. This type of error will not be further considered in noting that DSMC has been proven [148] in the homogeneous case and in the limit of $N \to \infty$ to “converge” to the Boltzmann equation. The present chapter is concerned with the second type of error inherent to

DSMC simulations, those introduced by the use of finite values for W_p, Δt and k. As previously mentioned, the effect of the cell size, h, on the error will not be considered in the present work. The total error for statistical estimator $\theta\left(Y_i\right)$ of variable Y_i at step t^k of a simulation conducted with W_p and Δt is defined as being the difference between statistical estimator $\hat{\theta}^k\left(W_p, \Delta t\right)$ and θ obtained in the limit of $W_p \rightarrow 1$ and $\Delta t \rightarrow 0$. This total error is itself the result of two distinct inexactitudes which either reinforce or mitigate one another. The first is due to the inability to estimate the true value of an estimator with a finite number of samples and is called the *convergence error*. It is so named because it decreases, as long as the properties being sampled are stationary, with increasing k as estimators converge to actual values of the parameter being estimated. The second, called *numerical error*, is caused by the use of finite W_p and Δt values. Precise mathematical definitions are now given for the two aforementioned errors.

3.2.4.1 Convergence Error

The most natural definition of convergence error $\tilde{\varepsilon}_i^k$ for estimator $\hat{\theta}^k$ for cell i at time step k is:

$$\tilde{\varepsilon}_i^k = |\hat{\theta}_i^k\left(W_p, \Delta t\right) - \lim_{k \rightarrow \infty} \hat{\theta}_i^k\left(W_p, \Delta t\right)|. \tag{3.13}$$

Because of the stochastic nature of DSMC, the convergence of statistical property estimators is jagged and non-monotonic with the final value being both overshot and undershot multiple times during convergence. Thus, if only a single simulation is considered, the convergence error, as previously defined, does not monotonically decrease with time which makes its study difficult. This in turn suggests that a more probabilistic definition of the convergence error is needed. A natural choice is to define the convergence error $\tilde{\varepsilon}_i^k$ of estimator $\hat{\theta}_i^k$ at time step k as the standard deviation of $\hat{\theta}_i^k$ between multiple simulations conducted with the same set of numerical parameters but different sequences of random numbers. Assuming that N_{sim} simulations are conducted, each will produce a different value for $\hat{\theta}^k$ which is denoted by $\hat{\theta}^{k,l}$ with $1 \leq l \leq N_{\text{sim}}$. An ensemble average operator $\langle \cdot \rangle$ between simulations is defined as:

$$\langle z \rangle^k \triangleq \frac{1}{N_{\text{sim}}} \sum_{l=1}^{N_{\text{sim}}} z^{k,l} \tag{3.14}$$

for every time and simulation dependent quantity $z^{k,l}$. Applying the ensemble average operator to estimators $\hat{\theta}_i^{k,l}$, its ensemble average at time step k is obtained:

$$\langle \theta_i \rangle^k \triangleq \frac{1}{N_{\text{sim}}} \sum_{l=1}^{N_{\text{sim}}} \hat{\theta}_i^{k,l} \ . \tag{3.15}$$

This in turn is used to calculate the variance of $\langle \theta_i \rangle^k$, $\text{Var}\left(\langle \theta_i \rangle^k\right)$ as:

$$\text{Var}\left(\langle \theta_i \rangle^k\right) = \langle \theta^2 \rangle^k - \left(\langle \theta_i \rangle^k\right)^2 , \tag{3.16}$$

which is used to define the convergence error:

$$\tilde{\varepsilon}_i^k = \sqrt{\text{Var}\left(\langle \theta_i \rangle^k\right)} \ . \tag{3.17}$$

In order to more accurately calculate statistical properties of $\langle \theta_i \rangle^k$ and to obtain its distribution function, a statistical bootstrap method [100] is used with N_b bootstrap samples. This amounts to randomly selecting N_b times, allowing for repetitions, N_{sim} simulations out of the N_{sim} that were actually run. The value of N_b is chosen large enough so as to yield statistical results for $\langle \theta_i^k \rangle$ that are independent of N_b.

3.2.4.2 Numerical Error

Numerical error $\bar{\varepsilon}_i$ for estimator θ_i is defined for a given simulation conducted with Δt and W_p as the difference between $\theta_i\left(W_p, \Delta t\right)$ and that obtained in the absence of any scaling:

$$\overline{\varepsilon}_i = \theta_i\left(W_p, \Delta t\right) - \lim_{W_p \to 1, \Delta t \to 0} \theta_i\left(W_p, \Delta t\right). \tag{3.18}$$

The estimator value in the absence of scaling cannot be obtained in practice so that it must be approximated by the value obtained with a simulation conducted with parameters $W_{p,0}$ and Δt_0 such that $W_{p,0} \ll W_p$ and $\Delta t_0 \ll \Delta t$ so that:

$$|\theta_i\left(W_{p,0}, \Delta t_0\right) - \lim_{W_p \to 1, \Delta t \to 0} \theta_i\left(W_p, \Delta t\right)| \ll |\theta_i\left(W_p, \Delta t\right) - \theta_i\left(W_{p,0}, \Delta t_0\right)|. \tag{3.19}$$

Using this approximation of the exact solution, Eq. (4.3) can be rewritten as:

$$\overline{\varepsilon}_i \approx \theta_i\left(W_p, \Delta t\right) - \theta_i\left(W_{p,0}, \Delta t_0\right). \tag{3.20}$$

3.3 Analysis and consequences of the correlation between samples

3.3.1 The Central Limit Theorem

Following the definition of Eq. (3.17), the convergence error for the sampled mean $\hat{\mu}_1^k\left(y\right)$ is due to the variance of $\hat{\mu}_1^k\left(y\right)$ over multiple simulations. In the case where $\left\{y^{k'}\right\}_{k'=1}^{k}$ are independent and identically distributed (iid), the central limit theorem (CLT) [58] states that

$$\sqrt{k}\left(\hat{\mu}_1^k\left(y\right) - \mu\left(Y\right)\right) \xrightarrow{\mathcal{L}} \mathcal{N}\left(0, \sigma^2\left(Y\right)\right) \tag{3.21}$$

which implies that for $k \gg 1$

$$\langle \hat{\mu}_1^k\left(y\right) \rangle \sim \mathcal{N}\left(\mu_1\left(Y\right), \frac{\sigma^2\left(Y\right)}{k}\right) \tag{3.22}$$

so that by definition for $k \gg 1$:

$$\tilde{\varepsilon}^k = \frac{\sqrt{\sigma^2(Y)}}{k^{1/2}} \approx \frac{\sqrt{\hat{\sigma}^2(y)}}{k^{1/2}}. \tag{3.23}$$

Eq. (3.23) gives an expression for the evolution of the convergence error that is valid if Y is an iid variable. This is, however, not generally true for most quantities in a DSMC simulation. N_i^{k+1} is for instance not independent of N_i^k but rather correlated with it as the movement of particles in and out of cell i is quite limited between time steps. This correlation furthermore extends to previous time steps i.e. $N^{k-1}, N^{k-2}, \ldots$ etc. Instantaneous quantities in each cell of the simulation Y_i^k can thus be viewed as discrete correlated time series.

3.3.2 Correlation measure

3.3.2.1 Time correlation of cell-based quantities

The central limit theorem outlined above is derived under the assumption that random variable Y is independent and identically distributed. It thus cannot be applied in its strongest form to a time-correlated time series. This is turn means that the time correlation of sampled quantities needs to be quantified. The autocovariance function γ_Y of Y_i at time k_0 is defined as:

$$\gamma_Y(k_0 + k, k_0) \triangleq \mathrm{Cov}\left(Y^{k_0}, Y^{k_0+k}\right) \tag{3.24}$$

(dropping the i subscript for conciseness) with:

$$\mathrm{Cov}\left(Y^{k_0}, Y^{k_0+k}\right) = \mathbb{E}\left[\left(Y^{k_0+k} - \mathbb{E}\left[Y^{k_0+k}\right]\right)\left(Y^{k_0} - \mathbb{E}\left[Y^{k_0}\right]\right)\right] \tag{3.25}$$

For stationary processes (such as instantaneous simulation quantities in the steady state regime), the autocovariance function is independent of time k_0 and only a function of the time difference k:

$$\gamma_Y\left(k\right) = \gamma_Y\left(k, 0\right). \tag{3.26}$$

The autocorrelation function (ACF) of Y^k is defined as:

$$\rho_Y\left(k\right) = \frac{\gamma_Y\left(k\right)}{\gamma_Y\left(0\right)}. \tag{3.27}$$

Its calculation is quite straightforward in practice and only requires the storage of y for the previous k steps.

3.3.2.2 Correlation of particle-based quantities

The calculation of the time correlation of particle-based variables (such as velocities) inside a cell is more complicated. This is because, at any time step, a cell contains multiple particles. These in turn can be correlated with one another. This occurs for instance in cells which contain few particles on average, i.e. when $\overline{N} \sim 1$ where repeated collisions between the same two particles [134] can lead to their velocities being correlated. This type of correlation will not be apparent when the measure defined in the previous section is used. For this reason, we distinguish between the *intra-step* and the *inter-step* correlation for the cell average of a particle-based variable. The analysis in the following will be presented for one velocity component V but is general enough to be readily applicable to any particle-based variable such as kinetic energy or internal energy by simple variable substitution. The sampled mean of velocity component V is defined at time k by:

$$\hat{\mu}^k\left(V\right) = \frac{1}{N_{\text{tot}}^k} \sum_{k'=1}^{k} \sum_{j=1}^{N^{k'}} V_j^{k'}, \tag{3.28}$$

where $V_j^{k'}$ is the velocity of particle i at step k' and $N_{\text{tot}}^k = \sum_{k'=1}^k N^{k'}$ is the total number of particles that the cell has contained up until time step k. The variance of the $\sum_{i=1}^k \sum_{j=1}^{N^k} V_i^k$ sum of Eq. (3.28) can be calculated as follows:

$$\operatorname{Var}\left(\sum_{k'=1}^{k}\sum_{j=1}^{N^{k'}}V_j^{k'}\right)$$

$$=$$

$$\begin{array}{rcl}
\operatorname{Var}\left(V_1^1\right)+ & ... & +\operatorname{Var}\left(V_{N^1}^1\right)+\\
 & ... & \\
\operatorname{Var}\left(V_1^k\right)+ & ... & +\operatorname{Var}\left(V_{N^k}^k\right)+\\
2\ \operatorname{Cov}\left(V_1^1,V_2^1\right)+2\ \operatorname{Cov}\left(V_1^1,V_3^1\right)+...+ & 2\ \operatorname{Cov}\left(V_1^1,V_{N^1-1}^1\right) & +2\ \operatorname{Cov}\left(V_1^1,V_{N^1}^1\right)+\\
2\ \operatorname{Cov}\left(V_2^1,V_3^1\right)+...+ & 2\ \operatorname{Cov}\left(V_2^1,V_{N^1-1}^1\right)+ & 2\ \operatorname{Cov}\left(V_2^1,V_{N^1}^1\right)+\\
 & ... & \\
+ & ... & +2\ \operatorname{Cov}\left(V_{N^1-1}^1,V_{N^1}^1\right)+\\
 & ... & \\
2\ \operatorname{Cov}\left(V_1^k,V_2^k\right)+2\ \operatorname{Cov}\left(V_k^k,V_3^k\right)+...+ & 2\ \operatorname{Cov}\left(V_1^k,V_{N^k-1}^k\right) & +2\ \operatorname{Cov}\left(V_1^k,V_{N^k}^k\right)+\\
2\ \operatorname{Cov}\left(V_2^k,V_3^k\right)+...+ & 2\ \operatorname{Cov}\left(V_2^k,V_{N^k-1}^k\right) & +2\ \operatorname{Cov}\left(V_2^k,V_{N^k}^k\right)+\\
 & ... & \\
+ & ... & +2\ \operatorname{Cov}\left(V_{N^k-1}^k,V_{N^k}^k\right)+\\
\operatorname{Cov}\left(V_1^1,\displaystyle\sum_{j=1}^{N^2}V_j^2\right)+ & ... & +\operatorname{Cov}\left(V_1^1,\displaystyle\sum_{j=1}^{N^k}V_j^k\right)+\\
 & ... & \\
\operatorname{Cov}\left(V_{N^1}^1,\displaystyle\sum_{j=1}^{N^2}V_j^2\right)+ & ... & +\operatorname{Cov}\left(V_{N^1}^1,\displaystyle\sum_{j=1}^{N^k}V_j^k\right)+\\
\operatorname{Cov}\left(V_1^2,\displaystyle\sum_{j=1}^{N^1}V_j^1\right)+ & ... & +\operatorname{Cov}\left(V_1^2,\displaystyle\sum_{j=1}^{N^k}V_j^k\right)+\\
 & ... & \\
\operatorname{Cov}\left(V_{N^k}^k,\displaystyle\sum_{j=1}^{N^1}V_j^1\right)+ & ... & +\operatorname{Cov}\left(V_{N^k}^k,\displaystyle\sum_{j=1}^{N^{k-1}}V_j^{k-1}\right)
\end{array}$$

The intra-step correlation is represented in the preceding equation by the sum of particle covariances at the same time step and can be written as:

$$\gamma_V^{\text{intra},k} = \sum_{p=1}^{N^k} \sum_{\substack{q=1 \\ q\neq p}}^{N^k} \text{Cov}\left(V_p^k, V_q^k\right) = \sum_{q=1}^{N^k} \sum_{p=i+1}^{N^k} 2\ \text{Cov}\left(V_p^k, V_q^k\right). \tag{3.29}$$

It involves calculating the correlation between $N_{\text{pairs}}^k = \frac{N^k}{2}\left(N^k - 1\right)$ pairs of particles at each time step. If the DSMC simulation is statistically stationary, each time step has the same statistical properties and an average intra-step correlation coefficient $\overline{\gamma}_V^k$ can be defined:

$$\overline{\gamma}_V^{\text{intra},k} = \frac{\sum_{k'=1}^{k} \gamma_V^{\text{intra},k'}}{\sum_{k'=1}^{k} N_{\text{pairs}}^{k'}} \tag{3.30}$$

which can be used to sample for $\overline{\gamma}_V^{\text{intra},k}$.

The correlation between time steps is represented by the $\text{Cov}\left(V, \sum\right)$ terms. For conciseness, in the following the sum of all particle velocities at time step k' will be abbreviated as $\Sigma^{k'} \equiv \sum_{j=1}^{N^{k'}} V_j^{k'}$. Using the bilinear property of the covariance operator, neglecting the covariance between particles inside a cell for time step differences greater than m (an assumption which will be later checked) and assuming stationary statistical properties, it can be simplified as follows to give an expression for the inter-step correlation of velocity in a cell:

$$\gamma_V^{\text{inter},k} = \sum_{j=1}^{m} 2\ \text{Cov}\left(\Sigma^k, \Sigma^{1\vee k-j}\right) \tag{3.31}$$

with $1 \vee i - j \triangleq \text{Min}\left(1, k-j\right)$. As previously, an average inter-step correlation coefficient $\overline{\gamma}_V^{\text{inter},k}$ can be derived:

$$\gamma_V^{\text{inter},k} = \frac{\sum_{k'=1}^{k} \gamma_V^{\text{inter},k'}}{k'} \tag{3.32}$$

which allows the sampling of a mean value.

3.3.3 A modified version of the central limit theorem

The estimator for $\mu_1(y)$ at time step k, $\hat{\mu}_1^k(y)$ is defined as:

$$\hat{\mu}_1^k(y) = \frac{1}{k} \sum_{k'=1}^{k} y^{k'}. \tag{3.33}$$

A central limit theorem for $\hat{\mu}_1^k(y)$ that takes into account the time correlation of y^k can be derived [135] [59] by taking the variance of Eq. (3.33):

$$\mathrm{Var}\left(\hat{\mu}_1^k(y)\right) = \frac{1}{k^2} \sum_{k'=1}^{k} \sum_{k''=1}^{k} \mathrm{Cov}\left(y^{k'}, y^{k''}\right). \tag{3.34}$$

From Eq. (3.26), $\gamma_Y(k'' - k') = \mathrm{Cov}\left(y^{k'}, y^{k''}\right)$ and using the symmetry of the $(k \times k)$ covariance matrix to simplify the double sum:

$$\mathrm{Var}\left(\hat{\mu}_1^k(y)\right) = \frac{1}{k^2} \left(k\ \gamma_Y(0) + 2 \sum_{k'''=1}^{k-1} (k - k''')\, \gamma_Y(k''')\right). \tag{3.35}$$

The variance $\mathrm{Var}\left(\hat{\mu}_1^k(y)\right)$ will be referred to in the following by $\tilde{\sigma}_y^{2,k}$. The effect of the time correlation of $\left\{y^k\right\}$ is to increase the variance of $\hat{\mu}_1^k(y)$, as $\tilde{\sigma}_y^{2,k}$ is greater than the value of the variance expected in the case of independent samples, i.e. $\frac{\gamma(0)}{k}$. It can further be proven [135] that when $\left\{y^k\right\}$ is a Gaussian time series:

$$\hat{\mu}_1^k(y) \sim \mathcal{N}\left(\mu_1^k(y), \frac{\tilde{\sigma}_y^{2,k}}{k}\right) \tag{3.36}$$

which constitutes a modified central limit theorem law applicable to a correlated normally distributed time series. The convergence result does hold when $\left\{y^k\right\}$ is non-Gaussian while the convergence in distribution to a normal law is subject to other conditions beyond the scope of the present work. The most important result for the following is the value of the variance provided by Eq. (3.35), not the distribution of $\hat{\mu}_1^k(y)$. Within the framework of the present study $\hat{\mu}_1^k(y) \sim \mathcal{N}\left(\mu_1^k(y), \frac{\tilde{\sigma}_y^{2,k}}{k}\right)$ has systematically been observed. A non-normal distribution limit can, however, not be excluded in all generality. Eq. (3.35) thus allows the calculation of the convergence error, defined by Eq. (3.17), for $\hat{\mu}_1^k(y)$ so long as its autocorrelation function γ_Y is known.

3.3.4 Variance of sampled velocity

In this section, an expression for the variance of $\hat{\mu}_1^k(\mathbf{y})$ is derived where $\mathbf{y}$ is a cell-based variable that has been obtained via Eq. (3.4) from a particle-based variable. To distinguish results in this section from those in the previous, y will denoted by V which could for instance designate a velocity component such as V_x. Following Eq. (3.10), the estimator $\hat{\mu}_1^k(V)$ of the mean $\mu_1^k(V)$ of V is obtained with:

$$\hat{\mu}_1^k(V) = \frac{\sum_{k'=1}^{k}\sum_{j=1}^{N^{k'}} V_j^{k'}}{\sum_{k'=1}^{k} N^{k'}} = \frac{\hat{\mu}_1^k(\Sigma V)}{\hat{\mu}_1^k(N)}, \tag{3.37}$$

which is the ratio of two mean estimators where random variable ΣV is defined as $\Sigma V^k \triangleq \sum_{j=1}^{N^k} V_j^k$. It cannot be put into the form of a single sum of a random variable so as to make it amenable to the central limit theorem. That is, there exists no readily obtainable random variable Ξ in the simulation such that

$$\hat{\mu}_1^k(V) \equiv \frac{1}{k}\sum_{k'=1}^{k} \xi^{k'}. \tag{3.38}$$

As previously discussed, $\hat{\mu}_1^k(\Sigma V) \sim \mathcal{N}\left(\mu_1^k(\Sigma V), \frac{\tilde{\sigma}_{\Sigma V}^{2,k}}{k}\right)$ and $\hat{\mu}_1^k(N) \sim \mathcal{N}\left(\mu_1^k(N), \frac{\tilde{\sigma}_N^{2,k}}{k}\right)$ so that Eq. (3.37) indicates that $\hat{\mu}_1^k(V)$ is equal to the ratio of two normal distributions. Random variables ΣV and N are, however, correlated with one another much like each is time-correlated with itself. This time-dependent cross-correlation can be estimated as follows:

$$\mathrm{Cov}\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right) = \frac{1}{k^2}\sum_{k'=1}^{k}\sum_{k''=1}^{k}\mathrm{Cov}\left(\Sigma V^{k'}, N^{k''}\right), \tag{3.39}$$

where, following the same approach as for Eq. (3.35), the double sum can be simplified so as to yield:

$$\begin{aligned}\mathrm{Cov}\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right) &\approx \frac{\gamma_{\Sigma V,N}\left(0\right)}{k} + \frac{2}{k}\sum_{\substack{k'''=-K\\k'''\neq 0}}^{K}\gamma_{\Sigma V,N}\left(k'''\right)\\ &= \frac{\sigma^2\left(\Sigma V, N\right)}{k} + \frac{2}{k}\sum_{\substack{k'''=-K\\k'''\neq 0}}^{K}\gamma_{\Sigma V,N}\left(k'''\right) \triangleq \frac{\tilde{\sigma}^2\left(\Sigma V, N\right)}{k},\end{aligned} \tag{3.40}$$

where

$$\gamma_{\Sigma V,N}\left(k'''\right) \triangleq \mathrm{Cov}\left(\Sigma V^{k'}, N^{k'+k'''}\right),\ \forall k' \tag{3.41}$$

as $\left\{\Sigma V^k\right\}$ and $\left\{N^k\right\}$ are stationary processes. Normalized correlations $\rho_{\Sigma V,N}\left(k\right)$ and $\rho\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right)$ are defined as:

$$\rho_{\Sigma V,N}\left(k\right) \triangleq \frac{\mathrm{Cov}\left(\Sigma V^{k'}, N^{k'+k}\right)}{\sigma_{\Sigma V}\ \sigma_N}\ \forall k' \tag{3.42}$$

and

$$\rho\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right) \triangleq \frac{\mathrm{Cov}\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right)}{\tilde{\sigma}_{\Sigma V}^k\ \tilde{\sigma}_N^k}. \tag{3.43}$$

Eq. (3.40) shows that when $k \gg 1$, $\mathrm{Cov}\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right) \mathcal{O}\left(1/k\right)$. The two estimators thus become increasingly uncorrelated as $k \to \infty$. Having determined the correlation $\rho\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right)$ and the normal distribution functions of $\hat{\mu}_1^k\left(\Sigma V\right)$ and $\hat{\mu}_1^k\left(N\right)$ being known, the exact distribution function of their ratio as defined by Eq. (3.37) can be determined following the approach presented in [37]. The exact distribution function of the ratio could then in principle be used to calculate its variance. The analytical expression of the distribution function presented in [37] is, however, so complicated that the calculation of its variance, even numerically, is difficult. An approximation of the variance of the ratio can, however, be obtained with the delta method [36] as follows:

$$\mathrm{Var}\left[\frac{\hat{\mu}_1^k\left(\Sigma V\right)}{\hat{\mu}_1^k\left(N\right)}\right] \approx \frac{\hat{\mu}_1^2\left(\Sigma V\right)}{\hat{\mu}_1^4\left(N\right)}\tilde{\sigma}_N^{2,k} + \frac{1}{\hat{\mu}_1^2\left(N\right)}\tilde{\sigma}_{\Sigma V}^{2,k} - \frac{2\ \hat{\mu}_1\left(\Sigma V\right)}{\hat{\mu}_1^3\left(N\right)}\rho\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right)\tilde{\sigma}_N^k\tilde{\sigma}_{\Sigma V}^k. \tag{3.44}$$

Eq. (3.44) thus provides an expression for the variance of the sampled mean velocity $\hat{\mu}_1^k\left(V\right)$ which, by the definition of Eq. (3.17), is also the convergence error $\tilde{\varepsilon}^k\left(V\right)$. In Eq. (3.44), $\tilde{\sigma}_N^{2,k}$ and $\tilde{\sigma}_{\Sigma V}^{2,k}$ are both evaluated with Eq. (3.35) which is respectively applied to N and ΣV. $\rho\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right)$ is calculated with Eq. (3.43) where $\mathrm{Cov}\left(\hat{\mu}_1^k\left(\Sigma V\right), \hat{\mu}_1^k\left(N\right)\right)$ is obtained from Eq. (3.40). The determination of the convergence error for V thus requires the calculation of the autocorrelation function of N, ΣV and the correlation function of N and ΣV.

3.4 Testcase

3.4.1 Geometry and flow conditions

The domain used in the simulation is shown in Fig. 3.1. A jet with an inlet of radius 0.001 m is located at $x = 0$[1]. The inlet injects argon with number density $8.85 \times 10^{22}\mathrm{m}^{-3}$ in equilibrium at a temperature of 750 K with a bulk velocity in the x direction of 510 m s^{-1}. This corresponds to Mach number M = 1 and Knudsen number Kn = 0.01 (based on the inlet radius). A supersonic outflow boundary condition is used for the top and right sides of the domain while a *de facto* symmetry condition along the centerline is enforced by the axisymmetric move procedure [15].

---

[1] The entire mesh is offset by $\Delta x = 10^{-6}$ m to avoid errors near $x = 0$ when running simulations.

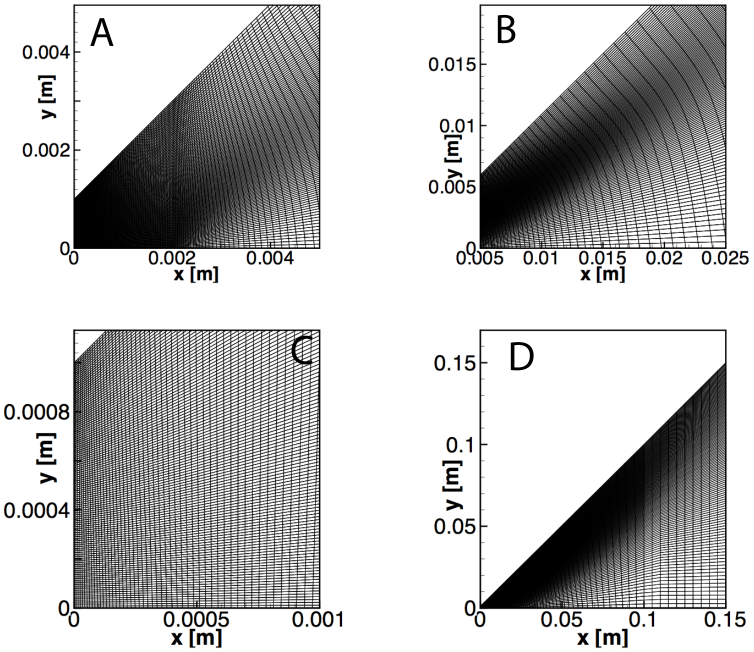


Figure 3.1: View of the computational domain and mesh. A, B: domain views, C: closeup view of the inlet and D: overall domain view

3.4.2 Computational Parameters

Details concerning the implementation of the DSMC procedure can be found in section 3.2.1. The different test cases that are run during the course of this study are summarized in Tables 3.1 which indicates the corresponding Δt and W_p values that are used. The time step does satisfy the DSMC requirement that it be less than the mean collision time for all test cases as $\left(\frac{\Delta t}{\tau_{\text{mct}}}\right)_{\max} = 0.159$ for $\Delta t = 5.0 \times 10^{-9}$ [s] in the highest density zone located in front of the inlet. The DSMC mean free path requirement is also satisfied as $\left(\frac{\Delta x}{\lambda}\right)_{\max} = 0.67$ in the cells around $(x, y) = (0.008, 0.0015)$. A view of the average number of particles obtained for all cells for test case C64 is shown in Fig. 3.2. The test cases will be referred to by the name indicated in the table in the following. Samples are always taken at every time step once the simulation was considered to have reached steady state. Steady state was assumed to occur after 100,000 time steps. Based on the bulk x-velocity of particles,

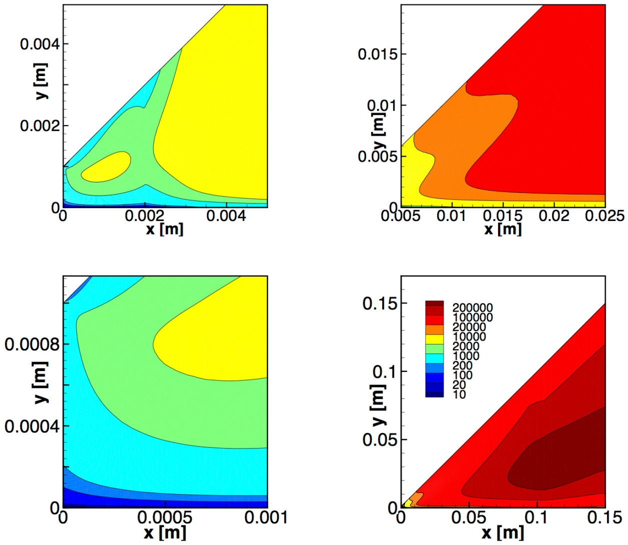


Figure 3.2: Average number of particles in each cell for test case C64. The cell with the smallest average number of particles is located at the corner of the centerline and the inlet and contains 0.3.

this corresponds to about 5 times the mean residence time of particles in the domain. For the baseline time step, the total number of particles in the domain is observed to remain approximately constant after 40,000 time steps while the total number of collisions per time step does so after 10,000. In order to gain greater insight into the simulation, instantaneous DSMC fields are examined in detail by outputting them at every time step for a few selected cells of the domain shown in Fig. 3.3.

Test Case	**Number of particles (M)**	$\mathbf{W_p}$	Δt [s]
C256	134.0	1.4×10^8	5.0×10^{-9}
C128	67.0	2.8×10^8	5.0×10^{-9}
C64200	33.5	5.6×10^8	1.0×10^{-8}
C64	33.5	5.6×10^8	5.0×10^{-9}
C64050	33.5	5.6×10^8	2.5×10^{-9}
C32	16.7	11.2×10^8	5.0×10^{-9}
C16	8.4	22.4×10^8	5.0×10^{-9}

Table 3.1: Numerical parameters of test cases

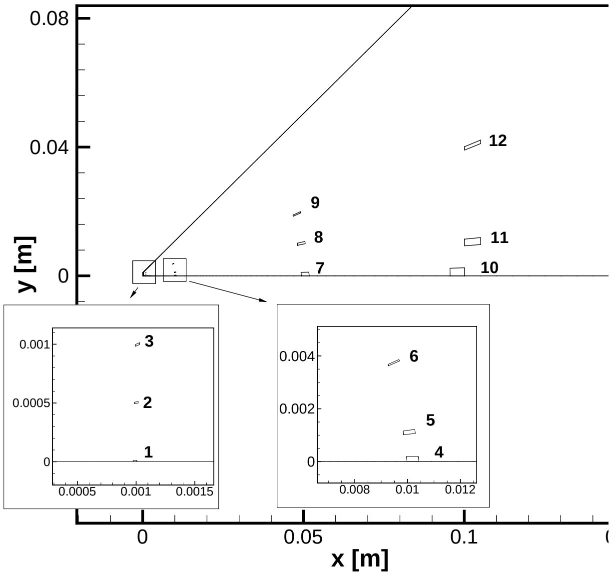


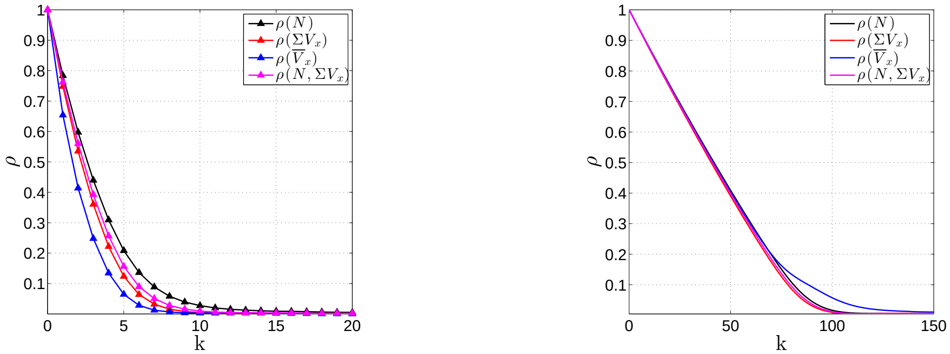
Figure 3.3: Location and numbering of cells studied in detail.

3.5 Convergence and correlation of sampled quantities

In this section, the time correlation of sampled quantities is studied within the framework of test case C64. The prediction of convergence error obtained is then checked against numerical observations for a selected number of cells. A general procedure for the a priori prediction of convergence error is then outlined.

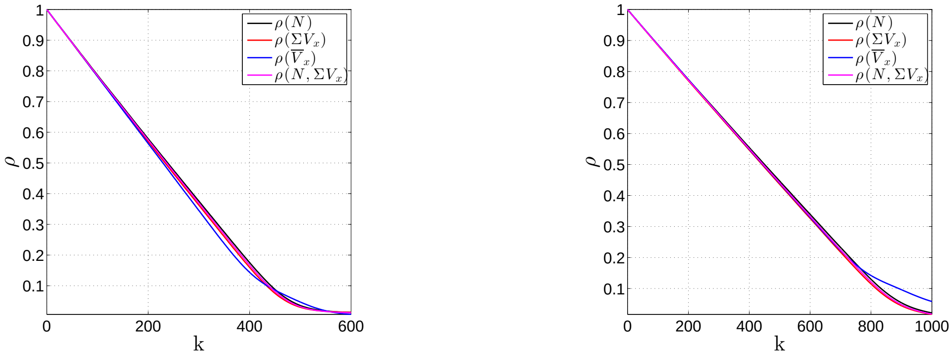
3.5.1 Time correlation between samples

The time correlation between sampled N^k, ΣV^k and $\overline{V}^k \triangleq \frac{\Sigma V^k}{N^k}$ can be quantified by evaluating their autocorrelation function as outlined in section 3.3.2.1. Figures 3.4(a) to 3.4(d) show $\rho_N(k)$, $\rho_{\Sigma V_x}(k)$, $\rho_{\overline{V}_x}(k)$ and $\rho_{\Sigma V,N}(k)$ for a few cells of the domain. For all the individual cells that are studied, N, ΣV_x and $\overline{V}$ are observed to have very similar (auto)correlation functions that are very well described with an exponential law i.e. $\rho(k) = 1 - \exp\left(-\frac{k}{\tau}\right)$ with a characteristic time τ that is a measure of the extent of the time correlation. Relatively small differences between autocorrelation functions are visible for cells close to the origin characterized by $\tau \sim 10$ whereas the functions are virtually identical further out in the domain for cells with $\tau \sim 100$. Large spatial variations in the extent of the time correlation of N are observed throughout the domain. This is visible in Figs. 3.5, 3.6 and 3.7 that display respectively $\rho_n(k = 1)$, $\rho_n(k = 10)$ and $\rho_n(k = 30)$. The large correlation increase in the downstream portion of the jet is caused by the use of an inappropriately small step in that portion of the domain. This is due to the use of a uniform time step for the entire domain that is dictated by the condition that it be less than the mean collision time in the densest part of the domain located in the cells adjacent to the inlet. The inadequacy of the time step value is further compounded by the acceleration of particles caused by the expansion of the jet and the decrease in the number of collisions in the downstream direction.



(a) Cell 1

(b) Cell 4



(c) Cell 7

(d) Cell 10

Figure 3.4: Autocorrelation and correlation functions for the instantaneous velocity and number density as a function of the time step

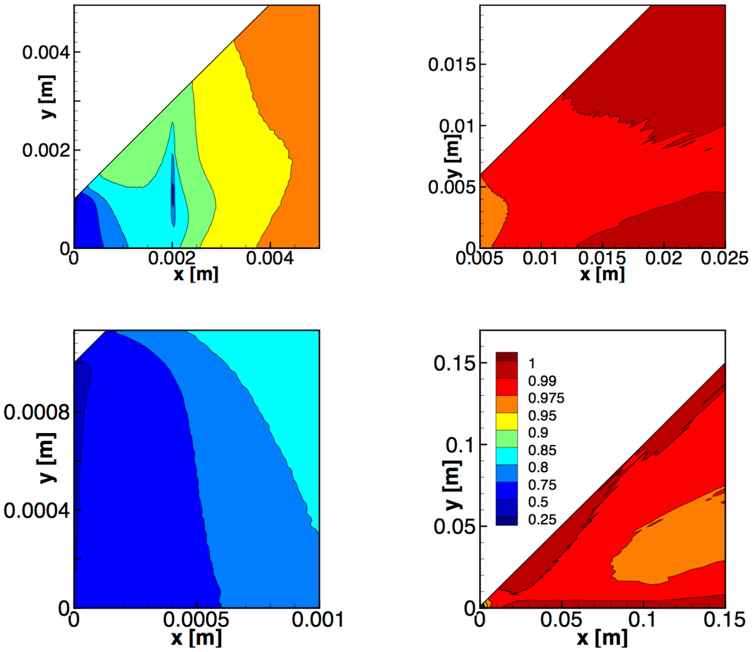


Figure 3.5: Spatial view of $\rho_N(k=1)$ for test case C64

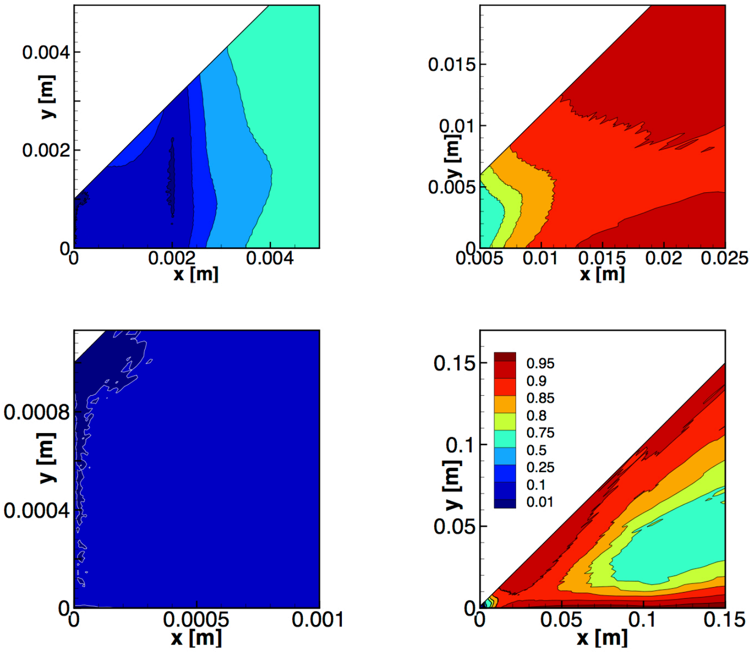


Figure 3.6: Spatial view of $\rho_N(k=10)$ for test case C64

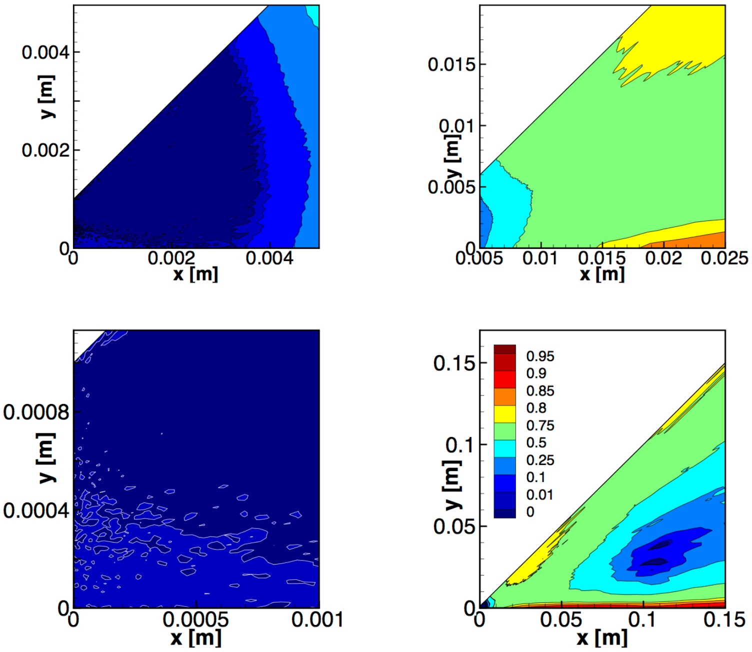
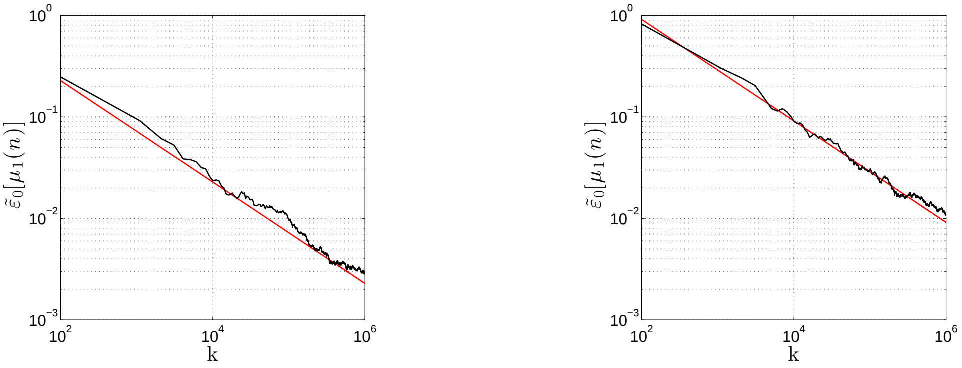


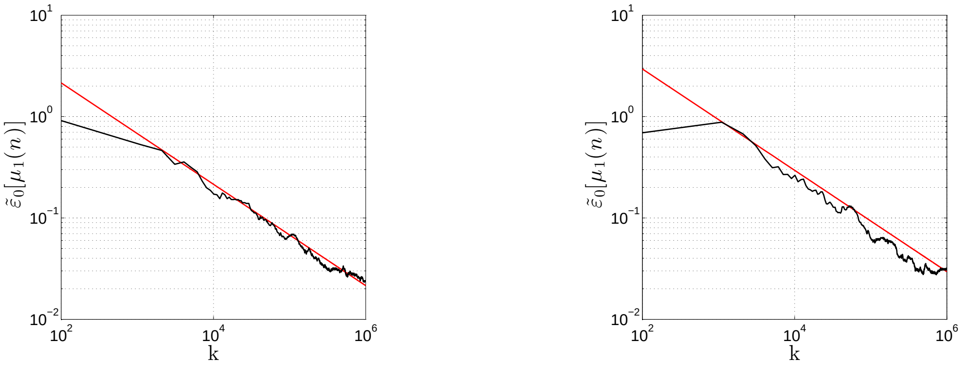
Figure 3.7: Spatial view of $\rho_N(k = 30)$ for test case C64

3.5.2 Convergence study for $\tilde{\varepsilon}_0 \left[\mu_1^k (n)\right]$

To obtain results for the convergence of the sampled mean for number density, the convergence error defined by Eq. (3.17) is calculated by running $N_{sim} = 22$ distinct simulations for the C64 test case. 40×10^3 bootstrap samples are then used to calculate the standard deviation of the ensemble average of sampled number density. The standard deviation of the ensemble average is then normalized by the standard deviation of the number density σ_n in the same cell and plotted in black against the number of time steps in a log-log plot. The plots are shown in Fig. 3.8 for 4 representative cells shown in Fig. 3.3. For each cell, the autocorrelation functions for N that were previously shown are used to calculate the predicted normalized standard deviation with Eq. (3.35). The prediction is plotted in red in Fig. 3.8. If the central limit theorem were to hold perfectly, the ensemble average of the standard deviation would be $\sigma_n \times k^{-0.5}$ (for $k \gg 1$). This is because the central limit theorem states that $\hat{\mu}_1^k (n) \sim \mathcal{N}(\mu_n, \sigma_n^2/k)$ if samples are independent and identically distributed. In this case the ensemble average would of course have a standard deviation of $\frac{\sigma_n}{k^{1/2}}$ so that the normalized standard deviation of the ensemble average would be $k^{-1/2}$. The increase in standard deviation predicted by Eq. (3.35) seems to describe very well that observed in practice for all cells as shown in Fig 3.8.



(a) Cell 1 (b) Cell 4



(c) Cell 7 (d) Cell 10

Figure 3.8: Observed convergence error for sampled number density (black) and theoretical prediction (red).

3.5.3 Convergence study for $\tilde{\varepsilon}_0\left[\mu_1^k\left(V_x\right)\right]$

The same procedure is applied to the study of the convergence of the sampled mean of V_x and results are shown in Fig. 3.9. The predicted normalized variance of Eq. (3.44) is plotted in red in the same plots for each individual cell. Very good agreement between the predicted and observed variance is observed for all cells.

3.5.4 Implementation considerations of the a priori determination of convergence error

The use of a limited number of samples to evaluate statistical properties in a DSMC simulation naturally introduces uncertainty in their estimated value. That uncertainty takes the form of the convergence error defined by Eq. (3.17) that, as previously seen, can be determined by Eq. (3.35) or (3.44) so long as the autocorrelation function of the variable being sampled for is known (in addition to the cross correlation with N in the case of Eq. (3.44)). By the definition of Eq. (3.25), determining the autocorrelation function up to order k, i.e. $\rho\left(k\right)$, in turn requires the storage of the value of the variable at the k preceding time steps as well as the storage of the sampled autocorrelation function itself. Thus the calculation of the autocorrelation function for a variable up until order k requires the storage of $2 \times k$ values per cell. The cost of the calculation is furthermore negligible compared to that of the simulation. For zones with a high level of correlation, where the decay of the autocorrelation with k is slow such as for instance in Fig. 3.4(d), the value of the autocorrelation function for large k values can be interpolated so that only the value of the autocorrelation for low k values (up until 20 perhaps) needs to be calculated and stored. The difference between the value of the apparent standard deviation obtained with the full and interpolated spectrum is expected to be minimal. Because of the great similarities between the autocorrelation of the number of particles, the total velocity, the mean velocity and the cross correlation, only one correlation function must furthermore be calculated and stored instead of four. More practical implementation considerations are, however, outside the scope of the present work except to add that the computational and storage requirement of the implementation of an a priori determination of the value of the convergence error via Eq. (3.35) are small compared to those of the underlying simulation.

3.5.5 Velocity correlation between particles

The time correlation between sampled quantities does not affect the ultimate (converged) value of statistical estimators. It only acts to slow down convergence compared to an idealized situation where no time correlation exists. In contrast to that, consistent correlation between the velocities of particles inside a cell, for instance caused by repeated collisions, is a well known source of numerical error in DSMC simulations [134]. This is because it does not allow collisions to be properly modeled in cells where such a phenomenon is present. Figure 3.10 shows the normalized intra step correlation of V_x, which is defined as:

$$\overline{\gamma}^{\text{intra}}_{V_x,0} = \frac{\overline{\gamma}^{\text{intra}}_{V_x}}{\sigma^2\left(V_x\right)}, \tag{3.45}$$

where $\overline{\gamma}^{\text{intra}}_{V_x} = \lim_{k\to\infty} \overline{\gamma}^{\text{intra},k}_{V_x}$ which is given by Eq. (3.30). The value of the absolute value of the intrastep correlation is very close to 0 for most cells of the domain except for those located close to the inlet of the jet and for those along the centerline near the inlet. The relatively large level of correlation present close to the centerline can directly be attributed to repeated collisions and/or the use of very few particles as those very cells are characterized by a low average number of particles as seen in Fig. 3.2. The high levels of correlation alongside the exit plane of the jet might be attributable to the formulation of inlet boundary conditions.

3.6 Effect of numerical parameters on the convergence error

Having previously determined the general behavior of the convergence error for a baseline simulation, i.e. test case C64, the aim of the present section is to determine the influence of the number of particles and time step.

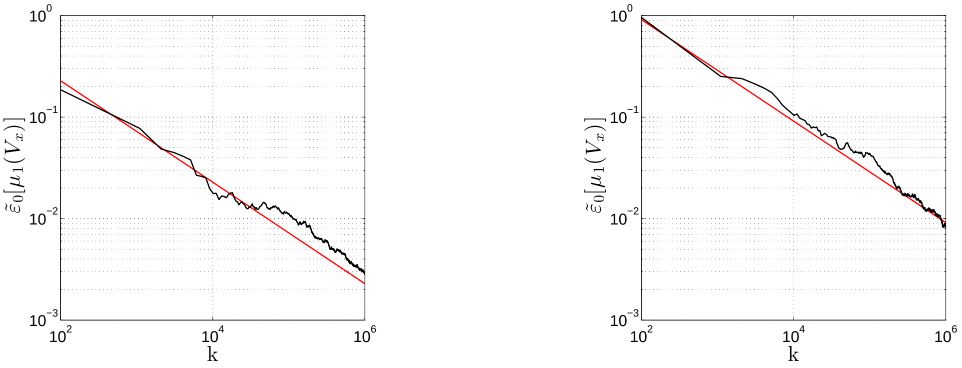
3.6.1 Influence of the number of particles on correlation and convergence

The impact of the number of particles in the simulation on the autocorrelation function is evaluated by calculating the autocorrelation function for test cases C16, C32, C64 and C128 for all cells of the simulation. Results obtained for selected cells are shown in Fig. 3.11 which shows that the autocorrelation function is unaffected by changes in the particle count. Identical results are obtained for all other cells that are examined. This indicates that the time correlation of samples, normalized by their variance, is independent of particle count. That is to say that, although the variance of N scales as W_p as shown by Eq. (3.12), the shape of its time correlation is maintained across particle counts.

To investigate the effect of the number of particles in the simulation on the rate of convergence of the convergence error, 10 simulations are run for each of the C16 and C256 cases in addition to the 20 simulations for C64. For each particle count, an ensemble average of the convergence error is calculated and is plotted in Fig. 3.12. For all particle counts, the same convergence rate for the error is observed. The discrepancies between the convergence paths are attributable to the relatively low number of simulations used to calculate the ensemble average of the convergence error. The fact that convergence is independent of particle count is a direct consequence of the invariance of the autocorrelation function.

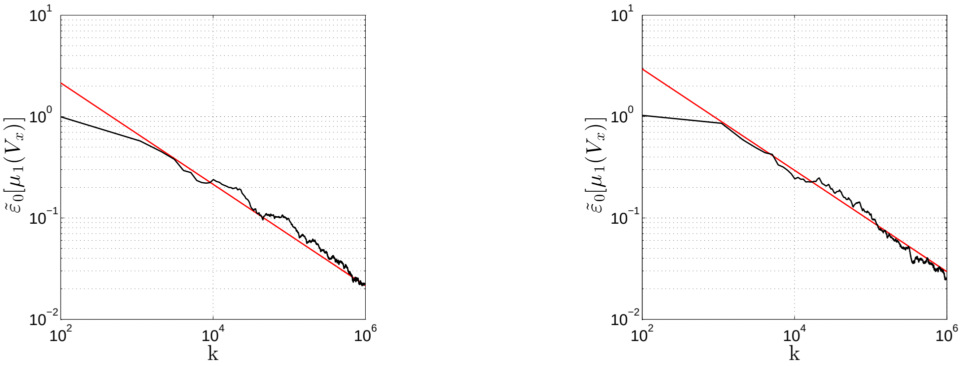
3.6.2 Influence of the time step on correlation and convergence

The effect of varying the time step on the correlation of samples and rate of convergence is now investigated. This is accomplished by running three simulations referred to in Table 3.1 by C64050, C64 and C64200 corresponding, respectively, to $\frac{\Delta t_0}{2}$, Δt_0 and $2\ \Delta t_0$. The autocorrelation function for N that is obtained for each test case is shown in Fig 3.13 as a function of $\tilde{k} = k\frac{\Delta t}{\Delta t_0}$. The autocorrelation functions are observed to collapse to the same curve under this change of variable. This indicates that $\rho_N\left(k, \Delta t\right) = \rho_{0,N}\left(k\ \Delta t\right)$ where $\rho_{0,N}$ is independent of both k and Δt. The invariance of the correlation function under this scaling quantifies the reduced or increased time correlation observed between samples when the time step is raised or lowered. The effect of the time step on the evolution of the convergence error is investigated by calculating its ensemble average for the three aforementioned test cases. The ensemble average is calculated with 10 simulations for both C64050 and C64200 while 20 simulations are used for C64. The evolution of the ensemble average as well as the prediction obtained from Eq. (3.35) by using the spectrum shown in Fig. 3.13 is shown for selected cells in Fig. 3.14. These plots clearly demonstrate the reduction in convergence error brought about by a decrease in the time correlation of samples caused by an increased time step value.



(a) Cell 1

(b) Cell 4



(c) Cell 7

(d) Cell 10

Figure 3.9: Observed convergence error for sampled V_x (black) and theoretical prediction (red).

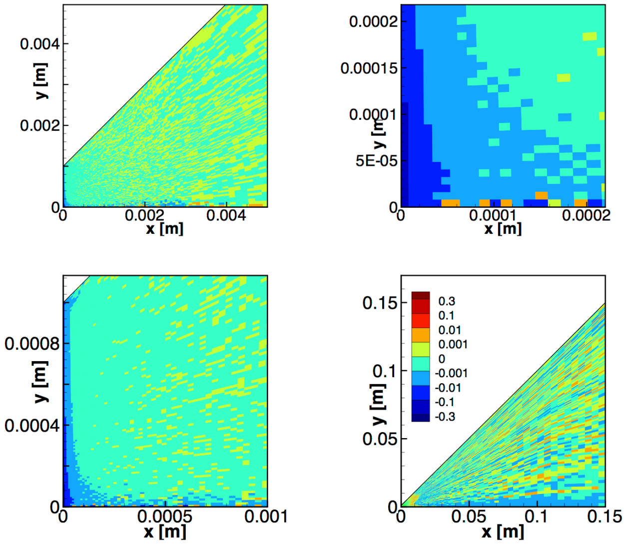
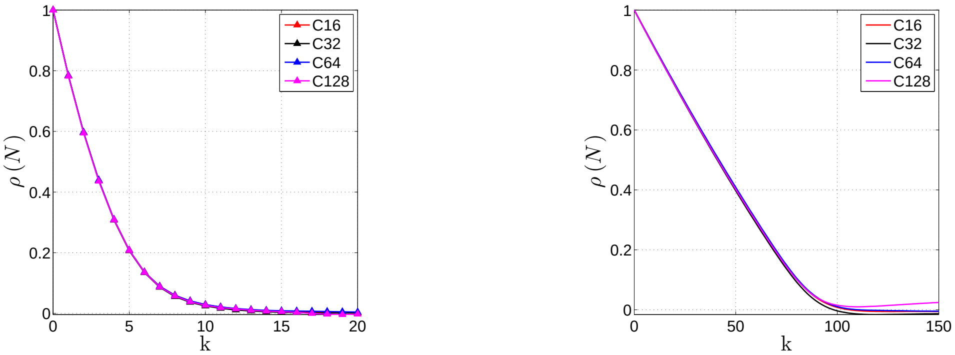
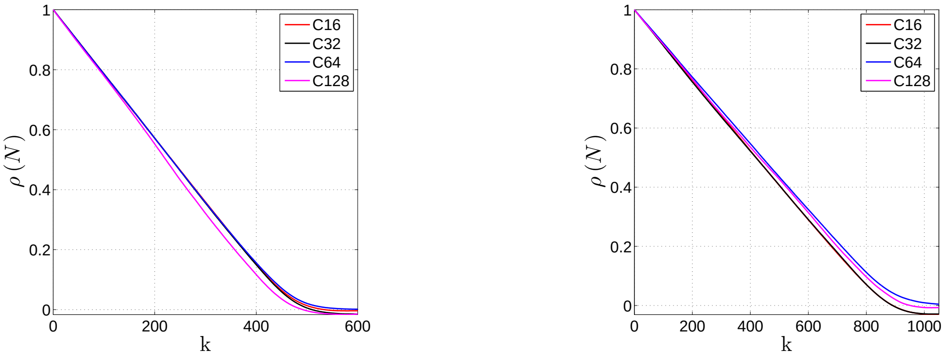


Figure 3.10: Normalized intrastep correlation coefficient for test case C64

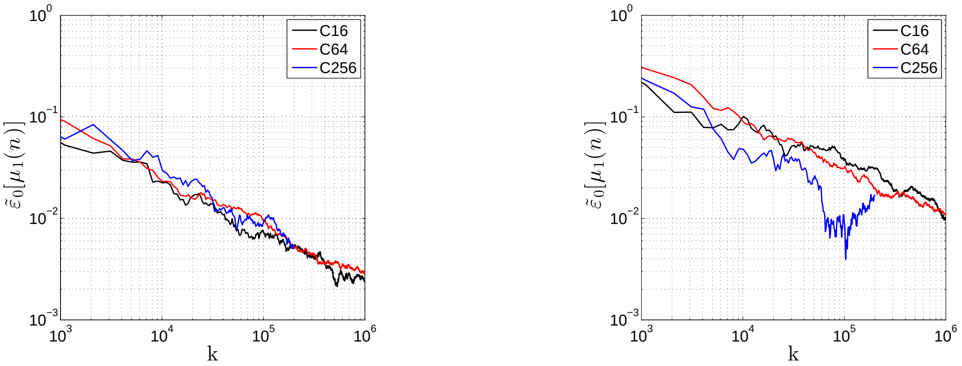


(a) Cell 1 (b) Cell 4



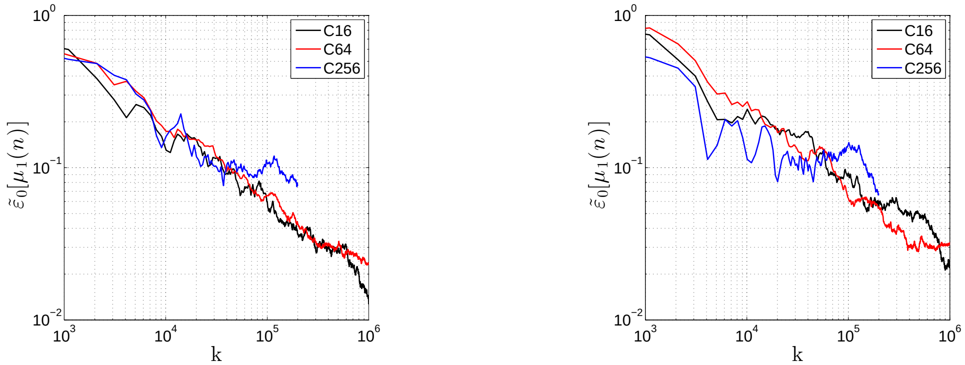
(c) Cell 7 (d) Cell 10

Figure 3.11: Influence of the particle count on the autocorrelation function for N.



(a) Cell 1

(b) Cell 4



(c) Cell 7

(d) Cell 10

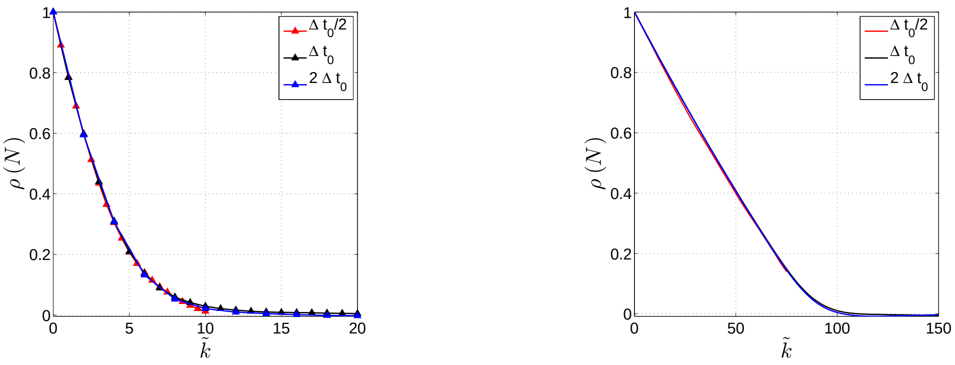
Figure 3.12: Convergence error for sampled n as a function of the number of samples for multiple particle counts.

3.7 Conclusions

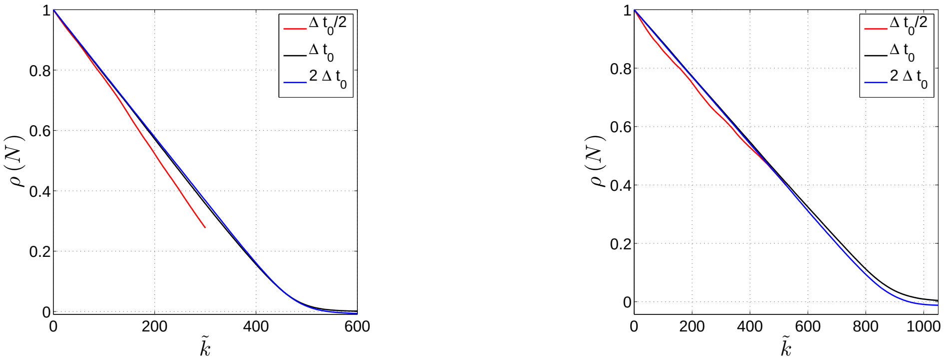
A framework for the analysis of the convergence and accuracy of a DSMC simulation was presented and two corresponding error types, the *convergence* and *numerical* error introduced.

The convergence error, the subject of this past chapter, was studied by taking ensemble averages of sampled means during their convergence. It was found to vary as $\tilde{\sigma}k^{-1/2}$ for the sampled mean of the number density and velocity. Time correlation between samples is such that the standard central limit theorem cannot be applied for sampled means. The autocorrelation function of sampled quantities was used to quantify the time correlation between samples. Particularly large time correlation was observed in zones where a small time step, relative to the mean collision time, was used. The autocorrelation function $\rho\left(k\right)$ was observed to be of very similar shape for N and V and very close to the correlation function between N and V . It was furthermore found to be invariant under particle count change and under $k\Delta t$ scaling. A modified version of the central limit theorem that takes correlation into account for both particle and cell-based variables (e.g; respectively n and V) was introduced. It was then used to formulate a numerical method to a priori predict the value of the convergence error during the course of the simulation. It was found to accurately predict the observed standard deviation $\tilde{\sigma}$ for all cells of the domain that were examined.

Such a method may be a useful tool to implement in a DSMC code to reduce the uncertainty associated with the lack of convergence of statistical properties. It would also reduce the burden of ensuring that sampled statistical properties have indeed converged for all cells in a simulation. A less storage intensive convergence error prediction scheme following the ideas that were previously outlined may have to be implemented if memory availability is an issue. We also note that the error analysis method and formulation can readily be applied to other particle methods, such as, e.g. PIC [78] where both convergence and numerical error are present.

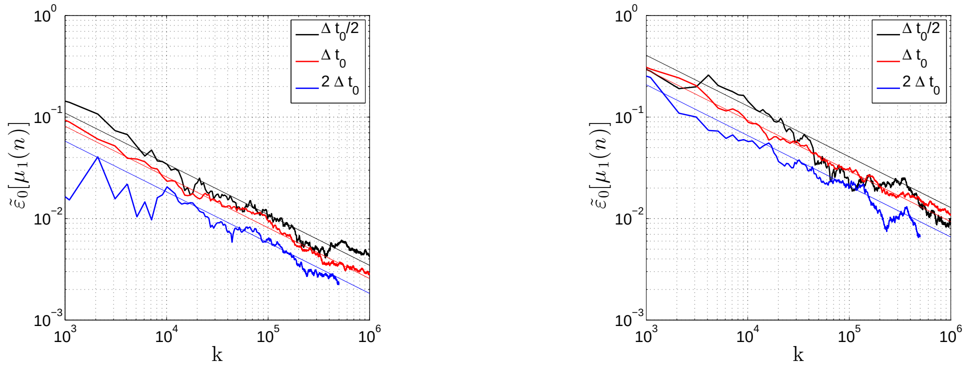


(a) Cell 1 (b) Cell 4



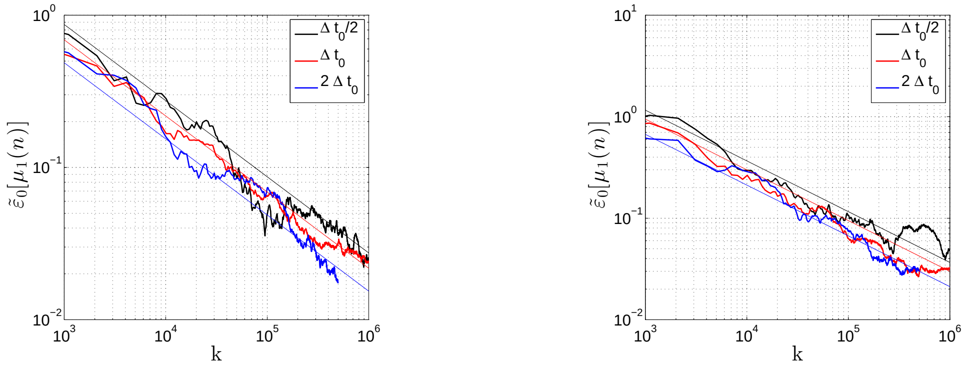
(c) Cell 7 (d) Cell 10

Figure 3.13: Influence of the time step on the autocorrelation function for N.



(a) Cell 1

(b) Cell 4



(c) Cell 7

(d) Cell 10

Figure 3.14: Convergence error for sampled n as a function of the time step

CHAPTER IV

On the accuracy of the direct simulation Monte Carlo method

4.1 Introduction

The direct simulation Monte Carlo method has been successfully applied to a wide variety of high speed rarefied flows [116] during the course of the past forty years. This is both the result of its superiority to most other numerical methods that presently exist for this flow regime and of its numerical efficacy. For most flows of practical interest, "satisfactory" simulation results are often obtained with as few as 20 particles per cell on average, a number which is often quoted in the literature as in e.g. [124], [63] or [90]. The accuracy of the DSMC method has thus been the subject of relatively few studies compared to other numerical techniques such as for instance finite volume schemes for the Euler equation [101]. This situation is also the consequence of both the relatively limited interest in and applicability of DSMC [116] compared to other numerical schemes (e.g. finite volume solution methods [101]) and of its formulation which is more physical in nature [15] than that of most other numerical methods. This situation makes traditional numerical error investigation tools commonly used for partial differential equations, such as Taylor series based analysises, difficult to apply. For a simple gas and for given boundary conditions, the numerical accuracy of the DSMC procedure, for the same pair selection scheme, collision procedure and boundary condition implementation is determined by the number of particles in the simulation N_p (or their weight $W_p \sim N_p^{-1}$), the time step Δt, and the spatial discretization of the domain into cells. For a fixed mesh, the error has been identified in previous studies as stemming from two sources. The first source of the error is the use of a finite number of samples S to estimate statistical properties of fluctuating quantities in the simulation. This error is termed the *convergence error* in the present work and was the subject of Chapter III. The second source of error, called *numerical error* is the subject of the present chapter. It is caused by the fact that the statistical properties of the stochastic process generated by the DSMC procedure, even when properly rescaled, depends on its numerical parameters. In previous work, this error is referred to by a variety of names. It is the *discretization error* of the thermal conductivity in [124], the *bias* in [41] or the *deterministic error* in [142]. It is often attributed, in cells with few particles, to the existence of repeated collisions between the same particles [134]. By analogy to other stochastic processes, it was argued in [41], that the numerical error was proportional to N^{-1} which was indeed observed to a certain extent in simulations. The same scaling was also observed in [124] for the numerical error of the heat flux for a Fourier flow. The influence of the mesh, which was for instance investigated in [139], on the solution will, however, not be studied in this work and all simulations will be conducted with the same mesh.

The aim of this chapter is to study the effect on the numerical error of the number of particles and time step in a DSMC simulation while keeping everything else fixed. The effect of the mesh size, collision schemes or the issue of non-convergence to the Boltzmann equation will not be investigated. The error in a DSMC simulation, as far as this work is concerned, is defined as the difference, in the limit of an infinite number of samples (so that convergence error cf. Chapter III is negligible), between simulation results for a given set of number of particles and time step and that obtained in the limit of both a much greater number of particles and much smaller time step. Both the number of particles and the time step of the reference solution are suitably chosen so that the value of the error for the given test case is reasonably insensitive to their values. The objective of this chapter is furthermore to present an investigation of the DSMC method within the framework of a test case that is sufficiently complex so as to highlight features or phenomena that users of the method are likely to encounter in practice. To accomplish this, instead of using a relatively simple canonical test case such as channel flow [77] [62], an axisymmetric jet flow is chosen. Axisymmetric jets are of great practical interest to the rarefied gas dynamics community [110] but are yet known to be difficult to simulate via DSMC which often produces sampled velocity or number density contours with spurious structures close to the centerline [61]. The source of these difficulties is the axisymmetric geometry whereby the volume of cells varies proportionally to the distance to the centerline far away from the axis (and quadratically close to it) that creates cells with a small volume close to the centerline which naturally tend to contain few particles. Problems in a jet flow geometry are further compounded by the decrease in number density in the streamwise direction and the propensity of particles to move away from the centerline.

One further aim of the current work is to investigate the spatial variation of the cellwise error (i.e. that of sampled quantities in each DSMC cell) throughout the domain instead of only focusing on a global measure of the error as was done in [55] or [41] although the latter will also be examined. The cellwise error, which, as will later be shown, exhibits great spatial variation throughout the flow, is that which is of most interest to practitioners. This is because most DSMC simulations are conducted so as to determine the value of moments of the velocity distribution function in a few cells of the simulation domain (such as the number density at a specific point) and not for the calculation of domain averaged quantities. A global error measure is more useful within the context of weighted residual methods [30] for partial differential equations for which an a priori expression for the error norm can actually be derived. The error framework previously introduced in Chapter III is first briefly presented. The effect of the number of particles on the numerical error is then assessed for a few cells of the domain. It is then examined for the entire domain with contour plots and the effect of the number of particles on global error norms is evaluated. The second part of the chapter focuses on the effect of the time step on the numerical error. Its influence on the cellwise error and global error norms is then studied. Finally, the findings of this study regarding the effect of the number of particles and time step on the error are summarized.

4.2 Analysis of the numerical error

4.2.1 Cell and particle based variables

Following the approach presented in Chapter III, the instantaneous number density of cell i can be written in terms of the indicator functions $\mathbf{1}_{\Omega_i}$ as follows:

$$n_i^k = \frac{W_p \sum_{j=1}^{N_p} \mathbf{1}_{\Omega_i}(\vec{\mathbf{x}}_j^k)}{\mathrm{V}_i}. \tag{4.1}$$

This clearly indicates that the number density in cell i only depends on the location $\{\vec{\mathbf{x}}\}_{j=1}^{N_p}$ of all the N_p particles in the simulation. This means that the numerical error for sampled number density is directly caused by inaccuracies for the location of the particles $\{\vec{\mathbf{x}}\}_{j=1}^{N_p}$. In case of variables that are carried by particles such as for instance the x velocity component V_x, the instantaneous velocity in the cell $V_{x,i}^k$ can be written as :

$$V_{x,i}^k = \frac{\sum_{j=1}^{N_p} \mathbf{V}_{x,j}^k \ \mathbf{1}_{\Omega_i}(\vec{\mathbf{x}}_j^k)}{\sum_{j=1}^{N_p} \mathbf{1}_{\Omega_i}(\vec{\mathbf{x}}_j^k)}. \tag{4.2}$$

which shows that the projection of a particle-based variable on the mesh depends on both the position of all particles $\{\vec{\mathbf{x}}\}_{j=1}^{N_p}$ and their velocities $\{\mathbf{V}_{x,j}\}_{j=1}^{N_p}$. This signifies that the numerical error of $\mu_i(V_x)$ will be affected by errors in the location and velocities of the particles.

4.2.2 Error types

Placing ourselves in the framework of Chapter III, the error in DSMC simulations is assumed to come from only two sources each corresponding to a specific error type. The first error, the *convergence error* is studied in Chapter III. It is caused by the use of a finite number of samples to estimate the statistical properties of computational DSMC particles. The second type of error, called *numerical error*, caused by the use of finite W_p and Δt values is the subject of the present part of this chapter.

4.2.3 Numerical Error

Let Y_i be any cell-based variable, e.g. N_i, and let y_i^k designate an observed output of that random variable at time step k. $\theta(Y_i)$ refers to any statistical property of random variable Y_i such as for example its first order moment μ. The estimator of $\theta(Y_i)$ at time step k is denoted by $\hat{\theta}^k(y_i)$. Because all the statistical estimators used are consistent i.e., $\lim_{k\to\infty}\hat{\theta}^k(y_i) = \theta(Y_i)$, in the following the limit of $\hat{\theta}^k(y_i)$ when $k\to\infty$ will be abbreviated as $\theta(Y_i)$. Numerical error $\bar{\varepsilon}_i$ for estimator θ_i is defined for a given simulation characterized by Δt and W_p as the difference between $\theta_i(W_p, \Delta t)$ and that obtained in the absence of any scaling:

$$\bar{\varepsilon}_i = \theta_i(W_p, \Delta t) - \lim_{W_p\to 1, \Delta t\to 0}\theta_i(W_p, \Delta t). \tag{4.3}$$

The value of the estimator in the limit is of course inaccessible in practice and must therefore be approximated by that obtained with a reference simulation with finite parameters $W_{p,0} \ll W_p$ and $\Delta t_0 \ll \Delta t$. The reference parameters have to be chosen so that the numerical error calculated for the test case at hand with the reference simulation is much larger than the true numerical error of the reference simulation, i.e.

$$|\theta_i\left(W_{p,0}, \Delta t_0\right) - \lim_{W_p \to 1, \Delta t \to 0} \theta_i\left(W_p, \Delta t\right)| \ll |\theta_i\left(W_p, \Delta t\right) - \theta_i\left(W_{p,0}, \Delta t_0\right)|. \tag{4.4}$$

This allows Eq. (4.3) to be approximated as:

$$\bar{\varepsilon}_i \approx \theta_i\left(W_p, \Delta t\right) - \theta_i\left(W_{p,0}, \Delta t_0\right). \tag{4.5}$$

4.2.4 Global error measure

The overall accuracy of the simulation can be assessed by considering $\bar{\varepsilon}_i$ for all N_c cells of the domain simultaneously by defining the following vector in $\mathbb{R}^{N_c}$:

$$\bar{\boldsymbol{\varepsilon}} \triangleq \left(\bar{\varepsilon}_1, \bar{\varepsilon}_2 \cdots, \bar{\varepsilon}_{N_c}\right). \tag{4.6}$$

The ℓ_∞, and normalized ℓ_1 or ℓ_2 norms of $\bar{\boldsymbol{\varepsilon}}$ are, respectively, defined as:

$$\|\bar{\boldsymbol{\varepsilon}}\|_\infty \triangleq \max_i |\bar{\varepsilon}_i|, \tag{4.7}$$

$$\|\bar{\boldsymbol{\varepsilon}}\|_1 \triangleq \frac{1}{N_c} \sum_i^{N_c} |\bar{\varepsilon}_i|, \tag{4.8}$$

and

$$\|\bar{\boldsymbol{\varepsilon}}\|_2 \triangleq \frac{1}{N_c} \left(\sum_i^{N_c} |\bar{\varepsilon}_i|^2 \right)^{1/2}. \tag{4.9}$$

The influence of numerical parameters W_p and Δt on the values of these norms will be addressed in sections 4.4.4 and 4.5.3, respectively.

4.3 Testcase

The exact same DSMC procedure implementation is used in this chapter as is used in Chapter III where further details are given. Collisions are modeled via the NTC (No Time Counter) collision scheme [15]:

$$N_{\text{coll}} = \frac{1}{2\text{V}} W_p \Delta t N \overline{N} \left(\sigma g\right)_{\max}, \tag{4.10}$$

where $\overline{N}$, the average number of particles in the cell, is obtained by an exponential moving average.

4.3.1 Geometry and flow conditions

Identical flow conditions as those employed in Chapter III are used to study the numerical error. They are, however, briefly summarized here for convenience. The test case consists of an axisymmetric jet with initial radius $R = 0.001$ m of simple argon gas at $\text{Kn} = 0.01$ and $\text{M} = 1$. The computational mesh and domain are shown in Fig. 4.1. Supersonic outflow conditions are used for the top and downstream parts of the domain. The upper portion of the domain is truncated to minimize undue computational expense as the region of interest is located close to the centerline. The effect of the domain size or shape on the solution is not investigated. Any effect of the domain size or shape on the solution is expected to have impacted equally as much all test cases of the study.

4.3.2 Computational Parameters

The different test cases that are run during the course of this study are summarized in Tables 4.1 and 4.2 which indicate the corresponding Δt and W_p values that are used for each test case. The time step does satisfy the DSMC requirement that it be less than the mean collision time for all test cases as discussed in section 4.5. The

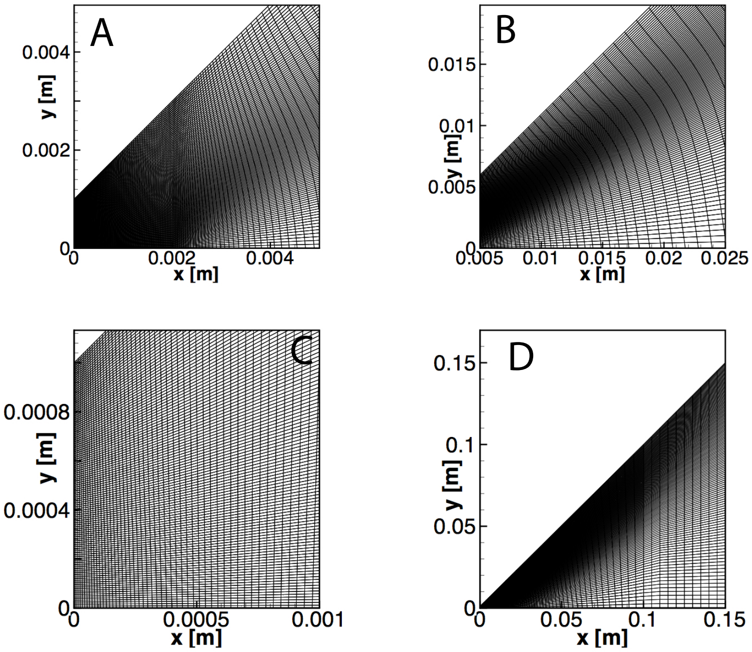


Figure 4.1: View of the computational domain and mesh. A, B: domain views, C: closeup view of the inlet and D: overall domain view

average number of particles obtained for test case C1024 is shown in Fig. 4.3 which is identical up to a multiplicative constant to that obtained for all other test cases of Table 4.1. As a constant cell weight and time step are used throughout the domain, its spatial variation is dictated both by the number density of the solution and the volume of the cells of the domain (which scales as $\sim y$ near the axis and $\sim y^2$ far away from the axis). The various test cases will be designated by the name indicated in these tables. Following the approach of Chapter III, the value of the numerical error is examined in greater detail in the following for the 12 cells shown in Fig. 4.2.

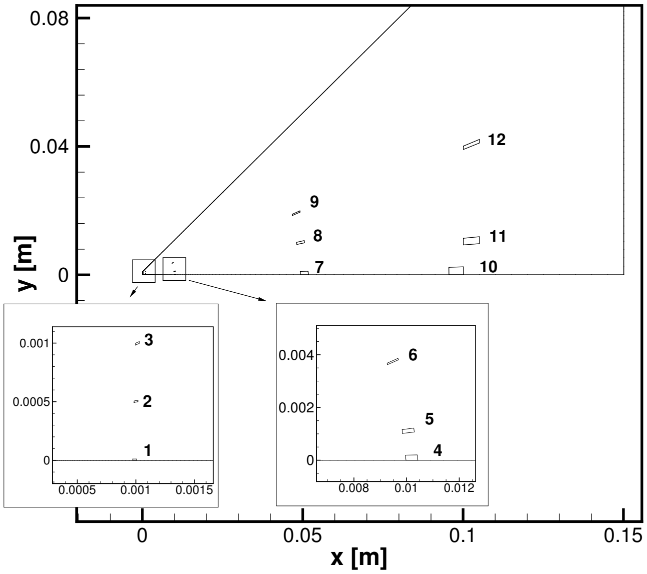


Figure 4.2: Location and numbering of cells studied in detail.

Test Case	**Number of particles (M)**	$\mathbf{W_p}$	Δt [s]
C1024	536.0	0.35×10^8	5.0×10^{-9}
C512	268.0	0.7×10^8	5.0×10^{-9}
C256	134.0	1.4×10^8	5.0×10^{-9}
C128	67.0	2.8×10^8	5.0×10^{-9}
C64	33.5	5.6×10^8	5.0×10^{-9}
C32	16.7	11.2×10^8	5.0×10^{-9}
C16	8.4	22.4×10^8	5.0×10^{-9}
C8	4.2	44.8×10^8	5.0×10^{-9}
C4	2.1	89.6×10^8	5.0×10^{-9}
C2	1.0	179.2×10^8	5.0×10^{-9}

Table 4.1: Test cases to investigate the effect of the particle count on the numerical error.

Test Case	**Number of particles (M)**	$\mathbf{W_p}$	Δt [s]
C16	8.4	22.4×10^8	5.0×10^{-9}
C16200	33.5	22.4×10^8	1.0×10^{-8}
C16400	33.5	22.4×10^8	2.0×10^{-8}
C64	33.5	5.6×10^8	5.0×10^{-9}
C64125	33.5	5.6×10^8	6.25×10^{-9}
C64150	33.5	5.6×10^8	7.5×10^{-9}
C64200	33.5	5.6×10^8	1.0×10^{-8}
C64400	33.5	5.6×10^8	2.0×10^{-8}
C256	134.0	1.4×10^8	5.0×10^{-9}
C256200	134.0	1.4×10^8	1.0×10^{-8}
C256400	134.0	1.4×10^8	2.0×10^{-8}
C512	268.0	0.7×10^8	5.0×10^{-9}
C512200	268.0	0.7×10^8	1.0×10^{-8}
C512300	268.0	0.7×10^8	1.5×10^{-8}
C512400	268.0	0.7×10^8	2.0×10^{-8}

Table 4.2: Test cases for the study of the influence of the time step on the numerical error.

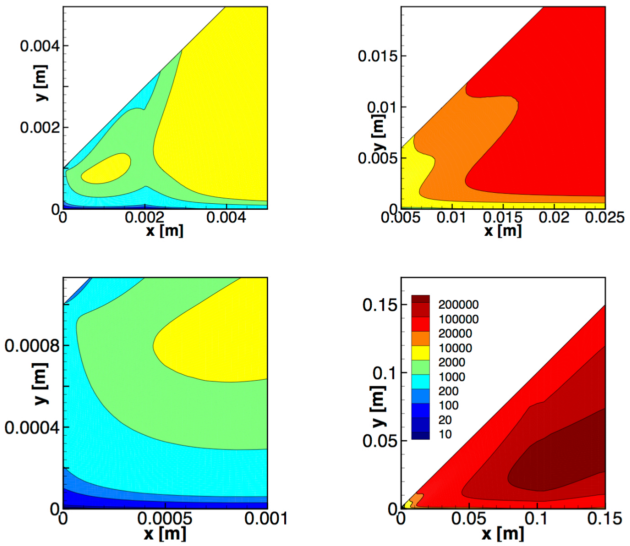


Figure 4.3: Average number of particles in each cell for test case C1024. The cell with the smallest average number of particles is located at the corner of the centerline and the inlet and contains 4.19.

4.4 Effect of the number of particles on the numerical error

The goal of this section is to try to establish a link between the value of the numerical error for a given cell and the total number of particles in the simulation or particular cell. The numerical error for $\mu_1(n)$ and $\mu_1(V_x)$ are examined both globally and in detail for a few selected cells. The numerical error $\bar{\varepsilon}_i$ is calculated following Eq. (4.5) using case C1024 (outlined in Table 4.1) as the reference solution by hence using $W_{p,0} = 0.35 \times 10^8$ and $\Delta t_0 = 5.0 \times 10^{-9}$s. To aid in the analysis, a normalized value of the error is defined as:

$$\bar{\varepsilon}_{i,0}[\theta] = \frac{|\bar{\varepsilon}_i[\theta]|}{|\theta_i(W_{p,0}, \Delta t_0)|}. \tag{4.11}$$

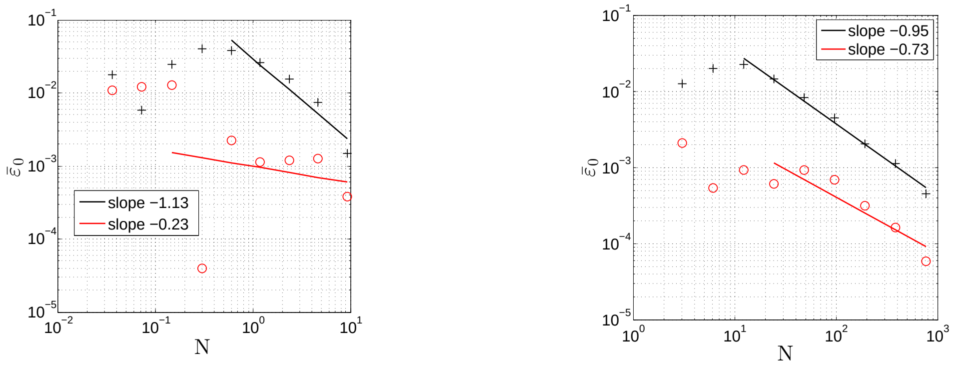
4.4.1 Numerical error for mean of number density $\bar{\varepsilon}[\mu_1(n)]$

4.4.1.1 Cellwise Analysis

For 8 of the cells displayed in Fig. 4.2, a plot of the error as a function of the average number of particles in that cell is plotted in Figs. 4.4 and 4.5. The 9 data points shown for each cell correspond to the 9 test cases outlined in Table 4.1. For each cell, the point with the lowest particle count corresponds to C1 while the largest is obtained with C512 (C1024 being the reference solution). There is not a direct relation between the average number of particles in the cell and the value of the numerical error. The value of the numerical error does furthermore not seem to correlate with any local cell variable such as, e.g., the local collision frequency. Identical average numbers of particles lead to different values of the numerical error for distinct cells. General conclusions are furthermore difficult to draw regarding the asymptotic behavior of the error with N. For most cells, the error is observed to vary as $N^{-\alpha}$ with $\alpha \approx 1$ although clear distinct behavior is observed for cells 2, 9 and 12 for which $\alpha \approx 0.5$. The N-dependency of the numerical error is thus observed to vary spatially through the domain. The spatial variation of the error is discussed in the next section.

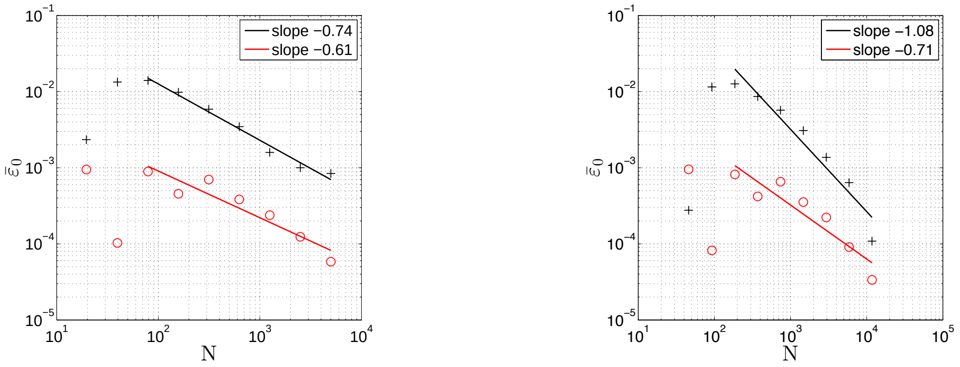
4.4.1.2 Global Analysis

We now investigate the spatial distribution of the error for multiple particle counts. The goal is to determine for the present test case the zones that are most prone to numerical error and, if possible, to determine if they share any characteristics that could potentially be used to identify them a priori. The absolute value of the normalized numerical error for $\mu_1(n)$ is plotted for three representative particle counts of Table 4.1 in Figs. 4.6, 4.8 and 4.10. What is apparent in these plots is that, as previously noted in the one-dimensional plots, the error magnitude varies widely throughout the domain for a given test case. For a given particle count, low numerical error zones are not characterized by larger numbers of particles per cell. This is apparent by comparing Fig. 4.3 which displays the average number of particles per cell ($\overline{N}$) for test case C1024 and Fig. 4.6 which shows error contour lines. When W_p changes for the entire domain, the average number of particles in each cell of the domain will only see its value change by a factor of $1/W_p$. Thus the general shape of $\overline{N}$ will be preserved across all particle counts and never correlates with the spatial variations of $\bar{\varepsilon}_0(\mu_1(n))$. Greater insight into the spatial behavior of the numerical error can be gained by plotting its signed value as opposed to its absolute value as done in Figs. 4.7, 4.9 and 4.11. In these figures, it can be seen that the number density is either underestimated ($\bar{\varepsilon}_i > 0$) or overestimated ($\bar{\varepsilon}_i < 0$) in distinct spatial regions separated by the border where $\bar{\varepsilon}_i \approx 0$. The magnitude of the overestimation (or underestimation) decreases within each zone as a border is approached leading to a smooth spatial variation of the error. This particular spatial behavior of the error explains the low error regions separated by high error regions (similar to valleys surrounded by peaks) that are observed when its absolute value is plotted in Figs.



(a) Cell 1

(b) Cell 4



(c) Cell 7

(d) Cell 10

Figure 4.4: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ $(+)$ and $\mu_1(V_x)$ $(\circ)$.

4.6, 4.8 and 4.10.

The effect of the particle count on the observed spatial patterns can be seen by comparing the numerical error along some specified lines of the domain. This allows a more quantitative comparison between particle counts of the results presented in Figs. 4.6, 4.8 and 4.10. The numerical error for $y = 0.0005$ m is shown in Figs. 4.13 and 4.15 and for $y = 0.001$ m in Figs. 4.14 and 4.16. Figures 4.13 and 4.14 clearly indicate the presence in the simulation domain of distinct zones where the error has a constant sign for all particle counts. For instance at $y = 0.001$ m, the error is consistently positive for $x < 0.00035$ m and negative otherwise. This corresponds to the same regions that are visible in Figs. 4.6 to 4.11. Although only two one-dimensional traces are shown in the present work for the sake of conciseness, identical observations and trends are obtained in 10 additional traces taken throughout the domain. The error is also seen to exhibit increasingly greater noise (i.e greater spatial oscillations) for smaller values, as for instance demonstrated by the behavior of the black compared to that of the blue line in Fig. 4.16. This is caused by the much greater convergence (i.e. small convergence error) that is required to calculate lower error values. We now seek to relate some of the results observed in the one dimensional traces to those previously obtained in this section. Point 2 was seen to have a slope of -0.32 for $\log(\bar{\varepsilon}_0)$ vs N in Fig. 4.5(a) which notably differs from the approximate -1 value observed for most points. It is located at $x = 0.001$ m in Figs. 4.13 where a sharp drop for all particle counts in the value of the error is observed. That same drop is visible in field plots such as e.g., in Fig. 4.6. In contrast to Point 2, Point 3 is located at $x = 0.001$ m in Fig. 4.14 where no precipitous drop in the error is observed. It is furthermore found in Fig. 4.5(b) to have a slope of $-0.92 \sim -1$, similar to that obtained for most other points in the domain. The difference between Points 2 and 3 thus suggests the occurrence of two types of error dependencies with N. In the first, illustrated by Point 2, the absolute value of the numerical error has a low value compared to the rest of the domain and convergences as $N^{-\alpha}$ with $\alpha \ll 1$. In the second, exemplified by Point 3, which is followed by most points of the domain, the absolute value of the numerical error is close to that of most cells in the domain and converges as $N^{-\alpha}$ with $\alpha \geq 1$. It thus seems like the convergence behavior of the error can be tied to its value relative to other cells. To further study the spatial variation of the convergence rate of the numerical error with N, a curve fit identical to that made to produce Figs. 4.4(a) to 4.5(d) is performed for each individual cell of the simulation domain by only using results from simulations C512, C256, C128 and C64, C1024 being the reference simulation. The value of the slope obtained for each individual cell is plotted in Fig. 4.12. Although the field plot is relatively non-smooth, zones in red with a low value of the numerical error slope with N clearly correspond to zones with a low absolute value of the numerical error that are shown in Figs. 4.6, 4.8 and 4.10. This can be further seen in Fig. 4.13 where both the value of the slope and the numerical error are plotted and a clear correlation is apparent between low numerical error and a less than 1 exponent value.

4.4.1.3 Error propagation

Information in a DSMC simulation consists in the location $\{\vec{\mathbf{x}}_j\}_{j=1}^{N_p}$ and velocity $\{\vec{\mathbf{v}}_j\}_{j=1}^{N_p}$ of all particles of the computational domain. Correspondingly, each particle of the simulation can be considered as subject to errors $\vec{\varepsilon}_{\vec{\mathbf{x}}}$ and $\vec{\varepsilon}_{\vec{\mathbf{v}}}$. As number density is defined by Eq. (4.1), the numerical error for the statistical properties of number density throughout the simulation domain only depends on $\vec{\varepsilon}_{\vec{\mathbf{x}}}$ whereas the numerical error for the statistical properties of cell-averaged velocities, because of their definition via Eq. (4.2) , depend on both $\vec{\varepsilon}_{\vec{\mathbf{x}}}$ and $\vec{\varepsilon}_{\vec{\mathbf{v}}}$. Given all the results of the current study, the generation and propagation of numerical error inside the domain can be thought of as occurring according to the somewhat simplified series of steps outlined below. Particles are injected at the inlet and then enter the first cell of the domain, designated by subscript i, that has its left edge at $x = 0$ with $\vec{\varepsilon}_{\vec{\mathbf{x}}} = \vec{0}$ and $\vec{\varepsilon}_{\vec{\mathbf{v}}} = \vec{0}$. In the first cell that the particles enter, the collision rate given by Eq. (4.10) is inaccurately calculated due to the finite number of particles in the cell and subject to error $\varepsilon_{\text{coll}}$. The first cell is furthermore subject to a back flow of particles from inside the computational domain which are each subject to numerical error. The end result of the inaccurate modeling of the collision rate inside cell i and the inflow of error from other cells into cell i is that particles inside the cell i end up being affected by errors $\vec{\varepsilon}_{\vec{\mathbf{x}},i}$ and $\vec{\varepsilon}_{\vec{\mathbf{v}},i}$. Futhermore, particles that move out of cell i into neighboring j cells are subject to errors $\vec{\varepsilon}_{\vec{\mathbf{x}},i\to j}$ and $\vec{\varepsilon}_{\vec{\mathbf{v}},i\to j}$ which are themselves different from $\vec{\varepsilon}_{\vec{\mathbf{x}},i}$ and $\vec{\varepsilon}_{\vec{\mathbf{v}},i}$ as they are biased by the velocity of particles and depend on the geometry of the interface between cells i and j. The two error types furthermore compound each other as the particles are advected inside the domain. That is because, $\frac{d\vec{x}}{dt} = \vec{v}$ so that an error introduced for the velocity of a particle is translated into an error for its position which magnitude increases with time. Conversely, systematic error for the location of particles in the domain will lead to systematic error in the calculation of collisions that will in turn lead to numerical error for the velocity of particles. The only way to fully explain the propagation of the error is to obtain the velocity distribution functions of the error for all cells. This is, however, exceedingly difficult to do because of the amount of data generated and the resolution required to compare the distribution function between test cases. We thus limit ourselves to a rather qualitative discussion of the propagation of the numerical error in the domain is necessarily limited by the data at our disposal.

i) The improper modeling of collisions introduces error in the simulation which manifests itself here by certain cells containing on average too many or too few particles. The number density error propagates to adjacent cells due to the movement of particles whilst simultaneously increased or decreased by the velocity error. Cells, however, receive particles from multiple cells each of which contain on average too many or too few particles so the error may end up increasing, decreasing or being cancelled out. It is the relative smoothness of particle trajectories, when averaged over many time steps, that causes the smooth patterns that are observed for the number density error. ii) When collisions cease being important, a negligible amount of error is generated due to an inaccurate modeling of collisions and the spatial distribution of the error is determined by the mixing of particles that occurs in the flow. This phenomenon is particularly visible for all test cases past $x \approx 0.005$ m where the flow is almost free molecular. In that zone, numerical error contours are observed to diverge becoming nearly parallel to the streamlines of the flow which indicates that no new error is generated in that zone and that its patterns in that region are only dictated by its diffusion via the movement of particles. The increasing width of the zone of $\overline{\varepsilon_i} \approx 0$ with x is a manifestation of the cancellation of the error by the mixing of particles. iii) An unfortunate consequence of the geometry adopted for this study is that the cell with the smallest average number of particles is located right next to the inlet at $x = 0$, as shown in Fig. 4.3. Because it contains only 4.19 particles on average in the reference simulation, it is likely subject to substantial numerical error. This error in turn propagates to the rest of the domain. This should not however overly impact the trends such as the power laws that were found in this study, as the proportion of particles that have traversed this zone is small in most cells downstream. Very few collisions also take place in the cells next to the centerline (less than 1 per time step in the C1024 case) because of their small volume, which further limit the impact of an insufficient average number of particles.

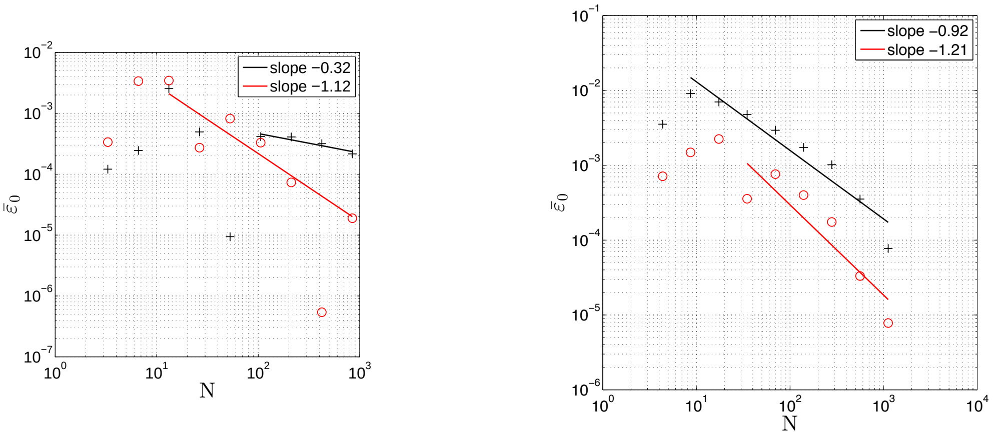
4.4.2 Numerical error for mean of velocity $\bar{\varepsilon}\left[\mu_1\left(V_x\right)\right]$

The convergence of the numerical error of $\mu_1\left(V_x\right)$ with N is studied in Figs. 4.4 and 4.5 for 8 distinct cells of the domain. A convergence rate rate of $N^{-\alpha}$ with $\alpha \sim 1$ is found for most cells albeit with some variations between cells. The normalized numerical error for $\mu_1(V_x)$ can be seen to be systematically an order of magnitude lower than that for $\mu_1(n)$. The spatial variation of the convergence rate is shown in Fig. 4.19. Part of the pattern appears similar to that observed for the number density in Fig. 4.12 in particular far from the inlet in the middle of the domain. That similarity can also be seen in the peak observed in both Figs 4.14 and 4.20 at $x = 0.00035$ m. The two rates, however, differ closer to the inlet. The same phenomena that were previously discussed for $\bar{\varepsilon}[\mu_1(n)]$ are also visible for $\bar{\varepsilon}[\mu_1(V_x)]$ such as the diffusion of error by particle mixing far from the inlet where few collisions are present. Another manifestation of this mixing phenomenon is visible close to the inlet in Fig. 4.17 where a narrow low error zone extends from $y = 0.0001$ m at the inlet exit to the centerline. In that zone, $V_y \sim 0^+$, so that it is traversed roughly equally as much by particles coming from below as by particles coming from above characterized by $\boldsymbol{\varepsilon}_{v_x} > 0$ and below characterized by $\boldsymbol{\varepsilon}_{v_x} < 0$ as shown in Fig. 4.18. This results in a sustained zone with a low error value in the x direction due to the two error fluxes counterbalancing each other.

The spatial distribution of the error, only shown for one test case in Figs. 4.17 and 4.18, also differs from that observed for velocity although some common features are present (most notably further removed from the inlet). Zones with a low error value are also characterized by a low convergence of the error with N similarly to what is previously observed for the number density. This is apparent in Fig. 4.20 which similarly to Fig. 4.14 possesses a peak for the slope that occurs at the minimum error location. This phenomenon is also observed in other one-dimensional traces that are not shown here.

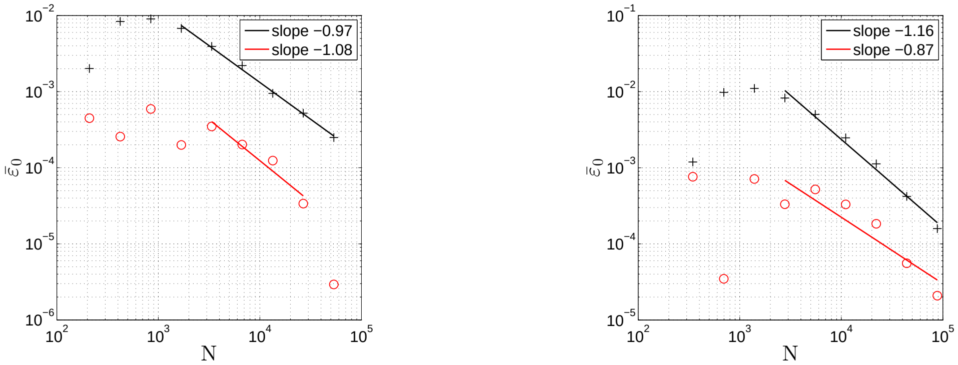
4.4.3 Discussion of results

The preceeding two sections have thus highlighted the fact that no universal scaling exists for the dependency of cellwise numerical error on particle count (save for the existence of power-law behavior). These results are not unexpected given the strong influence of the flow geometry on the numerical error as demonstrated by the existence and shape of the error patterns discussed in sections 4.4.1.2 and 4.4.2. It seems very likely that a universal scaling of the cellwise value of the numerical error with N can only be exhibited in cases with relatively mild spatial inhomogeneities. This would explain the N^{-1} scaling found in [124] for one dimensional channel and Fourier flow. A full exposition of an a priori determination procedure for the numerical error due to the time step is outside the scope of the present chapter. Taking advantage of the near-universal power law behavior that was highlighted, (i.e. $\overline{\varepsilon}_i = C_i N^{-\alpha_i}$), one might imagine the following procedure to a priori determine the value of the numerical error due to N. As such, three (or more) different simulations are conducted for three distinct N values so as to determine the value of the exponent of the power law α_i and constant C_i. These can in turn be used to estimate the value of the error obtained for any N value.



(a) Cell 2

(b) Cell 3



(c) Cell 8

(d) Cell 11

Figure 4.5: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ $(+)$ and $\mu_1(V_x)$ $(\circ)$.

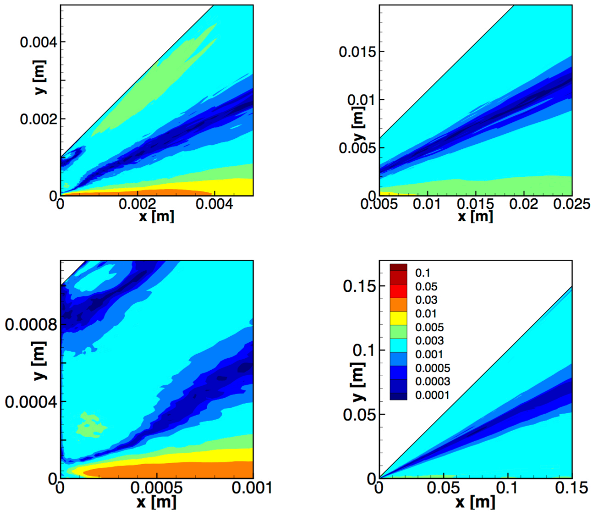


Figure 4.6: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ for C64.

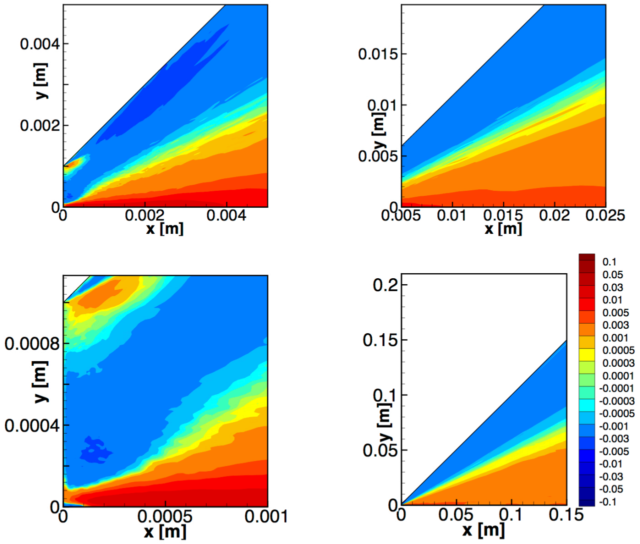


Figure 4.7: Signed normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ for C64.

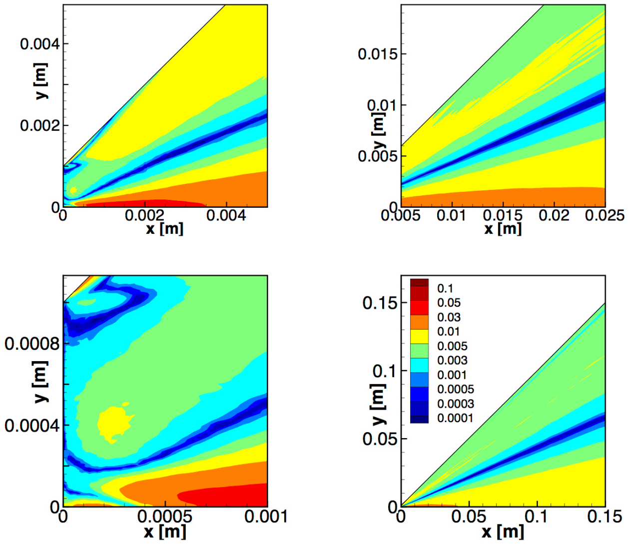


Figure 4.8: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ for C16.

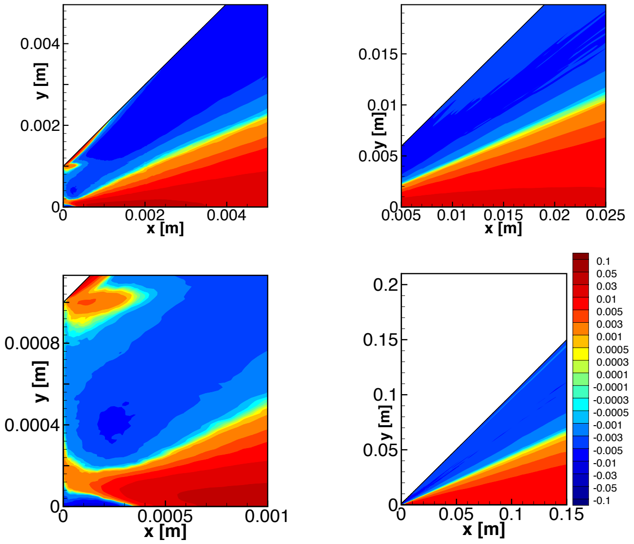


Figure 4.9: Signed normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ for C16.

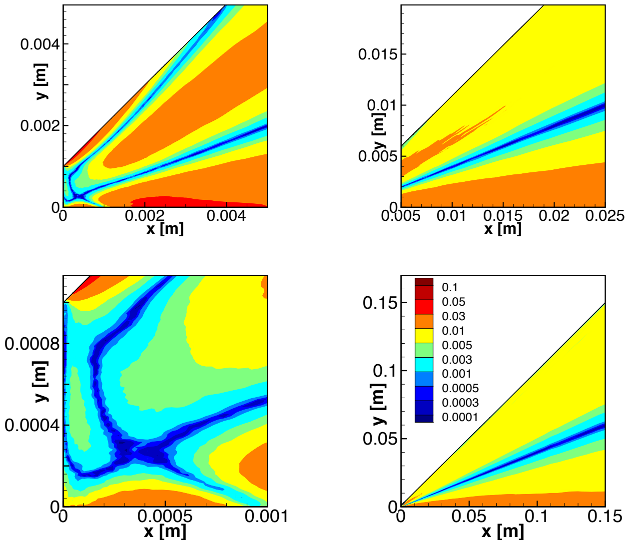


Figure 4.10: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1\,(n)$ for C4.

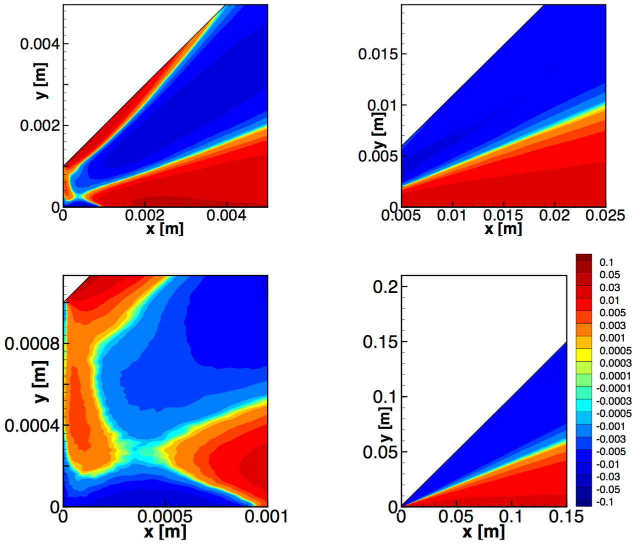


Figure 4.11: Signed normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1\,(n)$ for C4.

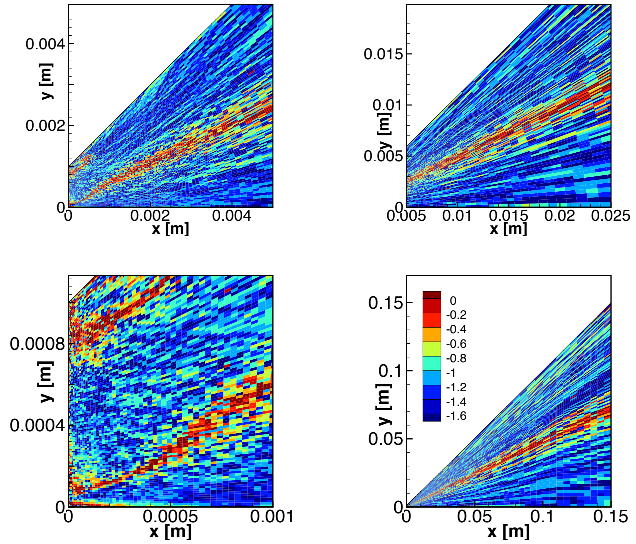


Figure 4.12: Value of the power of N for the slope of numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$.

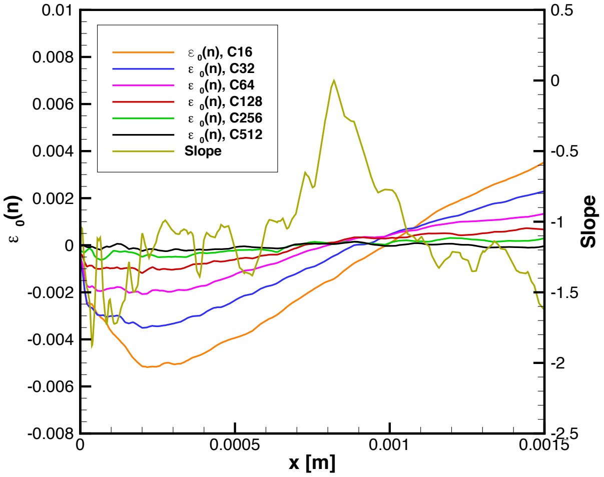


Figure 4.13: Normalized value of the numerical error of $\mu_1(n)$ at $y = 0.0005$ m.

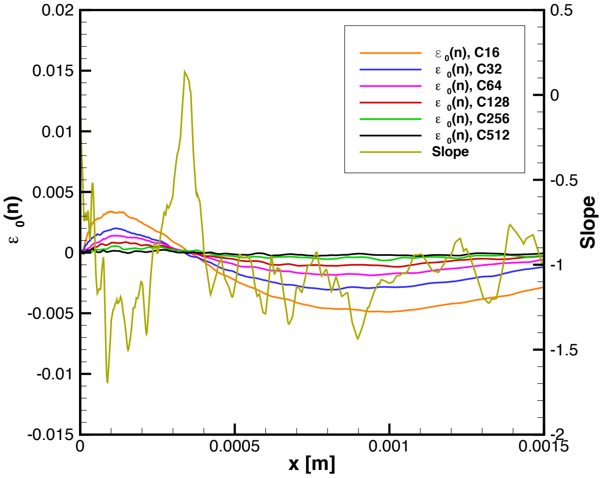


Figure 4.14: Normalized value of the numerical error of $\mu_1(n)$ at $y = 0.001$ m.

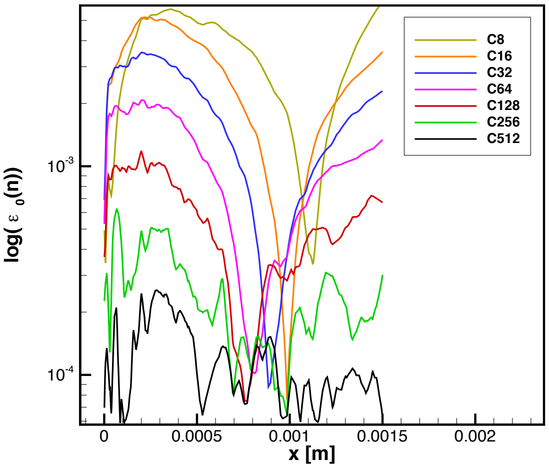


Figure 4.15: Normalized absolute value of the numerical error of $\mu_1(n)$ at $y = 0.0005$ m.

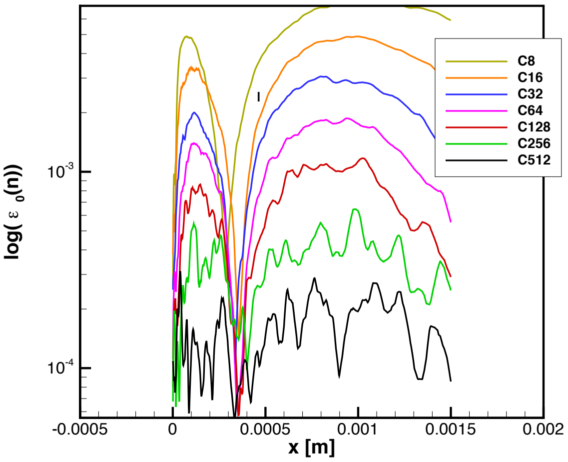


Figure 4.16: Normalized absolute value of the numerical error of $\mu_1(n)$ at $y = 0.001$ m.

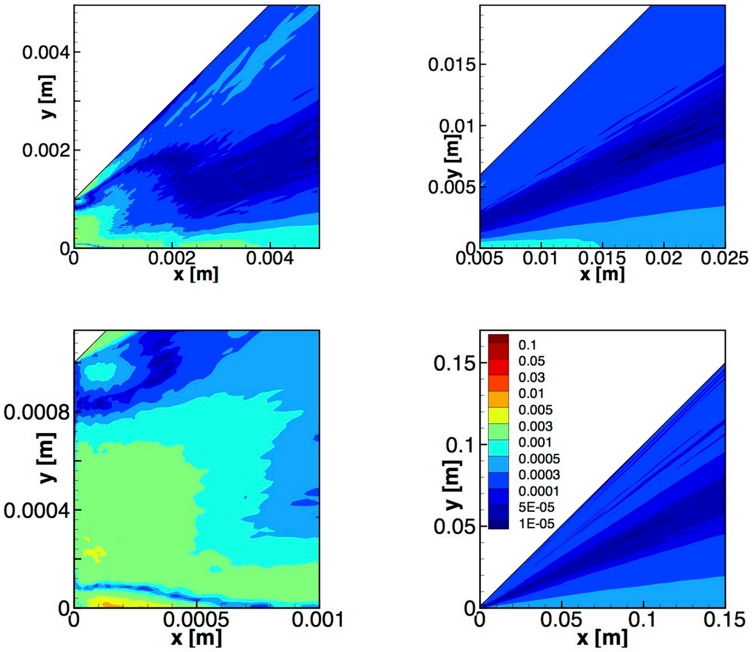


Figure 4.17: Normalized absolute value of the numerical error $\bar{\varepsilon}_0$ for $\mu_1(V_x)$ for C64.

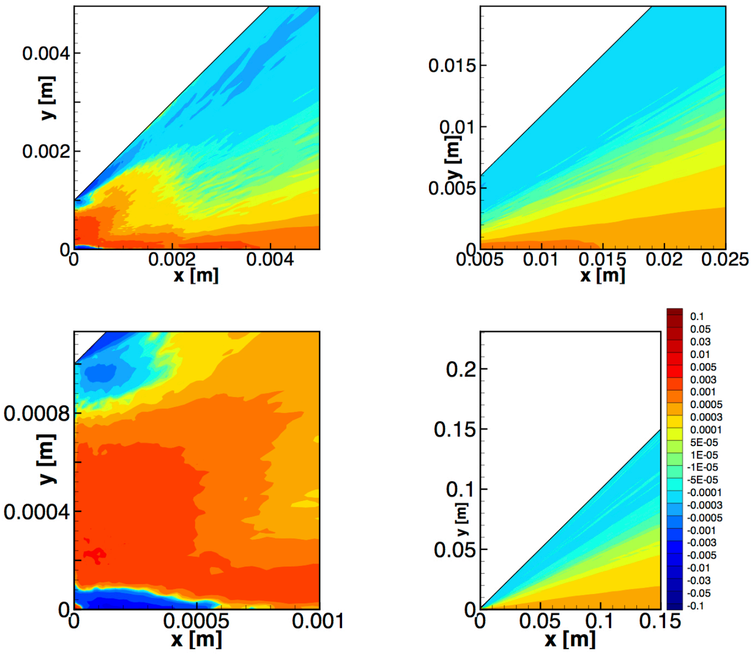


Figure 4.18: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1\left(V_x\right)$ for C64.

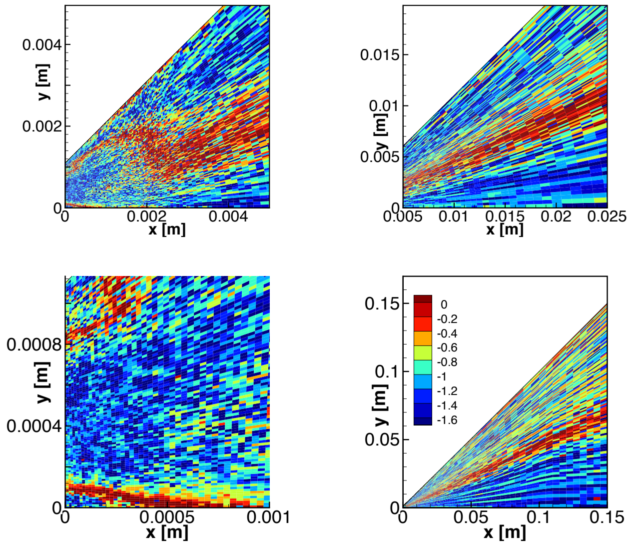


Figure 4.19: Value of the power of N for the slope of numerical error $\bar{\varepsilon}_0$ for $\mu_1\left(V_x\right)$

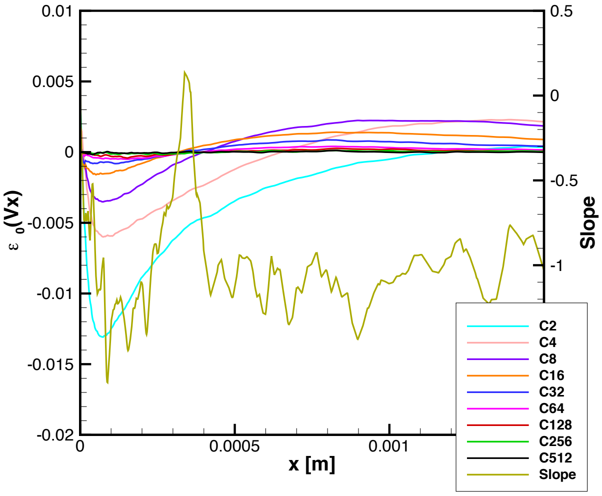


Figure 4.20: Numerical error for $\mu_1\,(V_x)$ at $y = 0.001$ m

4.4.4 Global error norms of the numerical error

The impact of spatial inhomogeneities or test case peculiarities on the numerical error can be lessened by considering its average value over all cells of the domain as was for instance done in [41]. This furthermore makes a universal scaling of the error with N more likely to emerge. To this effect, the values of $\|\bar{\boldsymbol{\varepsilon}}_0\|_\infty$, $\|\bar{\boldsymbol{\varepsilon}}_0\|_1$ and $\|\bar{\boldsymbol{\varepsilon}}_0\|_2$, respectively, defined by Eqs. (4.7) , (4.8) and (4.9) are plotted as a function of particle count in Fig. 4.21. The symbols correspond to error values that are obtained from actual simulations while the lines are linear curve fits to those results. The value of the slopes of the different lines are indicated in the same figure. The ℓ_1 and ℓ_2 norms are observed to behave fairly similarly for both n and V_x approximately scaling as N^{-1}. The infinity norm is observed to follow a power law $N^{-\alpha}$ with $\alpha \sim 0.5$. From a simulation accuracy standpoint, the infinity norm of the error is more useful as it provides an upper bound for the value of the error. The ℓ_1 and ℓ_2 norms are, however, more representative of the overall level of accuracy as they involve an averaging of the value of the error over multiple cells. The ℓ_1 norm of the numerical error has been shown to be of order N^{-1} in [35] under a series of assumptions for a stationary regularized form of the Boltzmann equation. This theoretical result has, however, still not been proven to be valid in the presence of inflow and outflow boundary conditions [126] as are used for the present simulation where the total number of particles fluctuates. There is, however, no reason why this scaling would not hold with such boundary conditions. The fact that this scaling was indeed observed in [124] in turn suggests that the N^{-1} observed in Fig. 4.21 for ℓ_1 and ℓ_2 might not be coincidental.

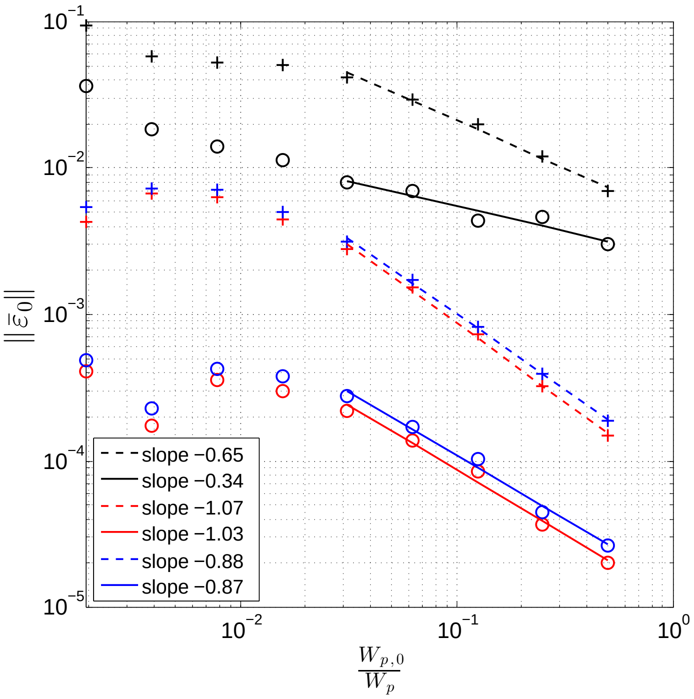


Figure 4.21: ℓ_∞ (black), ℓ_1 (red) and ℓ_2 (blue) norms of the normalized numerical error for $\mu_1(n)$ (dashed lines and + symbols) and $\mu_1(V_x)$ (solid lines and ∘ symbols) as a function of particle count.

4.5 Effect of the time step on the numerical error

4.5.1 Results for $\bar{\varepsilon}[\mu_1(n)]$ and $\bar{\varepsilon}[\mu_1(V_x)]$

The effect of the time step on numerical error is investigated by running the simulations outlined in Table 4.2, where, for four distinct particle counts, the time step is varied. The numerical error $\bar{\varepsilon}_i$ is calculated following Eq. (4.5) using case C1024 (outlined in Table 4.1) as the reference solution by hence using $W_{p,0} = 0.35 \times 10^8$ and $\Delta t_0 = 5.0 \times 10^{-9}$ s. The normalized value of the numerical error is defined by Eq. (4.11). Due to the formulation of the DSMC technique [15], Δt has to be less than τ_{mct}, the mean collision time, which is the average time between collisions for a particle in a cell. This requirement is well satisfied for the baseline time step of 5.0×10^{-9} s as shown in Fig. 4.22. This requirement is furthermore satisfied for all the cases in Table 4.2.

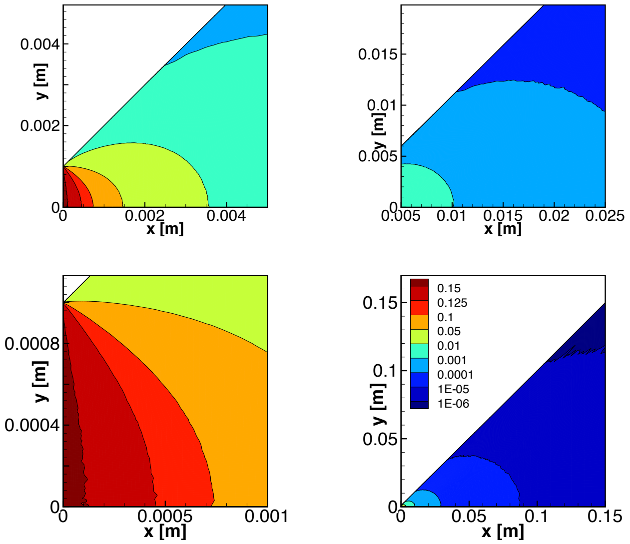


Figure 4.22: $\frac{\Delta t}{\tau_{\text{mct}}}$ for $\Delta t = 5.0 \times 10^{-9}$ [s], $\left(\frac{\Delta t}{\tau_{\text{mct}}}\right)_{\text{max}} = 0.159$ in front of the inlet.

The numerical error for both $\mu_1(n)$ and $\mu_1(V_x)$ is plotted as a function of Δt for four particle counts C16, C64, C256 and C512 in Fig. 4.23 for 4 different cells of the domain. A linear regression is fit to the numerical error obtained for $\mu_1(V_x)$ in the C512 case and the value of the slope is indicated on the plots. No clear mathematical relation for the value of the error as a function of $\left(\frac{\Delta t}{\Delta t_0}\right)$ is apparent in these plots for all particle counts for these cells or any others (not shown here). For the lower particle counts C16 and C64, the error actually drops for $\mu_1(n)$ with increasing Δt but rises for $\mu_1(Vx)$ or vice versa. As the particle count is increased, the rise of the numerical error with Δt becomes more systematic and monotonic for both $\mu_1(n)$ and $\mu_1(V_x)$. In particular, a power law seems to adequately describe the Δt dependency of the error of $\mu_1(V_x)$ for the C512 case but not that of $\mu_1(n)$. In order to gain further insight into the spatial variation of the error when Δt is increased, the ratio of the numerical error for $\mu_1(V_x)$ at $2 \times \Delta t_0$ over that obtained at Δt_0 is plotted in Figs. 4.24 and 4.25 for C16 and C256. In the near inlet region, $x \leq 0.0001$ m, increasing the particle count seems to produce a more uniform increase of the error. That is to say that the error is less likely to decrease with increasing Δt and that the ratio of the increase is more uniform. This can be seen in Fig. 4.26 where the error for C256 almost monotonically increases with Δt for $x < 0.002$ m as opposed to for the C64 cases where the error decreases for $\Delta t \leq 2 \times \Delta t_0$ before increasing when $\Delta t = 4 \times \Delta t_0$.

4.5.2 Discussion of results

Within the framework of this test case, no clear scaling of the cellwise error introduced by the time step in terms of Δt is found. Previous studies looking into the *truncation error* conducted for the homogeneous Boltzmann equation [114] and within the context of a one-dimensional Couette flow for wall quantities [65], however, found that the error scaled as $\mathcal{O}(\Delta t)^2$ in the limit of $\Delta t \to 0$. This scaling is likely not observed in the current study because of the relatively large value used for Δt and the spatial non homogeneities. For test cases of practical interest, however, a relatively large time step is used to maximize computational efficiency and spatial homogeneities are almost always present. Given these two conditions, the present results are much more representative of the types of conditions that the DSMC user is likely to encounter in practice. No clear a priori determination of the cellwise numerical error scaling with respect to Δt seems to be possible for reasonably complex DSMC flows such as this jet although a power law type dependency, i.e. $\overline{\varepsilon} \sim N^{-\alpha}$ with $\alpha > 0$, seems to be almost universally observed. The same procedure as that outlined in section 4.4.3 for N could therefore be used to determine error a priori for an arbitrary Δt value.

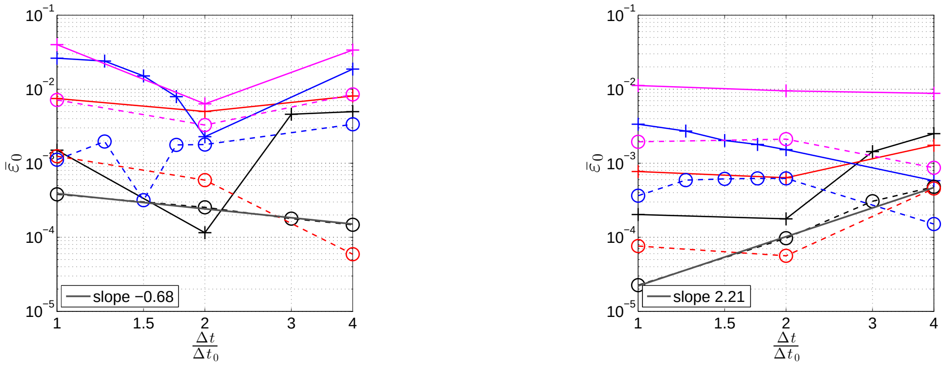
4.5.3 Global error norms of the numerical error

Having focused our attention on the cellwise error in the preceeding section, we now investigate the effect of the time step on the global norms of the numerical error, much like is done in section 4.4.4 for particle count. To this end, the ℓ_2 norms of the numerical error are plotted in Fig. 4.27 as a function of the time step for three different particles counts. Only the ℓ_2 norm is shown as the ℓ_1 demonstrates almost identical behavior. For the lowest particle count C64, the global error actually decreases as the time step is increased as opposed to the two larger particle counts where the global error monotonically increases with Δt. Such behavior was also previously observed for the cellwise value of the error and highlights the need for the simulation to contain enough particles if the expected increase in numerical error with Δt is to be observed. The numerical error for $\mu_1 (V_x)$ in particular seems to exhibit a quadratic scaling with respect to Δt. This $\mathcal{O} (\Delta t)^2$ scaling for the global error norm is in line with that obtained in [114] for the homogeneous Boltzmann equation or [65]. This scaling or the relative monotonicity of the global error norms with Δt is in marked contrast to what is observed for cellwise error in section 4.5.1. It thus appears that the behavior of the numerical error in a fairly homogeneously complex DSMC simulation, such as the present, can only be related to analytical scaling results with respect to Δt through its global norm.

4.6 Conclusions

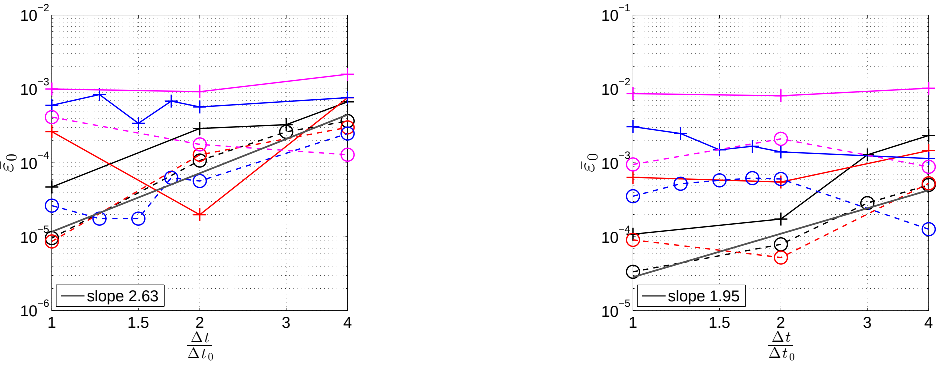
In this chapter, the effect of the number of particles and the time step on the numerical error $\overline{\varepsilon}$ of DSMC was investigated within the framework of an axisymmetric jet that was previously used to study the convergence error in Chapter III. The cellwise numerical error was found to vary for both number density and velocity following a power law that is, as $C \times N^{-\alpha}$ with $0 < \alpha < 1.5$, where C and α are cell dependent and vary between n and V. The normalized cellwise numerical error of V is observed to be consistently an order of magnitude smaller than that of n. Large spatial variations are observed for the value of the numerical error throughout the domain, with low error zones having consistently smaller α values. The rather complicated spatial patterns of the error can be explained by the trajectory of particles in the simulation domain and the location of highly collisional zones. The average number of particles in a local cell gives no indication as to the value of the error in the cell although increasing it by raising the total number of particles in the simulation systematically reduces it. No direct correspondence is observed between the value of the cellwise normalized numerical error and the average number of particles in a particular cell. Accuracy for sampled quantities in a particular cell should not be tied to the average number of particles (e.g. 20) that it contains but rather to the total number of particles employed in the simulation, i.e. W_p. In practical terms, this means that the accuracy of a simulation should be gauged by comparing the results obtained for multiple particle counts. As was noted, advantage can be taken of the general power law scaling that was observed for the error to a priori determine the number of particles required for a desired level of accuracy. The ℓ_1 and ℓ_2 norms of the numerical error for the entire domain were found to scale as N^{-1} for both $\mu_1(n)$ and $\mu_1(V)$. This scaling is, however, only marginally useful with regard to ensuring simulation accuracy.

For relatively small particle counts, an increase in the time step value results in an increase or decrease of the cellwise numerical error depending on the location in the flow. The effect of an increase in the time step value is observed to result in a more monotonic increase of the value of the numerical error as the particle count is increased. No general scaling of the cellwise numerical error with Δt could be consistently observed although a power law type dependency is usually admissible for $\mu_1(V)$ at relatively high particle counts. The global ℓ_1 and ℓ_2 norms of the numerical error for both $\mu_1(n)$ and $\mu_1(V)$ are, however, observed to scale as Δt^2.



(a) Cell 1

(b) Cell 5



(c) Cell 9

(d) Cell 10

Figure 4.23: Normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1(n)$ $(+)$ and $\mu_1(V_x)$ $(\circ)$ with C512 in black, C256 in red, C64 in blue and C16 in magenta. The value of the slope indicated corresponds to that of linear curve fit of the error of $\mu_1(V_x)$ for C512.

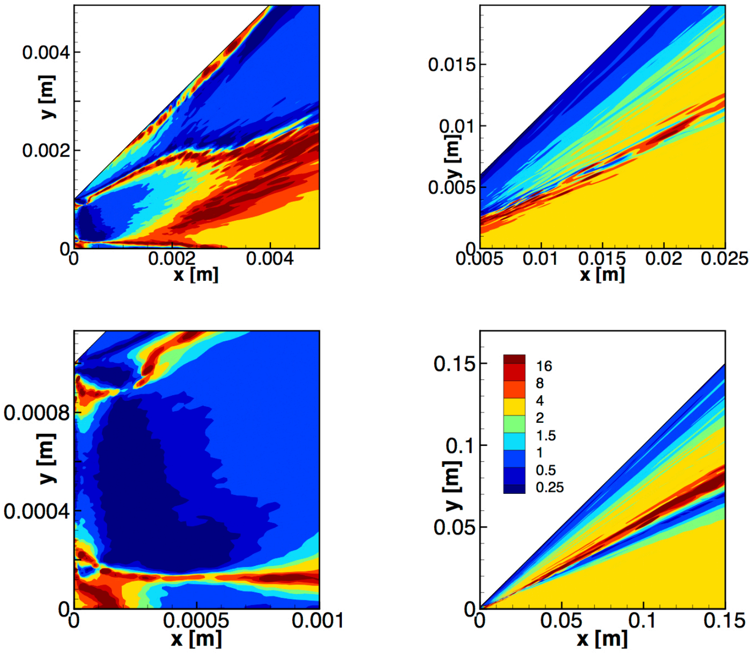


Figure 4.24: Ratio of the numerical error for $\mu_1\,(V_x)$ obtained with $\Delta t = 2 \times \Delta t_0$ with that obtained with $\Delta t = \Delta t_0$ for test case C16.

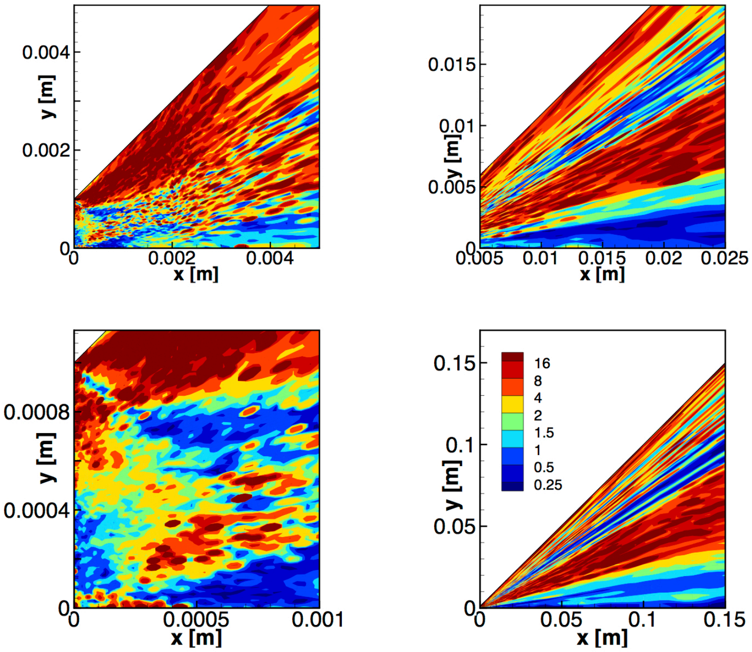


Figure 4.25: Ratio of the numerical error for $\mu_1\,(V_x)$ obtained with $\Delta t = 2 \times \Delta t_0$ with that obtained with $\Delta t = \Delta t_0$ for test case C256.

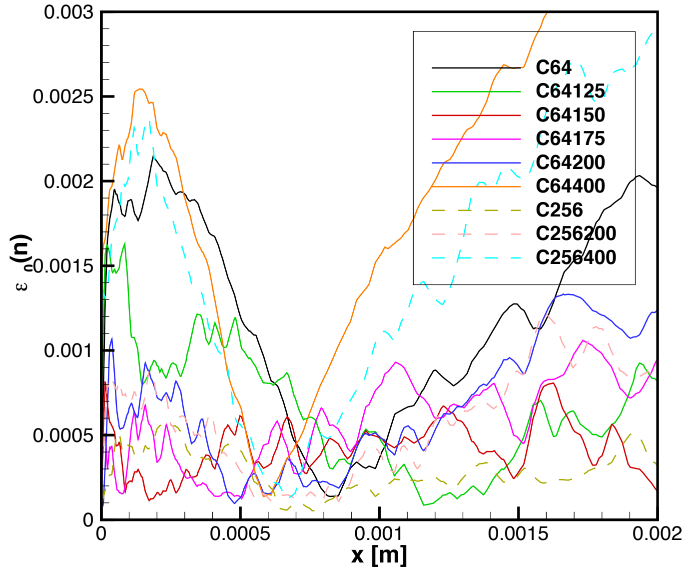


Figure 4.26: Influence of the time step on the normalized numerical error $\bar{\varepsilon}_0$ for $\mu_1\,(n)$.

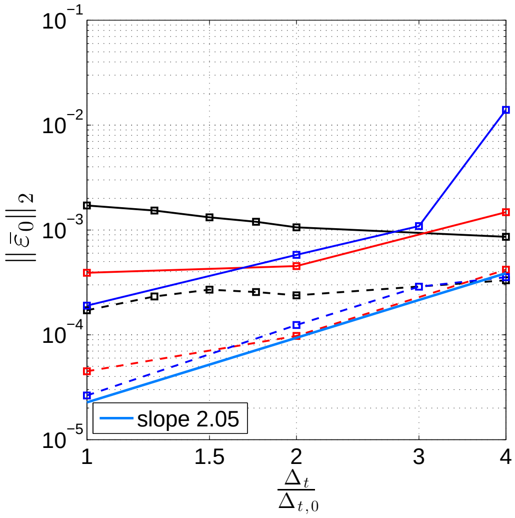


Figure 4.27: ℓ_2 norms of $\mu_1(n)$ (dashed lines and + symbols) and $\mu_1(V_x)$ (solid lines and $\circ$ symbols) as a function of the time step for test case C64 (black), C256 (red) and C512 (blue). The curve in light blue is a linear regression for the velocity error of case C512.

CHAPTER V

An adaptive procedure for the time step, cell and species weights for DSMC

5.1 Introduction

The DSMC (direct simulation Monte Carlo) method is today the most widely used simulation method for high speed rarefied flow [27]. Its accuracy and convergence is mainly determined by the number of particles employed in the simulation, the mesh size and the time step employed. Because of the formulation of DSMC [15], the number of particles in a cell is directly proportional to its number density while the time step of the simulation can be no more than the mean collision time so that the movement and collisions of particles can be decoupled. Similarly, collisions are only performed for particles located in the same cell, so that its characteristic length has to be less than the mean free path for this local assumption to be valid. This in turn signifies that the constraints which the numerical parameters of DSMC have to satisfy are inherently tied to the physics, particularly the number density field, of the test case being simulated. This however makes the efficient simulation of flows containing large variations in number density or large disparities between species number densities difficult. These drawbacks inherent to the DSMC method have led to a number of modifications to the basic DSMC algorithm that are in ubiquitous use today. The first is the use of spatially varying weights, particularly for axisymmetric simulations such as in (amongst many others) [91] [96] where the cell volume scales with the distance to the centerline, that allows enough particles to be present close to the centerline while reducing their number further away from it. The second improvement is the use of a spatially varying time step, and an adaptive procedure to calculate the optimum time step field as in, e.g., [115] [99] [19]. For multi-species flow, the use of relative weights, also called species weight, is widespread, such as in [23] [9] [5]. They allow to increase the number of particles simulating species with low number densities relative to others thereby allowing the use of fewer overall particles in the simulation. Spatially varying time steps, cell weights and relative weights are however almost never used simultaneously and whereas an adaptive procedure for the time step has been proposed before [115] [99], one for the cell weights and species relative weights has not. Relative weights are furthermore most often assumed to be spatially uniform thereby forfeiting some of the efficiency gains obtainable by having them vary through the computational domain. The aim of this chapter is to detail a formulation of an all-encompassing adaptive procedure for the time step, cell weights and relative weights where all are allowed to vary throughout space. Such a procedure greatly facilitates the conduct of efficient DSMC simulations by lessening the need for human inputs, such as e.g. running multiple simulations to determine optimum weight fields. The first part of the chapter describes the additional considerations that must be taken into account when using spatially varying weights or time steps in terms of particle movement and collisions including the potential deleterious effects of particle cloning. The adaptive procedure for the time step, cell weight and species relative weights is then detailed and its integration into existing DSMC codes discussed. A test case consisting of two counter-flowing axisymmetric jets at a Knudsen number of 0.015 is introduced which is used to illustrate the increased accuracy obtainable with the adaptive method compared to when only a spatially varying time step and weight are used.

5.2 DSMC Framework

5.2.1 DSMC with spatially varying time step and weights

5.2.1.1 Definitions

In the *canonical* version of DSMC, uniform values $W_{p,0}$ and Δt_0 for the cell weight and the time step are used throughout the simulation while all species are assigned the same relative weight of 1. In the following, a spatially variable time step $\Delta t\left(\vec{x}\right)$, cell weights $W_p\left(\vec{x}\right)$ and species relative weights $W_{\mathrm{rel},j}\left(\vec{x}\right)$ are considered. A non-dimensional time step $\tilde{\Delta} t\left(\vec{x}\right)$ and cell weight $\widetilde{W}_p\left(\vec{x}\right)$ are in turn respectively defined as

$$\Delta t\left(\vec{x}\right) = \Delta t_0 \ \widetilde{\Delta t}\left(\vec{x}\right) \tag{5.1}$$

$$W_p\left(\vec{x}\right) = W_{p,0} \ \widetilde{W}_p\left(\vec{x}\right) \ \widetilde{\Delta t}\left(\vec{x}\right) \tag{5.2}$$

The non-dimensionalized weight $\widetilde{W}_p\left(\vec{x}\right)$ is defined as such because it will later be more useful to consider the value of the normalized value of the weight divided by the non-dimensionalized time step, i.e. $\frac{W_p(\vec{x})}{W_{p,0} \ \widetilde{\Delta t}(\vec{x})}$ instead of $\widehat{W}_p\left(\vec{x}\right) \triangleq \frac{W_p(\vec{x})}{W_{p,0}}$. Cell-wise constant weights and time step are used, while a total of N_{spec} distinct species are present in the simulation, so that each cell i is characterized by $\widetilde{\Delta t}_i$, $\widetilde{W}_{p,i}$ and $\{W_{\mathrm{rel},j,i}\}_{j=1}^{N_{\mathrm{spec}}}$. As an example, using the previously defined terminology, the number density $n_{j,i}$ of species j inside cell i of volume V_i, when it contains $N_{j,i}$ particles of species j, is given by:

$$n_{j,i} = \frac{W_{p,0} \ W_{\mathrm{rel},j,i} \ \widetilde{W}_{p,i} \ \widetilde{\Delta t}_i \ N_{j,i}}{V_i}. \tag{5.3}$$

5.2.1.2 Particle Movement

When a computational particle of species j moves from one cell i, characterized $\widetilde{W}_{p,i}, W_{\mathrm{rel},j,i}$, $\widetilde{\Delta t}_i$ to cell $i+1$ with different weights $\widetilde{W}_{p,i+1}, W_{\mathrm{rel},j,i+1}$ and a different timestep $\widetilde{\Delta t}_{i+1}$, care must be taken to preserve the flux of particles between the two cells. The number density flux (i.e. the number of physical particles per unit area and time) from cell i to cell $i+1$ through their common face of area $S_{i+1/2}$ from the standpoint of cell i is given by:

$$\Phi^k_{i\to i+1|i}\left(n_j\right) = \frac{W_{\mathrm{rel},j,i}\ W_{p,i}}{\Delta t_i\ S_{i+1/2}}\ N^k_{j,i\to i+1|i} = \frac{W_{p,0}}{\Delta t_0\ S_{i+1/2}}\ W_{\mathrm{rel},j,i}\ \widetilde{W}_{p,i}\ N^k_{j,i\to i+1|i}\ , \tag{5.4}$$

where $N^k_{j,i\to i+1|i}$ designates the number of computational particles moving from cell i to cell $i+1$. Similarly, the number density flux of incoming particles from cell i into cell $i+1$ from the standpoint of cell $i+1$ is

$$\Phi_{i\to i+1|i+1}\left(n_j\right) = \frac{W_{\mathrm{rel},j,i+1}\ W_{p,i+1}}{\Delta t_{i+1}\ S_{i+1/2}}\ N^k_{j,i\to i+1|i+1} = \frac{W_{p,0}}{\Delta t_0\ S_{i+1/2}}\ W_{\mathrm{rel},j,i+1}\ \widetilde{W}_{p,i+1}\ N^k_{j,i\to i+1|i+1}\ . \tag{5.5}$$

As the number density flux from cell i to cell $i+1$ is a physical quantity, its value cannot dependent on the value of the numerical parameters used to express it or their spatial distribution, that is

$$\Phi^k_{i\to i+1|i}\left(n_j\right) = \Phi^k_{i\to i+1|i+1}\left(n_j\right)\ \forall\ \left\{W_{p,i'}\right\}_{i'}, \left\{W_{\mathrm{rel},j,i'}\right\}_{i'}, \left\{\Delta t_{i'}\right\}_{i'}\ , \tag{5.6}$$

so that:

$$\frac{W_{p,0}}{\Delta t_0\ S_{i+1/2}}\ W_{\mathrm{rel},j,i}\ \widetilde{W}_{p,i}\ N^k_{j,i\to i+1|i} = \frac{W_{p,0}}{\Delta t_0\ S_{i+1/2}}\ W_{\mathrm{rel},j,i+1}\ \widetilde{W}_{p,i+1}\ N^k_{j,i\to i+1|i+1}\ . \tag{5.7}$$

Eq. (5.7) can be satisfied for all weights and time step distributions if we allow the number of particles moving into cell $i+1$ from the standpoint of cell $i+1$ to depend on them, that is:

$$N^k_{j,i\to i+1|i+1} = \frac{\widetilde{W}_{p,i}\ W_{\mathrm{rel},j,i}}{\widetilde{W}_{p,i+1}\ W_{\mathrm{rel},j,i+1}} N^k_{j,i\to i+1|i}\ . \tag{5.8}$$

In terms of particles, this in turn signifies, that one particle (corresponding to $N^k_{j,i\to i+1|i} = 1$) entering cell $i+1$ from cell i, needs to give rise to $R_{j,i\to i+1} \triangleq \frac{\widetilde{W}_{p,i}\ W_{\mathrm{rel},j,i}}{\widetilde{W}_{p,i+1}\ W_{\mathrm{rel},j,i+1}}$ particles immediately upon entry into cell $i+1$. When $R_{j,i\to i+1} < 1$, the particle is discarded upon reaching the interface between the two cells with a probability of $R_{j,i\to i+1}$. When $R_{j,i\to i+1} > 1$, the original particle moves to cell $i+1$ and an additional ${}^{\mathrm{c}}N^k_{j,i\to i+1}$ particles are generated at the interface following a uniform distribution $\mathcal{U}\left(R_{j,i\to i+1} - 1, R_{j,i\to i+1}\right)$. The simplest approach and that adopted in this work is to give properties (i.e. position and velocity) that are identical to those of the original particle to all newly generated particles. When that scheme is used, the new particles are then referred to as *clones* and the particle creation process as *cloning.* The number of particles that are either generated or destroyed due to the spatial distribution of the weights and the time step is thus controlled by the spatial distribution of $\widetilde{W}_{p,i}$, instead of that of $W_{p,i}$. To be able to more closely control the distribution of $\widetilde{W}_{p,i}$, the adaptive procedure presented in Section 5.3 is thus formulated to adapt the values of $\widetilde{W}_{p,i}$ (instead of $\widetilde{\Delta}t$ and $\hat{W}_{p,i}$ as might have been expected).

5.2.2 Cloning issues

The generation of clone particles at the interface between cells where $R_{j,i\to i+1} < 1$ does lead to several issues with consequences that will later become apparent in simulations.

5.2.2.1 Accuracy

It is no better in terms of accuracy for a particular cell to contain two identical particles on average than for it to contain only one. This is because, absent any collisions, cloning does not increase the flow of information traveling from the rest of the computational domain to a particular cell. It only amplifies information by creating more vectors to carry it (i.e. particles). This is the broad reason why increasing the average particle count in a cell through cloning, while striving to maintain the same average number of particles in all cells of the domain, never produces as accurate results as are obtained by increasing the overall total number of particles in the simulation so as to obtain the exact same average particle count in that cell. The situation does change when collisions occur as the collision of cloned particles with other particles does generate new information (i.e. new velocities for some of the cloned particles). This is the reason why the present adaptive procedure will tend to produce the greatest relative accuracy gains compared to a simulation employing uniform weights for the same computational cost for flows that are relatively collisional. Such collisions, however, have to occur between non-clone particles, as two identical particles have the same velocity and thus can never collide. In the limiting case of a fully free molecular flow, it will by its definition produce identical results to those obtained with uniform weights and time steps. In extremely collisional flows, the time between the creation of a cloned particle and its first collision is expected to be so short that the deleterious effects of introducing cloned particles should be very small.

5.2.2.2 Avalanche phenomenon

The relative absence of collisions and the use of decreasing weights to counterbalance a decrease in number density along one direction can both combine to create what we refer to as a *particle avalanche* phenomenon. It occurs when an ever increasing number of cloned particles are created as an increasingly large group of identical particles travels along a direction with decreasing weights. It can best be understood by considering a simplified one-dimensional cold gas flowing with a bulk velocity of V_x with a number density that decreases in the x direction. As a Gedankenexperiment, we consider a one-dimensional perfectly cold gas flow in which the number density drops as $\sim x^{-1}$ so that its velocity distribution function is given by $f\left(V_x\right) = n_0 \left(\frac{x_0}{x}\right) \delta\left(V_x - V_0\right)$, where δ is Dirac's delta function and $V_0 > 0$. The gas flow is discretized with a one dimensional mesh from $x = 0$ to $x = L$ with uniform spacing $\Delta x = \frac{L}{N_c}$ using N_c cells each of volume V and constant time step Δt_0 is used (so that $\hat{W}_p = \tilde{W}_p$). The weights of the cells are "naively" chosen so as to maintain a constant average number of particles $\overline{N}_0$ in all cells of the domain despite the drop in number density in the x direction. As $\underline{\underline{\mathrm{n}}}_i = \frac{W_{p,i}\ \overline{N}_i}{V}$, the weights are given by $W_{p,i} = \frac{\underline{\underline{\mathrm{n}}}_i V}{\overline{N}_0}$ with the cell-averaged number density defined as $\underline{\underline{\mathrm{n}}}_i \triangleq \frac{1}{\Delta_x} \int_{x_{i-1/2}}^{x_{i+1/2}} n_0 \left(\frac{x_0}{x}\right) dx = \frac{n_0 x_0}{\Delta x} \log\left(\frac{x_{i+1/2}}{x_{i-1/2}}\right)$. As was previously explained, due to the spatial variation of the cell weights, when a particle travels from cell i to cell $i+1$, $\frac{\widetilde{W}_{p,i}}{\widetilde{W}_{p,i+1}}$ particles are created on the $i+1$ side of the interface between the two cells. Thus if a single particle happens to make its way from cell 1 to cell i, $\frac{\widetilde{W}_{p,1}}{\widetilde{W}_{p,2}} \times \frac{\widetilde{W}_{p,2}}{\widetilde{W}_{p,3}} \times ... \times \frac{\widetilde{W}_{p,i-1}}{\widetilde{W}_{p,i}} = \frac{\widetilde{W}_{p,1}}{\widetilde{W}_{p,i}}$ identical particles are in turn obtained in cell i. Because cell i has a lower density than cell 1, such a large number of particles will only seldom be observed in cell i which will contain no particles for most time steps. The frequency of occurrence of such a packet of particles in cell i, can be obtained by basic probability considerations. We designate the event "particle $\mathbf{p}$ is in cell $i+1$ in at least some time step k " by $\omega_{p\in i}$ and the event "particle $\mathbf{p}$ is in cell $i+1$ in at least some time step k" by $\omega_{p\in i+1}$. Due to the simple nature of the flow considered, $\mathbf{p}$ has to pass through cell i to reach cell $i+1$ so that $P\left(\omega_{p\in i} \cap \omega_{p\in i+1}\right) = P\left(\omega_{p\in i+1}\right)$ and furthermore, by definition $P\left(\omega_{p\in i+1} | \omega_{p\in i}\right) = \frac{P(\omega_{p\in i} \cap \omega_{p\in i+1})}{P(\omega_{p\in i+1})}$ so that $P\left(\omega_{p\in i+1} | \omega_{p\in i}\right) = \frac{P(\omega_{p\in i+1})}{P(\omega_{p\in i})}$. It can readily be seen that $P\left(\omega_{p\in i+1} | \omega_{p\in i}\right) = \frac{\underline{\underline{\mathrm{n}}}_{i+1}}{\underline{\underline{\mathrm{n}}}_i}$ so that $P\left(\omega_{p\in i}\right) = P\left(\omega_{p\in i} | \omega_{p\in i-1}\right) \times P\left(\omega_{p\in i-1} | \omega_{p\in i-2}\right) \times ... \times P\left(\omega_{p\in 2} | \omega_{p\in 1}\right) = \frac{\underline{\underline{\mathrm{n}}}_i}{\underline{\underline{\mathrm{n}}}_{i-1}} \times \frac{\underline{\underline{\mathrm{n}}}_{i-1}}{\underline{\underline{\mathrm{n}}}_{i-2}} \times ... \times \frac{\underline{\underline{\mathrm{n}}}_1}{\underline{\underline{\mathrm{n}}}_2} = \frac{\underline{\underline{\mathrm{n}}}_i}{\underline{\underline{\mathrm{n}}}_1}$. Hence a packet of $\frac{\widetilde{W}_{p,1}}{\widetilde{W}_{p,i}}$ particles will occur with a probability of $\frac{\underline{\underline{\mathrm{n}}}_i}{\underline{\underline{\mathrm{n}}}_1}$, i.e. once every $\frac{\underline{\underline{\mathrm{n}}}_1}{\underline{\underline{\mathrm{n}}}_i}$ steps in cell i. Thus, cell i will indeed contain an average of $N_0 \times \frac{\underline{\underline{\mathrm{n}}}_i}{\underline{\underline{\mathrm{n}}}_1} \times \frac{\widetilde{W}_{p,1}}{\widetilde{W}_{p,i}} = N_0$ over the course of the simulation as the process described above will be repeated N_0 times for all the particles initially in cell 1. The occurrence of a large number of identical particles, all traveling together in the form of a packet which grows larger and larger with time as it travels down a path with decreasing weights as an attempt to increase the average particle count to balance a decrease in average number density will be referred to as a *avalanching* phenomenon in the following. If collisions with other non-identical particles of the same species or with other species are not prevalent enough to *break up* the packet of particles, the avalanching phenomenon will occur. This undesirable phenomenon will later be seen to limit the useful range of applicability of the adaptive procedure.

5.2.3 Average quantities

The adaptive procedure presented in this chapter aims to vary the value of the weights so that all cells contain the same approximate number of particles *on average* as the instantaneous number of particles fluctuates. The average value needs to be reasonably responsive to changes in particle counts that are caused by the application of the adaptive procedure so that an exponential moving average [138] is used for its calculation:

$$\overline{N}_{j,i}^{k} = R\ N_{j,i}^{k} + (1-R)\,\overline{N}_{j,i}^{k-1}, \tag{5.9}$$

where $N_{j,i}^{k}$ and $\overline{N}_{j,i}^{k}$ are, respectively, the number and the average number of particles of species j in cell i at time step k. A value of 0.001 for the relaxation factor R is used for this present work. The average number of collisions $\overline{N}_{coll}\,(a,b)$, i.e. the average number of collisions where a particle of species a sees the value of its velocity changed as a result of a collision with a particle of species b, occurring in the cell at each time step for each (a,b) species pair is also similarly calculated. It is used to calculate the average mean collision time in each cell as follows:

$$\overline{\tau}_{m}\,(a,b) = \frac{\Delta t\ N_b}{(1+\delta_{ab})\,\overline{N}_{coll}\,(a,b)}, \tag{5.10}$$

which designates the average time elapsed between two collisions for a particle of species a with one of species b where the velocity of the former is indeed changed. As will later be seen, we are most interested in the minimum value of $\overline{\tau}_m(a,b)$ over all pairs that corresponds to the two species that collide with the greatest frequency:

$$\overline{\tau}_m = \min_{a,b} \overline{\tau}_m(a,b) . \tag{5.11}$$

5.2.4 Collision rates

The expression for the number of potential collision pairs to consider between different species with distinct relative weights is given without derivation in [23] (Eq. (5) therein which has a typographic error). Because the derivation is not straightforward, it is repeated here by distinguishing between the cross section of the physical particles and that of the computational particles [118]. This allows a much more rigorous derivation and presentation of all DSMC quantities related to collisions. The DSMC procedure can be viewed and examined from many different viewpoints, e.g. as a stochastic process that mimics the Boltzmann equation [126] or by adopting a more physical viewpoint, as we choose to do here, as a computational gas composed of computational particles. The properties of the computational gas such as its average number density over a specified region of space differ from those of the actual gas. That is, all quantities related to the computational gas have to be suitably rescaled to obtain those of the physical gas that is being modeled. Following the arguments presented in p. 7-8 of [15], the collision frequency of a single particle of the physical gas a with particles of gas b at a number density of n_b is given by:

$$\nu_{ab}^{(1)} = n_b \overline{\sigma_{ab} c_r}, \tag{5.12}$$

where c_r is the relative velocity between particles and σ_{ab} is the cross section for elastic collisions between species a and b. The bar over a quantity designates its average value amongst all particles of the sample. The properties of the DSMC gas differ from those of the physical gas and are denoted with a tilde. In the simulation, species b has a number density of

$$\tilde{n}_b = \frac{n_b}{W_p W_{\text{rel},b}}, \tag{5.13}$$

as each *macro* or *computational* particle represents $W_p \times W_{\text{rel},b}$ actual particles. Similarly, the single particle collision rate of the numerical gas is denoted by $\tilde{\nu}_{ab}^{(1)}$. The particles of the numerical gas have a collisional cross section that is different than those of the actual physical particles. It is denoted by $\tilde{\sigma}_{ab}$ to distinguish from σ_{ab}. Equation (5.12) correspondingly yields the single numerical particle collision rate:

$$\tilde{\nu}_{ab}^{(1)} = \tilde{n}_b \overline{\tilde{\sigma}_{ab} \tilde{c}_r}. \tag{5.14}$$

The DSMC procedure is based on the fundamental assumption that all particle-carried quantities (e.g. velocity, position, internal energy) are identical between the physical and computational gases. They thus share the same velocity, i.e. $\tilde{c}_r = c_r$ as well as the same collision rate. Thus, equaling Eqs. (5.12) and (5.14):

$$n_b \overline{\sigma_{ab} c_r} = \frac{n_b}{W_{\text{rel},b}\ W_p}, \overline{\tilde{\sigma}_{ab} c_r}, \tag{5.15}$$

the computational cross section $\tilde{\sigma}$ is found to be given by:

$$\tilde{\sigma}_{ab} = W_{\text{rel},b}\ W_p\ \sigma_{ab}. \tag{5.16}$$

The probability of a collision between a moving particle a' and a target particle b' belonging respectively to species a and b, both located inside the same cell of volume V can be interpreted as being given by the ratio of the volume of the cylinder engendered by the collision cross section through the relative motion of both particles during Δt to the volume of the cell:

$$P_{coll}\left(a', b'\right) = \frac{\tilde{\sigma}_{ab}\left(a', b'\right) c_r\left(a', b'\right) \Delta t}{V} \ . \tag{5.17}$$

Collisions between particle pairs in a cell at a given time step are independent events, that is the probability of a collision occurring for a given $\{a', b'\}$ pair is independent of those of all other pairs in the cell. Thus, the number of collisions that occurs at a given time step is obtained by summing the probability of occurrence of each potential collision:

$$N_{\text{coll}}\left(a, b\right) = \frac{1}{1+\delta_{ab}} \sum_{a'=1}^{N_a} \sum_{b'=1}^{N_b} P_{\text{coll}}\left(a', b'\right) = \frac{1}{1+\delta_{ab}} \sum_{a'=1}^{N_a} \sum_{b'=1}^{N_b} \frac{\tilde{\sigma}_{ab}\left(a', b'\right) c_r\left(a', b'\right) \Delta t}{V}. \tag{5.18}$$

Defining the average value of the cross section times the velocity amongst the $N_a \times N_b$ pairs as:

$$\overline{\tilde{\sigma}_{ab} c_r} \triangleq \frac{1}{N_a N_b} \sum_{a'=1}^{N_a} \sum_{b'=1}^{N_b} \tilde{\sigma}_{ab}\left(a', b'\right) c_r\left(a', b'\right), \tag{5.19}$$

by using Eq. (5.16), Eq. (5.18), can be rewritten as:

$$N_{\text{coll}}\left(a, b\right) = \frac{1}{1+\delta_{ab}} \frac{\overline{\tilde{\sigma}_{ab} c_r} \Delta t}{V} N_a N_b = \frac{N_a \ n_b}{1+\delta_{ab}} \overline{\sigma_{ab} c_r} \ \Delta t. \tag{5.20}$$

This gives an expression for the number of collisions that has to occur during Δt amongst the particles of the numerical gas so that Eq. (5.15) is satisfied. To increase computational efficiency, the DSMC procedure only considers potential collisions amongst a fraction of those collision pairs, i.e. $N_{\text{pairs}}\left(a, b\right) = \frac{\left(\sigma_{ab} c_r\right)_{\max}}{\overline{\tilde{\sigma}_{ab} c_r}} \times N_{\text{coll}}\left(a, b\right)$:

$$N_{\text{pairs}}\left(a, b\right) = \frac{N_a \ n_b}{1+\delta_{ab}} \left(\sigma_{ab} c_r\right)_{\max} \ \Delta t. \tag{5.21}$$

while suitably increasing the collision probability by defining it as $P_{coll} = \frac{(\sigma c_r)}{(\sigma c_r)_{\max}}$ so that Eq. (5.20) is indeed obtained on average. In the case where $W_{\text{rel},a} \neq W_{\text{rel},b}$, the number of pairs to consider for collision, i.e. Eq. (5.21), has to be altered as the velocity of both particles cannot simultaneously change to conserve average momentum during the collision. This issue is addressed in the next two sections that culminate in the derivation of a form of Eq. (5.20), i.e. Eq. (5.36), that takes into account the collision dynamics of particles with different relative weights.

5.2.5 Collision Mechanics with relative weights

In the absence of species relative weights and in the actual physical gas, the properties of both participating particles change after a collision. In the numerical gas, however, when particles with different relative weights participate in a collision, their properties cannot both equally be affected by it. If they were, the overall translational energy of particles and momentum would not be conserved on average in the gas and the effect of collisions on the species with the largest weight from that with the lowest would be systematically overestimated. In the following, we present the collision scheme that is most commonly used to handle collisions between particles with different relative weights which was first introduced in [13] p. 171. Because the scheme does not conserve energy, an energy-conserving collision scheme was subsequently proposed in [23]. The latter was for instance later used in [150] where a detailed description of the impact of relative weights on the modeling of chemical reactions via DSMC is given. An energy conserving scheme is however of limited utility for the test case that will later be studied as the residence time of particles inside the domain is relatively short with very few undergoing multiple collisions and is therefore not implemented. Non energy-conservation is however much more important for closed systems, as discussed in [23], such as homogeneous heat baths where boundary conditions are not constantly resupplying new particles (and thus energy). Considering a collision between a particle of species a with weight $W_{\mathrm{rel},a}$ and one of species b with weight $W_{\mathrm{rel},b}$, the pre-collision velocities of both particles are designated by $\vec{v}_a$ and $\vec{v}_b$ and their post-collision velocities are designated by $\vec{v}_a'$ and $\vec{v}_b'$. In the case where $W_{\mathrm{rel},a} = W_{\mathrm{rel},b}$, the post-collision velocities are denoted respectively by $\vec{v}_a^*$ and $\vec{v}_b^*$ and are obtained by the application of the DSMC collision operator C_{coll} (i.e. the variable hard sphere model in present study) that is:

$$\left(\vec{v}_a^*, \vec{v}_b^*\right) = C_{\mathrm{coll}}\left(\vec{v}_a, \vec{v}_b\right) \tag{5.22}$$

with

$$\begin{cases} \vec{v}_a' = \vec{v}_m + \frac{m_b}{m_a+m_b}\vec{v}_r^* \\ \vec{v}_b' = \vec{v}_m - \frac{m_a}{m_a+m_b}\vec{v}_r^* \end{cases} \tag{5.23}$$

where the center of mass velocity $\vec{v}_m$ is given by $\frac{m_a\vec{v}_a+m_b\vec{v}_b}{m_a+m_b}$ while the expression for the post-collision relative velocity $\vec{v}_r^* \triangleq \vec{v}_a^* - \vec{v}_b^*$ as a function of $\vec{v}_a$ and $\vec{v}_b$ for the VHS model is given in [15]. In order for the gas to conserve overall momentum during the collision, the following has to be satisfied:

$$m_a W_{\mathrm{rel},a}\vec{v}_a + m_b W_{\mathrm{rel},b}\vec{v}_b = m_a W_{\mathrm{rel},a}\vec{v}_a' + m_b W_{\mathrm{rel},b}\vec{v}_b'. \tag{5.24}$$

As C_{coll} conserves momentum, $m_a\vec{v}_a + m_b\vec{v}_b = m_a\vec{v}_a^* + m_b\vec{v}_b^*$, Eq. (5.24) can be rewritten as:

$$W_{\mathrm{rel},b}\left(m_a\vec{v}_b^* + m_b\vec{v}_b^*\right) + \left(W_{\mathrm{rel},a} - W_{\mathrm{rel},b}\right) m_a\vec{v}_a = m_a W_{\mathrm{rel},a}\vec{v}_a' + m_b W_{\mathrm{rel},b}\vec{v}_b' \tag{5.25}$$

Because, we do not require conservation of energy this equation is the only one that must be satisfied by the post collision velocities $\vec{v}_a'$ and $\vec{v}_b'$. They can thus be defined with a certain latitude. The most natural choice is to let the particle with the lowest relative weight be *fully* influenced by the collision, that is to assign it $\vec{v}^*$ as its post-collisional velocities. Using that assumption, Eq. (5.25) can readily be shown to admit the following two solutions:

$$\begin{cases} \vec{v}_a^{\,\prime} = \left(1 - \frac{W_{\mathrm{rel},b}}{W_{\mathrm{rel},a}}\right) \vec{v}_a + \frac{W_{\mathrm{rel},b}}{W_{\mathrm{rel},a}} \vec{v}_a^{\,*} \\ \vec{v}_b^{\,\prime} = \vec{v}_b^{\,*} \end{cases} \quad (5.26) \qquad \begin{cases} \vec{v}_a^{\,\prime} = \vec{v}_a^{\,*} \\ \vec{v}_b^{\,\prime} = \left(1 - \frac{W_{\mathrm{rel},a}}{W_{\mathrm{rel},b}}\right) \vec{v}_b + \frac{W_{\mathrm{rel},a}}{W_{\mathrm{rel},b}} \vec{v}_b^{\,*} \end{cases} \quad (5.27)$$

The solution of either Eq. (5.26) or Eq. (5.27) are acceptable to conserve momentum during the collision. However, in order to properly simulate the effect of collisions, particles from the entire range of the velocity distribution function must be potentially allowed to exist in the flow. In particular, specifying velocities after each collision following Eq. (5.26) or (5.27), especially when $\frac{W_{\mathrm{rel},b}}{W_{\mathrm{rel},a}} \ll 1$ only produces a $\vec{v}_a^{\,\prime}$ or $\vec{v}_b^{\,\prime}$ that are very similar to their pre-collision values so that almost no “real” collision actually takes place in the flow. Hence, a solution to this, is to only satisfy Eq. (5.27) or (5.26) on average, that is in a probabilistic sense by allowing a collision to occur for the particles with the greatest relative weight with a probability given by the ratio of the weights, i.e.:

$$\text{If} \;\; W_{\mathrm{rel},a} < W_{\mathrm{rel},b} \qquad \vec{v}_a^{\,\prime} = \vec{v}_a^{\,*} \qquad \vec{v}_b^{\,\prime} = \begin{cases} \vec{v}_b^{\,*} \text{ with } \mathrm{P} = \frac{W_{\mathrm{rel},a}}{W_{\mathrm{rel},b}} \\ \vec{v}_b \text{ otherwise} \end{cases} \tag{5.28}$$

and

$$\text{If} \;\; W_{\mathrm{rel},a} > W_{\mathrm{rel},b} \qquad \vec{v}_a^{\,\prime} = \begin{cases} \vec{v}_a^{\,*} \text{ with } \mathrm{P} = \frac{W_{\mathrm{rel},b}}{W_{\mathrm{rel},a}} \\ \vec{v}_a \text{ otherwise} \end{cases} \qquad \vec{v}_b^{\,\prime} = \vec{v}_b^{\,*}. \tag{5.29}$$

excluding the case where $W_{\mathrm{rel},b} > W_{\mathrm{rel},a}$ for Eq. (5.26) and the reverse for Eq. (5.27) to avoid generating large unphysical velocities after collision. As shown by Eqs. (5.28) and (5.29), a collision does not always result in a change of velocity for both particles when these have different relative weights. One must therefore define a *real* collision for a given particle as a collision in which its velocity is changed. The probability of such an event occurring during a collision of a particle from species a with one from species b is denoted by P_{ab}, while P_{ba} refers to the probability that the particles from the b species sees its velocity change in an encounter with one of a. Eqs. (5.28) and (5.29) respectively yield:

$$W_{\mathrm{rel},a} < W_{\mathrm{rel},b} \qquad\qquad P_{ab} = 1 \qquad\qquad P_{ba} = \frac{W_{\mathrm{rel},a}}{W_{\mathrm{rel},b}} \tag{5.30}$$

and

$$W_{\mathrm{rel},a} < W_{\mathrm{rel},b} \qquad\qquad P_{ab} = \frac{W_{\mathrm{rel},b}}{W_{\mathrm{rel},a}} \qquad\qquad P_{ba} = 1. \tag{5.31}$$

so that one can remark that the identity $P_{ab} = \frac{W_{\mathrm{rel},b}}{W_{\mathrm{rel},a}} P_{ba}$ is always satisfied.

5.2.6 Collision rates with relative weights

When the mechanism detailed in the last section is used, a collision between two particles results in a real collision for one or both particles (as opposed to the case where both particles have the same relative weight, where a collision between two particles always results in a real collision for both). One must therefore distinguish between the real collision cross section that was previously introduced, denoted $\tilde{\sigma}_{ab}$, and the virtual collision cross section $\hat{\sigma}_{ab}$. The real collision cross section, $\tilde{\sigma}_{ab}$, is the cross section of the process "{Particle a collides with particle b AND its velocity is changed}" as opposed to $\hat{\sigma}_{ab}$ which is that of "{Particle a collides with particle b}". Both are numerical cross sections and identical in the absence of relative weights. Because a collision of a particle of species a with one of b has a probability P_{ab} of leading to a real collision:

$$\tilde{\sigma}_{ab} = P_{ab} \hat{\sigma}_{ab}. \tag{5.32}$$

Hence using $\hat{\sigma}$, the single particle collision rate of Eq. (5.14) can be rewritten, using Eq. (5.16) as:

$$\tilde{\nu}_{ab}^{(1)} = \tilde{n}_b \overline{\tilde{\sigma}_{ab} c_r} = P_{ab} \tilde{n}_b \overline{\hat{\sigma}_{ab} c_r} = n_b \overline{\sigma_{ab} c_r} \tag{5.33}$$

so that using Eq. (5.13), the virtual collision cross section $\hat{\sigma}_{ab}$ is found to be:

$$\hat{\sigma}_{ab} = \frac{W_p W_{\text{rel},b}}{P_{ab}} \sigma_{ab}. \tag{5.34}$$

By analogy with Eq. (5.20), which gives an expression for the number of real collisions that has to take place in a cell, the number of virtual numerical collisions in a cell is given by:

$$\hat{N}_{\text{coll}}\left(a,b\right) = \frac{\overline{\hat{\sigma}_{ab} c_r}\ \Delta t}{V\left(1+\delta_{ab}\right)} N_a N_b. \tag{5.35}$$

which can be rewritten using Eq. (5.34) as:

$$\hat{N}_{\text{coll}}\left(a,b\right) = \frac{W_p W_{\text{rel},b}\ \overline{\sigma_{ab} c_r}\ \Delta t}{\left(1+\delta_{ab}\right) P_{ab} V} N_a N_b. \tag{5.36}$$

This only gives an expression for the number of collision pairs between species a and b to consider from the standpoint of species a, so that the total number of collision pairs to consider between species a and b is given by:

$$\hat{\mathrm{N}}_{\text{coll}}\left\{a,b\right\} = \frac{1}{1+\delta_{ab}} \left[\hat{N}_{\text{coll}}\left(a,b\right) + \hat{N}_{\text{coll}}\left(b,a\right)\right]\ , \tag{5.37}$$

with $\hat{N}_{\text{coll}}$ being defined by Eq. (5.36). Equation (5.36) corresponds to the "sophisticated DSMC" pair selection scheme introduced in [18] (with $N_b = \overline{N}_b - 1$ if $a = b$), while the "no time counter" pair selection scheme [15] is obtained by letting $N_b = \overline{N}_b$. Our implementation of the DSMC method with relative weights thus amounts to selecting the number of particle pairs indicated by Eqs. (5.37) and (5.36) with $\overline{\sigma_{ab} c_r} = \left(\sigma_{ab} c_r\right)_{\max}$ for all species pairs. Amongst those selected particle pairs, (virtual) collisions are assumed to occur with a probability of $\hat{P}_{\text{coll}} = \frac{(\sigma_{ab} c_r)}{(\sigma_{ab} c_r)_{\max}}$, in which case they lead to the application of the collision mechanics described by Eq. (5.28) or (5.29). The latter always results in a real collision, i.e. a velocity change, for the particle with the lowest relative weight while a real collision only occurs with a probability given by the ratio of the relative weights for the other particle.

5.3 Formulation of the adaptive procedure

5.3.1 Calculation of the time step

5.3.1.1 DSMC requirements

The DSMC procedure is based on the decoupling of the two main mechanisms governing the dynamics of a rarefied gas: the ballistic motion of particles through physical space and their collisions. This basic assumption is an approximation as both are coupled in reality which in turn introduces a splitting error. In terms of the modeling, this means that collisions should really impact the "move" part of the procedure as they cannot be fully assumed to take place instantaneously at a fixed spatial location. An often cited guideline to mitigate this error is to ensure that the time step is less than the mean collision time. Because this error most likely follows a power law, i.e. $(\Delta t)^{\eta}$ with $\eta > 0$ as discussed in Chapter IV it can never be fully eliminated and it is already significant for values of Δt below τ_{mct} as no distinct threshold phenomenon occurs when $\Delta t = \tau_{\text{mct}}$. In the adaptive procedure, the value of the time step is thus set equal to a fraction of the mean collision time $\Xi_{\tau_{\text{m}}} < 1$:

$$\Delta t = \Xi_{\tau_{\text{m}}} \tau_m, \tag{5.38}$$

where τ_m is defined according to Eq. (5.11). Particles should furthermore avoid crossing multiple cells during their movement. That is, they should not overfly a cell without having had a chance to collide with the particles occupying it. When this undesirable situation occurs, the move procedure prevents the proper modeling of collisions by spatially biasing the collision partner population of the particles being moved. An approximate measure of the average number of neighboring cells traversed by the particles contained inside cell i during Δt can be given by:

$$K_i = \frac{\langle v \rangle_i \, \Delta t_i}{\langle \Delta x \rangle_i}, \tag{5.39}$$

where $\langle v \rangle$ designates the average velocity magnitude of the particles of cell i and $\langle \Delta x \rangle_i$ designates a characteristic length of the cell which can be taken as the value of its minimum edge length. We can prevent particles from crossing more than one cell on average during Δt by imposing:

$$K_i \leq \beta_{K,\max} \;\; \text{with} \;\; \beta_{K,\max} < 1 \; . \tag{5.40}$$

Whereas the two previous constraints provide upper bounds for the time step value, the latter should however also not be too small. This is because the autocorrelation spectrum $\rho\,(k', \Delta t)$ of sampled quantities as a function of the number of time steps k', for a given time step Δt is such that $\rho\,(k', \Delta t) = \rho_0\,(k' \; \Delta t_0)$ (c.f. Chapter III), so that reducing the time step increases the correlation between sampled quantities. This is turn reduces the rate of convergence of sampled quantities thereby requiring more sampling steps and hence making the simulation less computationally efficient. Without calculating the correlation between samples, this can for instance be accomplished by specifying a minimum bound for K_i:

$$\beta_{K,\min} \leq K_i \;\; \text{with} \;\; \beta_{K,\min} > 0. \tag{5.41}$$

The two previous conditions, i.e. Eqs. (5.40) and (5.41) can thus be summarized as:

$$\beta_{K,\min} < \frac{\langle v \rangle_i \, \Delta t_i}{\Delta x_i} < \beta_{K,\max} \ , \tag{5.42}$$

which has to be satisfied for all cells of the domain. The exact values of the $\beta_{K,\min}$ and $\beta_{K,\max}$ bounds must ultimately be determined by trial and error. Good results have been obtained for a variety of flows with $\beta_{K,\min} = 0.05$ and $\beta_{K,\max} = 0.1$ which are the values that are used in the following.

5.3.1.2 Implementation considerations

Applying the adaptive procedure essentially amounts to calculating $\left\{\Delta t_i^{k+1}\right\}_{i=1}^{N_c}$ at time step t^k so that Eqs. (5.38) and (5.42) are satisfied and updating the simulation to account for the new time step. Because the adaptive procedures for the time step (and the weights) are applied during the march to steady state of the flow during which the latter greatly changes, it must carefully be applied so as not to excessively interfere with it. For that the reason, as will be discussed in Section 5.3.4, the weights and time step are only updated every few thousands time steps to allow the flow to sufficiently develop between updates. The range of the time step is further restricted to prevent too low or high values from being used that would hinder the development of the flow:

$$\Delta t_{\min} < \Delta t_i^{k+1} < \Delta t_{\max} \ . \tag{5.43}$$

The value of these two bounds obviously depends on that of Δt_0, They should be respectively small and large enough so that constraints of Eqs. (5.38) and (5.42) can be enforced. The magnitude of the increase or decrease of the time step is further restricted for each cell by imposing:

$$\frac{1}{\Xi_{\Delta t,\max}} \leq \frac{\Delta t_i^{k+1}}{\Delta t_i^k} \leq \Xi_{\Delta t,\max} \quad \text{with} \quad \Xi_{\Delta t,\max} > 1, \tag{5.44}$$

which prevents possible oscillations in the overall number of particles due to a too tight coupling of the adaptive procedure with the flow. A moderate value should thus be used for $\Xi_{\Delta t,\max}$ such as e.g. 2.

We further desire a time step field that is relatively smooth, that is without abrupt increases or decreases between adjacent cells. This is particularly necessary during the development period of the flow where large disparities between time steps are observed between zones with and without particles. Too sudden time step variations along the trajectory of particles furthermore degrades accuracy. A smooth time step field is obtained by performing a Laplacian smoothing, denoted by $L\left(\cdot\right)$, of the time step field Δt^{k+1}. After sequentially applying Eqs. (5.38), (5.42), (5.44) and (5.43) to obtain Δt^{k+1}, the smoothed time step field $\left\{\widehat{\Delta t}_i^{k+1}\right\}_{i=1}^{N_c} = L\left(\left\{\Delta t_i^{k+1}\right\}_{i=1}^{N_c}\right)$ is determined according to the following:

$$\begin{cases} \check{\Delta} t_i^0 & = \Delta t_i^{k+1} \\ \check{\Delta} t_i^l & = \frac{1}{N_{b,i}+1}\left(\check{\Delta} t_i^{l-1} + \sum_{j=1}^{N_{b,i}} \check{\Delta} t_{J_{i,j}}^{l-1}\right) \text{ with } 1 \leq l \leq N_L \\ \widehat{\Delta t}_i^{k+1} & = \check{\Delta} t_i^{N_L} \end{cases} \tag{5.45}$$

The new average at step l of the smoothing is obtained by averaging the value of the time step over the $N_{b,i}$ neighbors of cell i, $J_{i,j}$ designating the index of the cell that is the jth neighbor of cell i. The averaging is repeated multiple times so that a total of N_L smoothing iterations are performed.

The final step of the procedure is to suitably update the number of particles and the properties of the flow so that all flow quantities when rescaled with the new time step value are not altered. Because of the definition of number density following Eq. (5.3), the number of particles in cell i of all j species needs to be altered to keep the number density of the cell constant:

$$N_{j,i}^{k+1} = N_{j,i}^{k} \, \frac{\Delta t_i^k}{\Delta t_i^{k+1}} \quad \forall j \ . \tag{5.46}$$

When, $\frac{\Delta t_i^k}{\Delta t_i^{k+1}} > 1$, the additional particles required are obtained by cloning the particles that are already present in the cell. The average number of particles in the cell is similarly updated while no change is required for the average number of collisions. All the steps outlined above are briefly summarized below and the numerical values of all the parameters such that were used in the test case of Section 5.4 are specified. The integration of the time step procedure within a DSMC code and its frequency of application is discussed in Section 5.3.4.

Pseudo code of the time step update

```
For all cells, $i = 1, ..., N_c$, do:
    i) Calculate $\tau_{m,i}$ with Eqs. (5.10) and (5.11) and enforce Eq. (5.38) with $\Xi_{\tau_\mathrm{m}} = 0.2$.
    ii) Calculate $K_i$ with Eq. (5.39) and enforce Eq. (5.42) with $\beta_{K,\mathrm{min}} = 0.05$ and $\beta_{K,\mathrm{max}} = 0.1$.
    iii) Limit variation of $\Delta t_i^{k+1}$ with Eq. (5.44) using $\Xi_{\Delta t,\mathrm{max}} = 2$.
    iv) Check that $\Delta t_i^{k+1}$ is within the prescribed bounds of $[\Delta t_{min}, \Delta t_{max}]$[a]
    iv) Smooth $\Delta t_i^{k+1}$ as in Eq. (5.45), using $N_L = 20$ smoothing steps.
    v) Clone/destroy particles as dictated by Eq. (5.46) and update running averages.
end
```

---

[a] Respectively small and large enough so that i) and ii) can be enforced

5.3.2 Calculation of cell weight

5.3.2.1 Motivation

The accuracy of sampled quantities in a cell of a DSMC simulation depends on both the total number of particles in the simulation and the average number of particles in the cell. For fairly homogeneous test cases, such as channel flow, there is a direct connection between the average number of particles in a cell and the accuracy of sampled quantities so that the "20 particles per cell" [15] typical guideline may be used to guarantee accurate results. For more complex flow, such as axisymmetric jets, there is no direct relationship between the average number of particles per cell and the accuracy of sampled quantities as discussed in Chapter IV, although an increase in the average number of particles in a cell is always associated with an increase in accuracy. Accurate results can, however, never be obtained in a cell with too few, e.g. < 5 particles, so that one may ideally desire a minimum average number of particles in all cells of the domain as a necessary condition for accuracy.

Due to large number density gradients in the flow being simulated and/or large cell volume disparities in the mesh employed, DSMC users often employ spatially varying weights in their simulations to increase computational efficiency. This is in particular necessary for axisymmetric flows to obtain accurate results close to the centerline while avoiding too many particles in cells with greater volume further away from the centerline. Except for simple situations, such as single axisymmetric jets, the optimum shape of the distribution of weights to adopt is not obvious. Very often, users have to resort to a trial and error approach by running multiple simulations to determine a "good" weight distribution often with the implicit goal of trying to obtain a constant average number of particles in all cells.

Building on the two previous ideas, the goal of the procedure presented in this section is thus to obtain a user-defined constant $N_{p,\mathrm{want}}^{\mathrm{tot}}$ particles per cell on average for all the cells of the domain. When N_{spec} species are present in the flow and the adaptive procedure for the relative weights is used, such as in the following, $N_{p,\mathrm{want}}^{\mathrm{tot}} = N_{\mathrm{spec}} \times N_{p,\mathrm{want}}$ is chosen so as to obtain $N_{p,\mathrm{want}}$ particles per species per cell on average.

Having tested various implementations of the adaptive procedure, the best results have been obtained when the procedure for the relative weights is separated from that for the cell weight. This furthermore has the advantage of allowing the use of the procedure without relative weights. It is, however, possible to combine the procedures for the cell weight and the relative weights into a single one.

5.3.2.2 Implementation

The update equation for the value of the weight $\widetilde{W}_p$ is obtained by noting that the total average number of particles in cell i, $\overline{N}_i^{\text{tot},k} \triangleq \sum_{j=1}^{N_{\text{spec}}} \overline{N}_{j,i}^k$ is proportional to $\widetilde{W}_{p,i}^k$, so that to have $\overline{N}_i^{\text{tot},k+1} = N_{p,\text{want}}^{\text{tot}}$, we must impose:

$$\widetilde{W}_{p,i}^{k+1} = \frac{\overline{N}_i^{\text{tot},k}}{N_{p,\text{want}}^{\text{tot}}} \widetilde{W}_{p,i}^k \ . \tag{5.47}$$

Similar conditions as for the time step are imposed, preventing too large increases or decreases while enforcing a lower and an upper bound with

$$\frac{1}{\Xi_{\widetilde{W}_p,\max}} \leq \frac{\widetilde{W}_{p,i}^{k+1}}{\widetilde{W}_{p,i}^k} \leq \Xi_{\widetilde{W}_p,\max} \text{ with } \ \Xi_{\widetilde{W}_p,\max} > 1 \tag{5.48}$$

and

$$\widetilde{W}_{p,\min} < \widetilde{W}_{p,i}^{k+1} < \widetilde{W}_{p,\max} \ . \tag{5.49}$$

As opposed to for the time step, it is imperative that the weight field be smooth in space and its variation gradual. This is to distribute the generation of cloned particles as equally as possible throughout space so as to promote the collisions of newly generated cloned particles with other particles. The aim of this is to *i)* Prevent the occurrence of the avalanche phenomenon which was discussed in Section 5.2.2.2 by breaking up packets of particles through collisions before they have the occasion to form *ii)* Improve the contribution of newly generated cloned particles to the accuracy of the simulation by having them collide so as make them carry new information into the rest of the flow.

The new weight field is thus also smoothed with a Laplacian smoothing operator. The smoothing, however, has to take the axisymmetric geometry into account. This is because axisymmetric flows present very strong variations in the radial direction in which the cell volume varies as $\sim y$. Furthermore, because particles move parallel to the axis of symmetry in most axisymmetric flows, it is more important for the weight field to be smooth in the longitudinal direction than in the radial direction. The banded weight structures commonly used for axisymmetric jets, as in [15] p. 374, are for instance not smooth in the radial direction. In order to obtain accurate simulation results close to the centerline, enough particles have to be present in that region which necessitates, as the volume scales $\sim y$, that very low weights be employed for that region. Using centerline weights for the entire domain, however, leads to an excessive number of particles in the cells at greater radial distances, as $N \sim V \sim y$, which is the reason the weights of such cells are increased. Because the region requiring low weights close to the centerline is relatively limited in extent (i.e. it consists of only a few cells) compared to the zone requiring higher weights, the result of the application of the smoothing procedure described by Eq. (5.45) would be for it to increase the weights of the centerline region so as to make weight variations smoother in the radial direction. This would in turn systematically lead to an insufficient number of particles in the cells close to the centerline. Possible solutions to the issue include using radial weights in the smoothing, so that $\left\{w_i \; \widetilde{W}_{p,i}\right\}_{i=1}^{N_c}$ is smoothed with for instance $w_i = \frac{1}{y_{c,i}}$ or the solution that is adopted in this work, which consists in excluding centerline cells from the smoothing thus forcing other neighboring cells to conform to their low values. Such a smoothing operator, which produces $\widehat{\widetilde{W}}_p$, the smoothed weight field, is defined as follows:

$$\begin{cases} \check{\widetilde{W}}^{0}_{p,i} & = \widetilde{W}^{k+1}_{p,i} \\ \check{\widetilde{W}}^{l}_{p,i} & = \begin{cases} \check{\widetilde{W}}^{l-1}_{p,i} & \text{if } i \text{ on the centerline} \\ \frac{1}{N_{b,i}+1}\left(\check{\widetilde{W}}^{l-1}_{p,i} + \sum_{j=1}^{N_{b,i}} \check{\widetilde{W}}^{l-1}_{p,J_{i,j}}\right) & \text{otherwise} \end{cases} \quad \text{with } 1 \leq l \leq N_L \\ \widehat{\widetilde{W}}^{k+1}_{p,i} & = \check{\widetilde{W}}^{N_L}_{p,i} \end{cases} \tag{5.50}$$

This smoothing operator increases the average number of particles in cells that are not on the centerline but in close proximity to it past $N_{p,\text{want}}$ which, however, represents a small computational penalty (in view of the increased accuracy afforded by the scheme). The adaptive weight procedure concludes by an update of the number of particles in each cell so that each maintains the same number density that it had with the old weight value:

$$N^{k+1}_{j,i} = N^{k}_{j,i}\,\frac{\widetilde{W}^{k}_{p,i}}{\widetilde{W}^{k+1}_{p,i}} \quad \forall j\ , \tag{5.51}$$

while the average numbers of particles and collisions are suitably updated. All the different steps of the weight algorithm are summarized in the pseudo code below.

Pseudo code of the $\widetilde{W}_p$ update procedure

```
For all cells, $i = 1, ..., N_c$, do:
    i) Calculate the new weight value $\widetilde{W}_{p,i}^{k+1}$ with Eq. (5.47)
    iii) Limit variation of $\widetilde{W}_{p,i}^{k+1}$ with Eq. (5.48) using $\Xi_{\widetilde{W}_p,\max} = 2$.
    iv) Check that $\widetilde{W}_{p,i}^{k+1}$ is within the prescribed bounds of $\left[\widetilde{W}_{p,\min}, \widetilde{W}_{p,\max}\right]$ [a]
    iv) Smooth $\widetilde{W}_{p,i}^{k+1}$ as in Eq. (5.50), using $N_L = 20$ smoothing steps.
    v) Clone/destroy particles as dictated by Eq. (5.51) and update averages.
end
```

5.3.3 Calculation of relative weights

The aim of the adaptive procedure for the relative weights is to determine N_{spec} different relative weight fields, $\{W_{\text{rel},j,i}\}_{j=1}^{N_{\text{spec}}}$, such that all cells of the flow contain $N_{p,\text{want},j}$ particles of species j. It essentially consists in applying the procedure that was previously presented for the weight to each individual j species. Due to the similarity of this procedure with that for the cell weights, it will only be presented in the form of the pseudo-code shown below.

---

[a] The value of the bounds depend on that of $W_{p,0}$ so are not specified here.

Pseudo code of the update procedure for the relative weights

```
For all cells, $i = 1, ..., N_c$, do:
    For all species, $j = 1, ..., N_{\text{spec}}$, do
        i) Calculate the new relative weight value:
            $W^{k+1}_{\text{rel},j,i} = \frac{\overline{N}^k_{j,i}}{N_{p,\text{want},j}} W^k_{\text{rel},j,i}.$    (5.52)
        ii) Limit variation of $W^{k+1}_{\text{rel},j,i}$ with Eq. (5.48) using $\Xi_{\widetilde{W}_p,\max} = 2$.
        iii) Check that $W^{k+1}_{\text{rel},j,i}$ is within the prescribed bounds of $[W_{\text{rel},j,\min}, 1]$ [a]
        iv) Check that $\widetilde{W}_p \times W^{k+1}_{\text{rel},j,i}$ is within the prescribed bounds of $\left[\left(\widetilde{W}_p\ W_{\text{rel},j}\right)_{\min}, \widetilde{W}_{p,\max}\right]$ [b]
        v) Smooth $W^{k+1}_{\text{rel},j,i}$ as in Eq. (5.45), using $N_L = 20$ smoothing steps.
        vi) Clone/destroy particles following
            $N^{k+1}_{j,i} = N^k_{j,i} \frac{W^k_{\text{rel},j,i}}{W^{k+1}_{\text{rel},j,i}}.$    (5.53)
        vii) Update the average numbers of particles and collisions.
    end
end
```

---

[a] The bounds are test case dependent although we always require that relative weights be less than 1 so that $W_{\text{rel},j,\max} = 1$.

[b] *ibid.*

5.3.4 Integration in a DSMC code

The time step and weights have to be updated multiple times as the flow develops from its initial conditions at time step $k = 0$ to its steady state. This is because of the coupling mechanism that exists between the weight and time step values and the flow. The particle properties and locations depend on the time step and weight values while these both depend on all the particles. The flow must further be allowed to sufficiently develop between each time step or weight change so that these in turn do not excessively impact the flow. Following these considerations, the solution that is retained is thus to alternatively apply every N_{adapt} steps either simultaneously the adaptive procedure for the time step and cell weight or the procedure for the relative weights. The adaptive procedure is applied until the flow reaches steady state, i.e. $k = N_{\text{stop}}$ at which point the sampling of flow quantities commences. The context and location of the adaptive procedures within a standard DSMC implementation is shown in the following pseudo-code:

Integration of the adaptive procedure within a DSMC code

```
For time steps, $k = 0, ..., N_{\text{step}}$, do:

        Every $N_{\text{adapt}}$ steps and if $k <$     }
        $N_{\text{stop}}$ do:                           }
            Update$\Delta t$()          and             }
            Update$W_p$()                               } Adaptive procedure
            or                                          }
            Update$W_{\text{rel}}$()                    }
        end                                             }
        For all cells, $i = 1, ..., N_c$, do:           }
          i)  CalculateCollisionPairs()                 }
         ii)  PerformCollisions()                       } "Standard" DSMC implementation
        iii)  MoveParticles                             }
        iii)  ApplyBCs()                                }
        end                                             }
end
```

The application of the procedure creates a sharp rise or drop in the total number of particles in the domain as particles are either created or destroyed to conform to the new time step and weight fields following Eqs. (5.46), (5.51) and (5.53). The magnitude of these discontinuities in the number of particles gradually decays as the flow and its weight and time fields approach steady state. A typical evolution of the number of particles produced by the adaptive procedure is shown in Fig. 5.1 for the test cases summarized in Table 5.2. The {weight+time step} procedure is applied every odd thousand of steps while that for the relative weights is applied every even thousand of steps, thus corresponding to $N_{\text{adapt}} = 1,000$ in the pseudo code above. For the following test case, the application of the adaptive procedure is stopped at $N_{\text{stop}} = 100,000$ at which point the sampling of flow quantities begins. Sampled quantities can, however, also be suitably rescaled so as to allow for sampling during the application of the adaptive procedure which we have chosen not to do in the following. As the adaptive procedure is relatively independent of the main DSMC algorithm, as shown by the pseudo code above, it is fairly easy to implement in an existing DSMC code or in a stand-alone code that would processes DSMC simulation result files. The computational cost of the procedure compared to that of the DSMC simulation is negligible as it is seldom applied, e.g. every 1,000 steps here, and only $\mathcal{O}\left(N_{\text{particles}}\right)$ at its worst (if every particle is cloned).

5.4 Application of the adaptive method

In order to illustrate the increased computational efficiency brought about by the use of the adaptive method, it is applied to a test case consisting of 3 species which is presented in the next section. Results of the application of the method are shown and the efficiency of the method is then evaluated by comparing it against a reference solution.

5.4.1 Test Case

The test case depicted in Fig. 5.2 consists of two jets of the same $D = 0.002$ m diameter separated by distance $\Delta L = 0.25$ m. Jet 1 emits species 1 and 3 while jet 2 emits species 2 for which respective boundary conditions are summarized in Table 5.1. This test case was chosen due to the difficulties that a traditional DSMC approach has to solve such flows. Because of spatial diffusion and the effects of collisions, it is indeed challenging to obtain accurate sampled quantities for a species in front of the jet opposite that from which it is emitted. Good resolution in such a zone is for

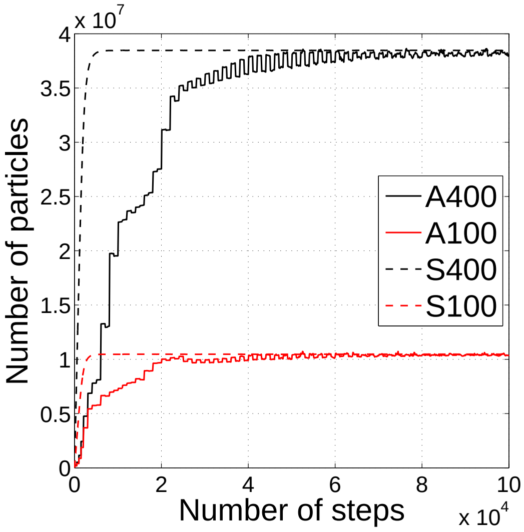


Figure 5.1: Evolution of the number of particles in the simulations from an empty domain at step 0 to steady state with $N_{\text{adapt}} = 1,000$ for the test cases summarized in Table 5.2.

instance required in the case of chemically reacting flows such as that presented in [60]. All species are assumed to be in equilibrium with no bulk velocity in the y or z directions when emitted. The Knudsen number of the two jets, based on their radius and VHS equilibrium results is 0.015 and all species are emitted at Mach number of 1. All species are assumed to consist of argon and no chemical reactions take place so that the tag, i.e. the species number of a particle, remains constant. Supersonic outflow conditions are used for the left, top and right boundaries while a *de facto* axisymmetric boundary condition is enforced by the axisymmetric move procedure at the centerline (see [15] p. 370). Specular reflection wall boundary conditions are imposed at the wall between jet 2 and the outflow of the domain. Collisions between particles are modeled via the variable hard sphere (VHS) model [15] with a reference diameter of 4.17×10^{-10} m at 273 K and a temperature exponent of 0.81. The sophisticated DSMC pair selection scheme of Eq. (5.37) is used while further details concerning our implementation of the DSMC method can be found in [50]. The mesh consists of 60,200 triangular cells with a maximum value for $\frac{\lambda}{\langle \Delta x \rangle}$ of 0.40 close to the inlet of the two jets.

	Number density m^{-3}	x-Bulk velocity m/s	Temperature [K]
Jet 1			
1	1.2574×10^{23}	589	1,000
3	1.2701×10^{21}	589	1,000
Jet 2			
2	1.2701×10^{23}	-589	1,000

Table 5.1: Jet boundary conditions.

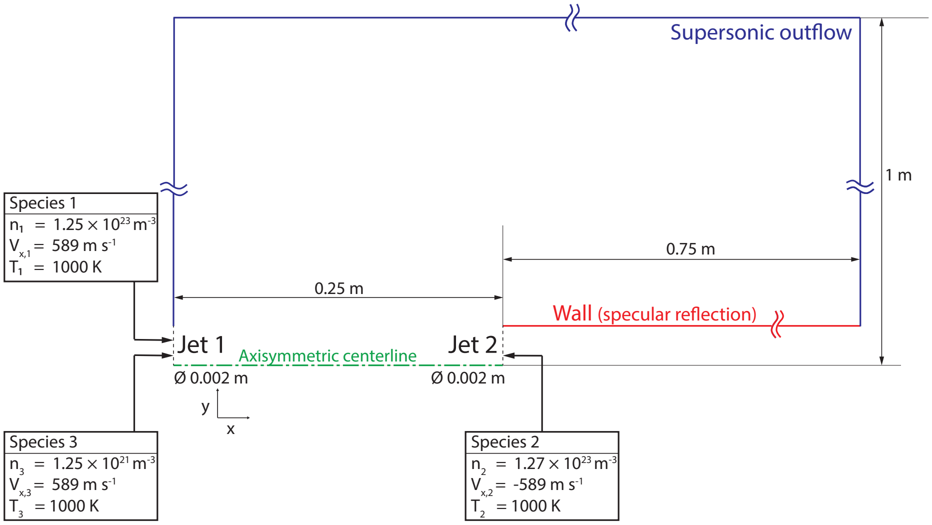


Figure 5.2: Schematic of test case with boundary conditions and dimensions.

5.4.2 Results

The results presented in the following are obtained by using the numerical parameters indicated in Tables 5.2 and 5.3. The value of the bounds used in the application of the adaptive procedure are reported between brackets along with the values attained by the corresponding variable in the actual simulation (between parenthesis). For instance in the case of simulation A400, the minimum allowable value for the relative weight of all species is set at 10^{-8}, while the minimum relative weight value obtained by the adaptive procedure for species 1 is 30×10^{8}, i.e. greater than the minimum bound. Test case A400 represents an attempt with a relatively moderate particle count to resolve the flow at hand with the help of the adaptive method for which detailed results will be presented in the following. More accurate results can be obtained by increasing the main governing parameter of the method, i.e. $N_{p,\text{want}}$, which will later be accomplished with test case A4800 to obtain a reference "exact" solution. Conversely, the computational cost of the simulation can be lessened, at the cost of reduced accuracy, by decreasing $N_{p,\text{want}}$ as done with test case A100. The adaptive procedure is applied from time 0 (where the computational domains contains no particles) up until time step $100,000$ at which point the sampling of flow quantities begins. The evolution of the total number of particles in the domain for the period of time during which the adaptive procedure is applied is shown in Fig. 5.1. The normalized weight $\widehat{W}_p = \widetilde{W}_p \, \tilde{\Delta} t$ and time step $\tilde{\Delta} t$ obtained with the method for test case A400 are shown in Fig. 5.4. A relatively low time step value is obtained close to the inlets of the two jets where the large number density translates into a low mean collision time which is shown in the bottom part of Fig. 5.5. This in turn constrains the value of the time step following Eq. (5.38). To combat an excessive increase in the number of particles in the radial direction due to the radial growth of cell volumes, visible in Fig. 5.5, the radially increasing weight field shown in Fig. 5.4 is obtained. It is somewhat similar to the banded patterns, i.e. $\widetilde{W}_p \sim y$, commonly used to simulate axisymmetric flow. Figure 5.6 displays the relative weight fields obtained for species 1 and 2. As expected, the relative weight of a given species gradually decreases away from the jet from which it is emitted, ultimately reaching its minimum value in front of the opposing jet. The time step and weight fields generated by the procedure are indeed seen to produce an almost constant $N_{p,\mathrm{want}}$ average number of particles throughout the domain as shown in Fig. 5.7 which shows that $\overline{N}_1 \sim \overline{N}_2 \sim 400$, as would be expected, in the A400 case.

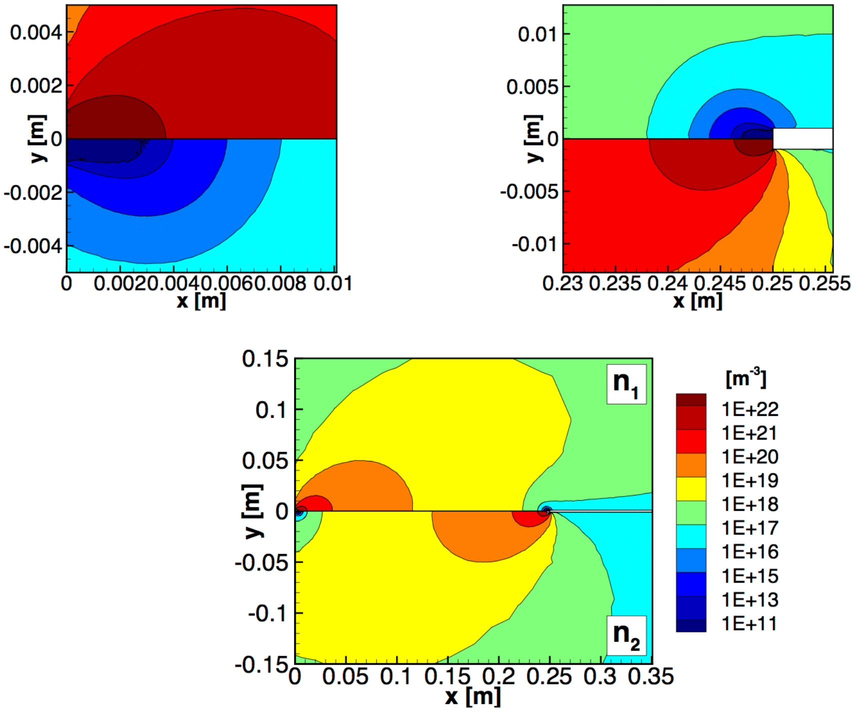


Figure 5.3: Sampled number density of species 1 (top) and 2 (bottom).

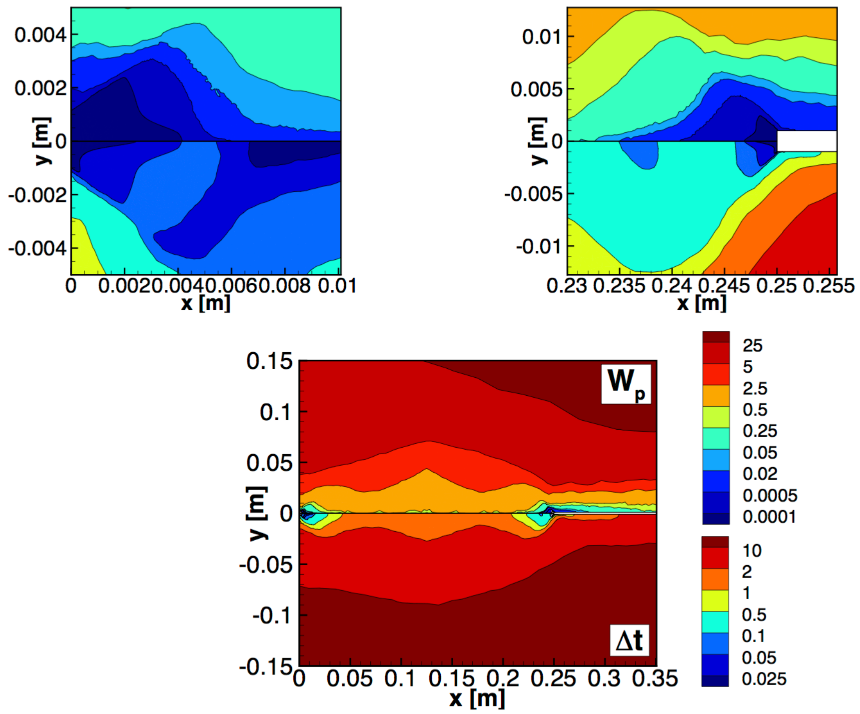


Figure 5.4: Normalized cell weight $\widehat{W}_p = \widetilde{W}_p \tilde{\Delta}t$ and time step $\tilde{\Delta}t$ (bottom) for A400.

5.4.3 Conditions of use and limitations of the scheme

The explicitly stated goal of the adaptive procedure, i.e. to obtain $N_{p,\text{want}}$ particles per species in all cells of the domain, c.f. Eqs. (5.47) and (5.52), although certainly possible from a purely numerical standpoint, reveals itself to be somewhat shortsighted as far as accuracy is considered once the scheme is actually implemented. This is in large part due to the occurrence of the *avalanching* phenomenon that was described in detail in Section 5.2.2.2, i.e. the fact that one cannot decrease weights *ad infinitum* in a certain direction to offset the natural decrease in particle count brought about by a negative number density gradient if the flow is not collisional enough. The avalanching phenomenon is only a very obvious manifestation of a broader phenomenon, i.e. the fact that cloning particles at a weight interface does not introduce new information in the simulation. This explains why the expected

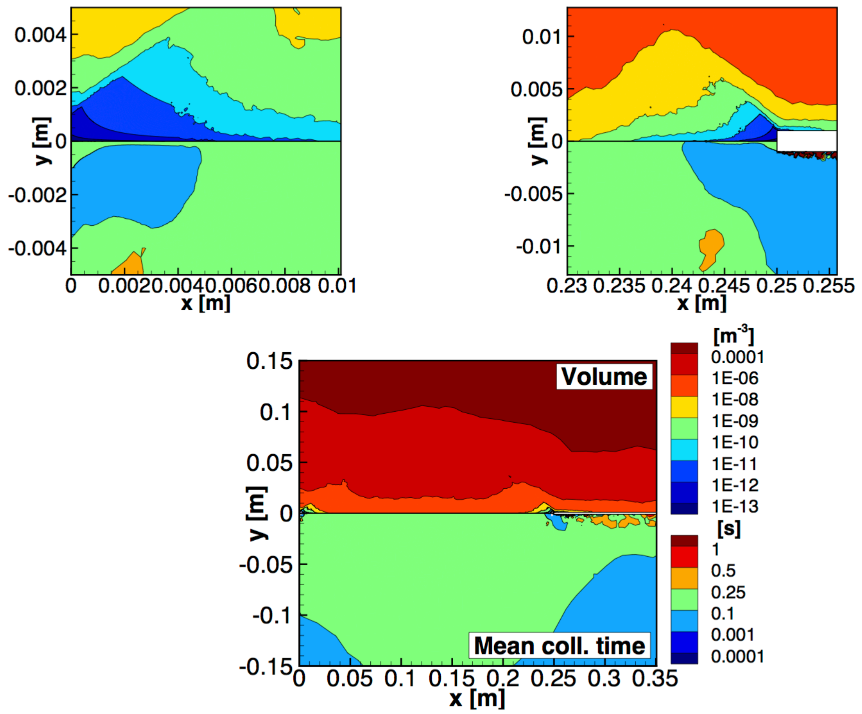


Figure 5.5: Cell volume V of the mesh employed (top) and mean collision time τ_m.

level of accuracy that one should expect of a given cell in a simulation with spatially varying weights that contains $\overline{N}$ particles on average is inferior to what is obtained when such a cell contains the same average number of particles in a simulation where uniform weights are employed. The decrease in the quality of the information content of a particle in a simulation with spatially varying weights is, however, extremely difficult to quantify *a priori*, that is before actually conducting a simulation. During the course of the development and testing of the adaptive procedure, two important best practices have been identified. 1) The average number of particles per cell that one is aiming for, i.e. $N_{p,\text{want}}$, should be relatively high bearing in mind that the commonly cited guideline of “20 particles per cell for accuracy” does not apply when cloned particles are present. This is the reason why the results that were previously shown, i.e. A400 and A100 are obtained with respectively 400 and 100 particles, far

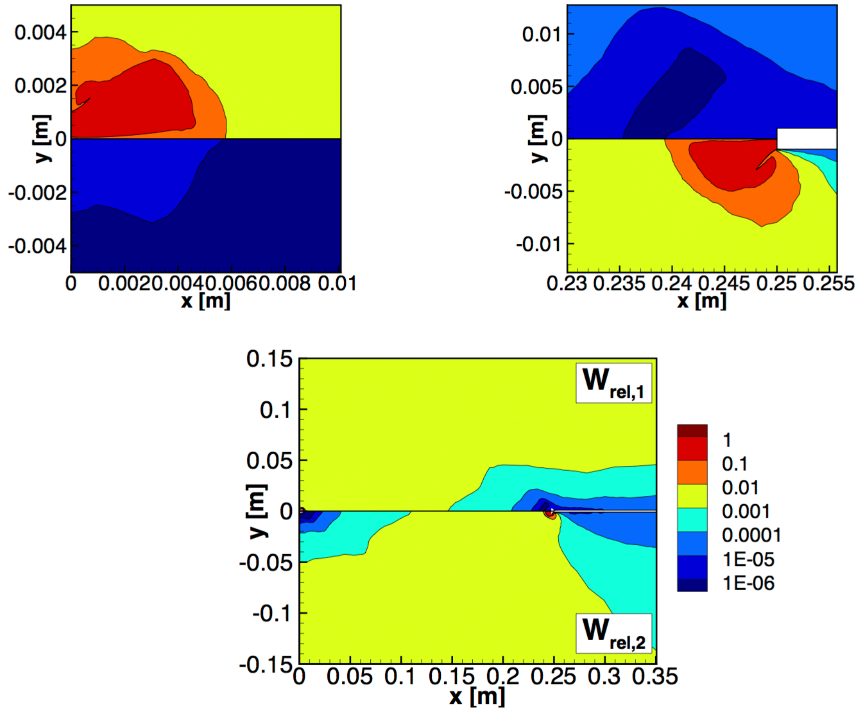


Figure 5.6: Relative weight for species 1, $W_{\text{rel},1}$ (top) and 2, $W_{\text{rel},2}$ (bottom) for A400.

in excess of what one traditionally aims for in a DSMC simulation. 2) One should not use minimum weight bounds that are too low in the hope of *really* having $N_{p,\text{want}}$ particles in absolutely all cells of the domain. The value of the bounds should be commensurate with that of $N_{p,\text{want}}$ so that the avalanching phenomenon does not become too manifest. When the occurrence of such a phenomenon becomes apparent, it is the sign that sampled quantities will be negatively impacted by the phenomenon in zones where they occur and that at least equally good results are obtainable without the use of such low weights for the same $N_{p,\text{want}}$ value. This is the reason behind the use of relatively high minimum weights bounds, i.e. 10^{-8} in Table 5.2, for test cases A100 and A400. Looking at the sampled value of the number density in Fig. 5.3, there is an approximate 12 order of magnitude drop (from $\sim 10^{23}$ to $\sim 10^{11}$ [m^{-3}]) between the value of the number density at the outlet of jet and that of jet 2, al-

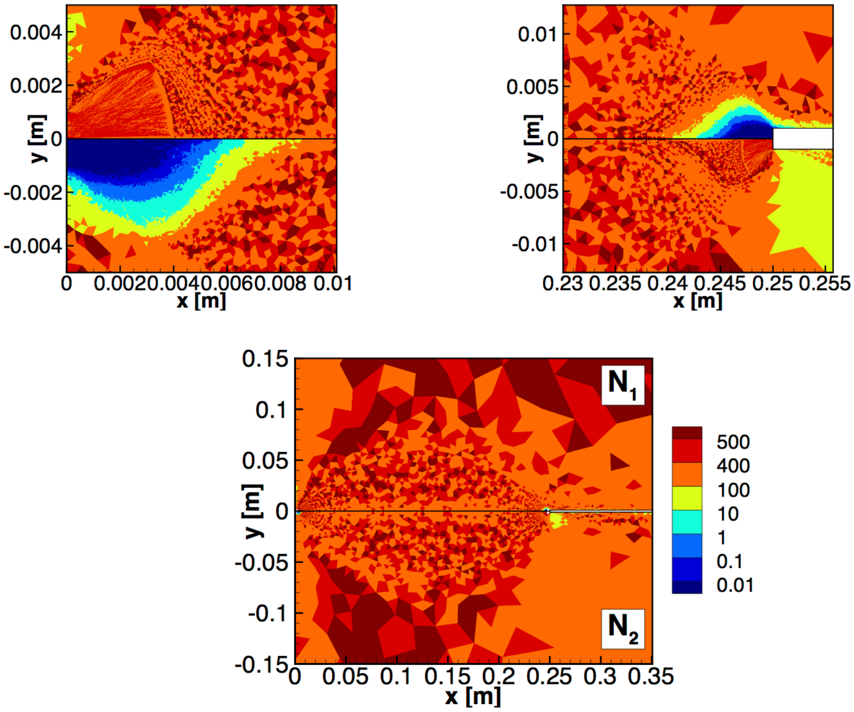


Figure 5.7: Average number of particles for species 1, $\overline{N}_1$ (top) and 2, $\overline{N}_2$ (bottom) for A400.

though the latter cannot be properly resolved, even with the adaptive procedure. As the mesh geometry is identical close to the two inlets so that cells have the same volume in front of the two jets (Fig 5.5), this would in turn entail that a relative weight of at most $\frac{1}{10^{12}} = 10^{-12}$ is needed to properly populate that region with the species emitted by the opposite jet. This is far below the specified weight of 10^{-8}, which explains the very low average number of particles of species 1 observed close to jet 2 in Fig. 5.7. One can, for instance, try to populate the region next to jet 2 with more particles of species 1 by lowering the minimum allowable $W_{\mathrm{rel},1}$ from the 10^{-8} value that is used for test case A100, to 10^{-10}, as done in test case A100-1em10 (outlined in Table 5.3). The results of the latter test case are presented in Figs. 5.8 and 5.9 where the hallmarks of the avalanching phenomenon described in Section 5.2.2.2 are clearly visible. One sees in Fig. 5.8 groups of large numbers ($\sim 10^5$) of particles of species 1 approaching jet 2 with $V_{x,1} > 0$, as shown in Fig. 5.9. Particles of species 1 are seen emanating from that packet of particles with $V_{x,1} < 0$ velocities due to collisions with particles of species 2 which become increasingly prevalent as it approaches the outlet of jet 2. Those particles form the tail of the packet of particle that is clearly visible in light blue in Fig. 5.8 and dark blue in Fig. 5.9. The infrequent occurrence of groups containing large numbers of particles of species 1 is the main mechanism populating the region close to the inlet of jet 2 with particles of species 1. The collisions of these particles with those of species 2 is what gives rise to the negative value of the sampled $V_{x,1}$ in that region which is shown in Fig. 5.9. In fluid dynamics terms, this corresponds to the occurrence of a recirculation region which is clearly visible if streamlines are plotted (not shown here). The deleterious effects of the avalanching phenomenon on sampled quantities is clearly shown in Fig. 5.9 where streaks are visible in the sampled $V_{x,1}$ contours. One further computational consequence of the avalanching phenomenon, visible in Fig. 5.8, is that some cells are seen to contain very large numbers of particles, i.e. in excess of 200,000 in the present case. Because they only seldom do, this only translates into a moderate increase in the average number of particles, calculated with Eq. (5.9), as seen in the lower part of Fig. 5.8 which in turn does not lead to the adaptive procedure increasing the value of the weights in that region. Most DSMC codes, such as these authors, can however not contain an arbitrarily large number of particles without ultimately running out of memory, so that the minimum bound value employed has to be limited. If for instance, as previously described, one uses $W_{\mathrm{rel},1} = 10^{-12}$ in the zone in front of jet 2, whenever a particle of species 1 ultimately makes its way unhindered (i.e. without collisions) from jet 1 to jet 2, and is cloned from $\widetilde{W}_p W_{\mathrm{rel},1} \sim 1$ to 10^{-12} , one ultimately obtains at some time step $\sim 10^{12}$ particles of species 1 in some cell in front of jet 2. This potential memory issue is one further reason that cautions against the use of excessively low weights.

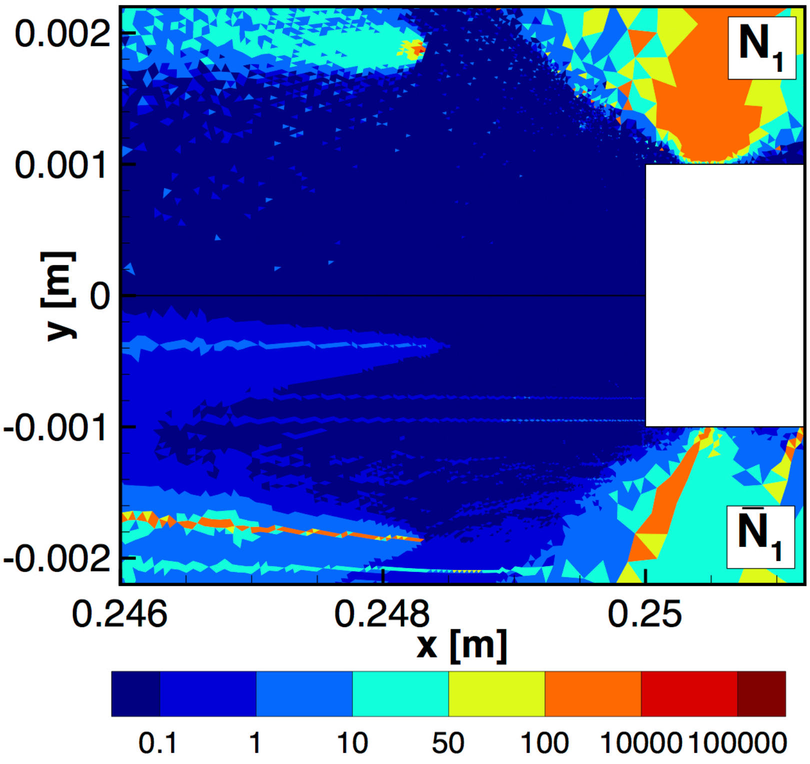


Figure 5.8: Number of particles per cell, N_1, (top) and average number of particles per cell, $\overline{N}_1$, (bottom) for species 1 at a given time step obtained with test case A100-1em10.

5.4.4 Accuracy and Efficiency

As previously stated, most DSMC simulations of axisymmetric flows are at least conducted with radially varying weights [15] [96] [115] with values that increase with the distance to the centerline. This is performed to increase particle counts close to the centerline while preventing an excessive number of particles in cells further removed from it as their volume scales with y. Whenever such capability exists [99] [19], a spatially varying time step is also used in DSMC simulations where large number density (and thus mean collision time) gradients exist. This prevents the value of the overall time step of the simulation from being dictated by the usually small low Knudsen number zone thus accelerating the march to steady state of the simulation while increasing the convergence of sampled quantities in zones where the time step

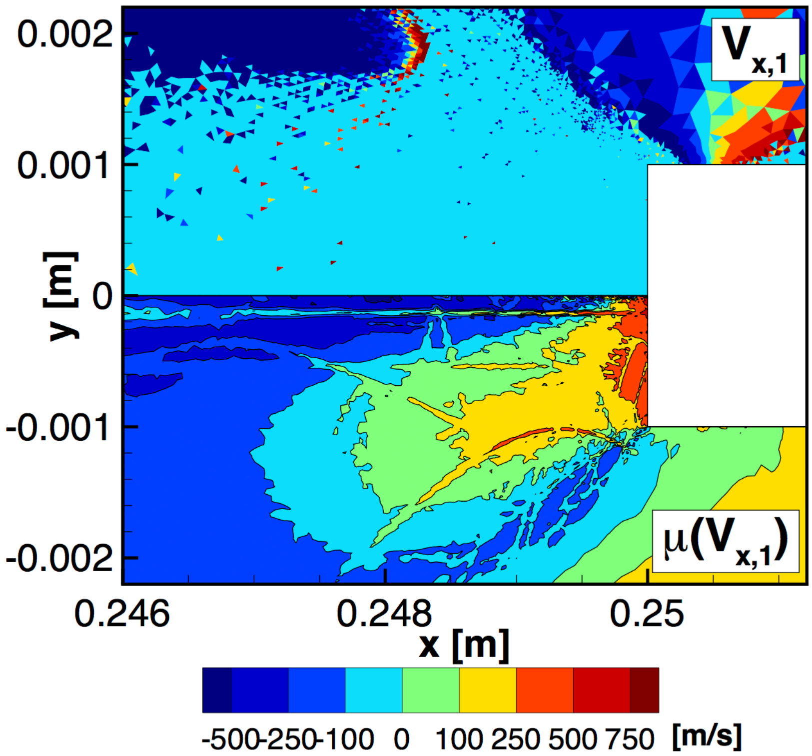


Figure 5.9: Average velocity of particles of species 1 in each individual cell $\overline{V}_{x,1}$, with $\overline{V}_{x,1} = 0$ in the absence of particles (top) and sampled $V_{x,1}$ velocity (bottom) in m/s obtained with test case A100-1em10.

is increased. The value of the weight and the time step is often determined *a priori* which often requires that the simulation be run at least twice to determine optimum time step and weight fields. The adaptive procedure presented in this work removes this need thus allowing simulations to be conducted with less human intervention. This is in particular useful for parametric studies where a large number of simulations are run on the same mesh with different boundary conditions or to allow non-expert DSMC users to run efficient simulations. In order to obtain a fair assessment of the efficiency gain of the adaptive method within the framework of this test case, the results of the adaptive method should be compared to those obtained with spatially varying time step and weights, as would almost certainly be used with most DSMC codes to simulate the present test case. The most natural time step and weight fields to use in the comparison are those that are themselves obtained by the adaptive method when applied to the test case. However, using these without relative weights (such as those produced by the adaptive method) results in a drop in the total number particles in the simulation as relative weights naturally increase particle count in the cells for which they are less than 1. To compensate for this drop in the number of particles, $W_{p,0}$ is reduced in the simulation so as to match the particle count produced by the adaptive method. The simulations conducted without relative weight, using the time step and weight field produced by the adaptive technique and with a suitably chosen $W_{p,0}$, are designated as *standard* simulations in the following. The reference simulations corresponding to simulation A400 and A100 respectively are designated by S400 and S100 in Table 5.2 where their numerical parameters are summarized. A comparison of the numerical costs between the standard and adaptive simulations is shown in Table 5.4. The reduced computational cost of the adaptive method is due to the use of a smaller number of particles (visible in Fig. 5.1) in the run up to steady state (which is assumed to occur at $k = 100,000$) as well as a slightly smaller number of collisions.

A comparison of the value of the sampled number density and velocity for species 1 obtained with the standard (S100, S400) and adaptive methods (A100, A400) is presented in Figs. 5.10 and 5.11. The adaptive method is seen to produce a much smoother and more realistic sampled number density field than the standard method for both particle counts while an even greater improvement is visible for the sampled velocity. The adaptive method is thus seen to provide a large improvement over the standard method for the same particle count. This large qualitative improvement can be quantified by studying the numerical error of the various solutions plotted in Figs. 5.10 and 5.11.

We designate any cell-based quantity, such as number density or velocity of cell i, by

x_i. The sampled value of x_i is referred to by $\mu_i(x)$, while numerical error $\overline{\varepsilon}_i[\mu(x)]$ for a given simulation is defined by:

$$\overline{\varepsilon}_i[\mu(x)] \triangleq \mu_i(x) - \mu_i^E(x), \tag{5.54}$$

where $\mu_i^E(x)$ designates the exact solution, i.e. that obtainable in the limit of an infinite number of particles and an infinitesimally small time step. This definition of numerical error can in turn be used to define a normalized numerical error:

$$\overline{\varepsilon}_{0,i}[\mu(x)] = \frac{|\overline{\varepsilon}_i[\mu(x)]|}{|\mu_i^E(x)|}. \tag{5.55}$$

The exact solution for such a complex case is of course inaccessible in practice and must be approximated by that obtained with a much larger number of particles than the simulation for which the error is calculated. In the present study, the exact solution is thus assumed to be furnished by test case A4800 which is detailed in Table 5.3. Although one would ideally not want to use a solution obtained with the adaptive method to gauge its accuracy, we are forced to do so as obtaining a solution of comparable quality without relative weights is prohibitively expensive. This, however, seems justified given the seemingly good results obtained with the adaptive method at lower $N_{p,\mathrm{want}}$ values which are shown in Fig. 5.10. The normalized numerical error along the centerline for $\mu(n)$ and $\mu(V_x)$ obtained with both the adaptive and standard methods are plotted for species 1 in Fig. 5.12 and species 2 in Fig. 5.13. The log scale and the scaling of the x axis are chosen for each particular species so as to highlight the behavior of the error close to the jet opposite that from which it is emitted, where the error is highest. The adaptive method is observed to systematically produce a solution with a smaller numerical error than the standard method. An order of magnitude reduction in the value of the error for sampled number density and velocity is observed for both species for most of the centerline. The adaptive procedure for the two $N_{p,\text{want}}$ examined is, however, still not able to obtain accurate sampled quantities in the remaining $\Delta L \sim 0.005$ m. Proper resolution of the zone requires a further increase in $N_{p,\text{want}}$ such as that used to calculate the “exact” solution. The default value of the sampled number density and velocity in a cell is 0, and both remain unchanged until at least one particle enters it. This explains the value of 1 for the normalized error that is observed at some points close to the opposing jet with the standard method which sometimes proves to be smaller than that obtained with the adaptive method. Similar improvements (not shown here) are obtained with the adaptive procedure for species 3 along the centerline as well as for sampled collision rates between species.

The large improvement observed for sampled quantities with the adaptive method for the same particle count can be explained by the improved spatial distribution of particles amongst the cells of the domain. Figure 5.14 illustrates this point, showing that in the case S400, a few cells close to the inlet contain a large average number of particles of species 1 while that number rapidly decreases in the x direction. In contrast to that, the average number of particles for test case A400 is much more uniform with no cell containing large, e.g. ~ 1000, average numbers of particles, and only drops in close proximity to jet 2. The situation can be summarized by the histogram presented in Fig. 5.15 which shows the distribution of all the cells of the domain according to their $\overline{N}_1$ value for both A400 and S400. In test case S400, the mean value of $\overline{N}_1$ is greater than that observed for S400, reflecting the fact that the simulation actually contains more particles of species 1 than A400 (18.6 vs 12.9 M as indicated in Table 5.2). Whereas the distribution of $\overline{N}_1$ for A400, rapidly falls off for large $\overline{N}_1$ values, that for S400 displays a non-negligible tail (i.e. $\sim 14,000$ cells having $\overline{N}_1 \in (800, 1800]$). That tail corresponds to the cells located close to the inlet of jet 1 in Fig. 5.14, shown in dark red. The presence of a large number of particles in those cells amounts to a waste of computational effort as these particles do little to increase the accuracy of the simulation further downstream, in particular close to the second jet, since they quickly move away from the centerline. In the case of species 1, the adaptive procedure thus both prevents cells close to jet 1 from containing an excessive number of particles while ensuring that the number of particles remains constant in the downstream direction which, in the present case, can be accomplished with fewer particles than with test case S400. The exact same phenomenon occurs for species 2 while the adaptive procedure, in the case of A400, because of the choice of $N_{p,\text{want},3} = 400$, creates many more particles of species 3 (12.3 vs 0.19 M as indicated in Table 5.2) than are present in the S400 simulation. This in turn suggests that the adaptive method will of greatest benefit for DSMC simulations with a U-shaped distribution of the number of particles between cells such that of S400 in Fig. 5.15.

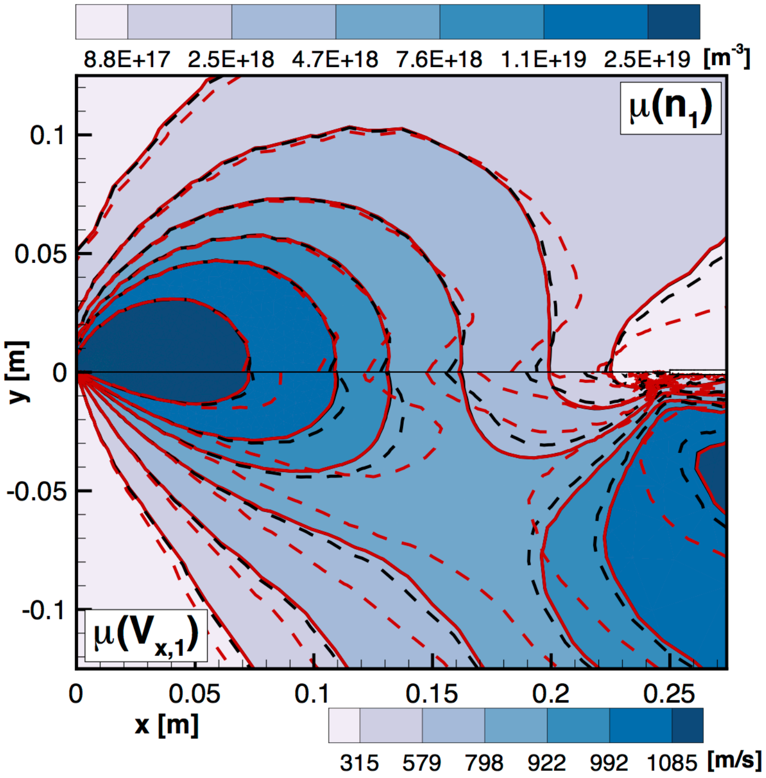


Figure 5.10: Sampled number density (top) and $V_{x,1}$ velocity (bottom) obtained with the adaptive (solid) and standard method (dashed) with A or S400 (black) and A or S100 (red).

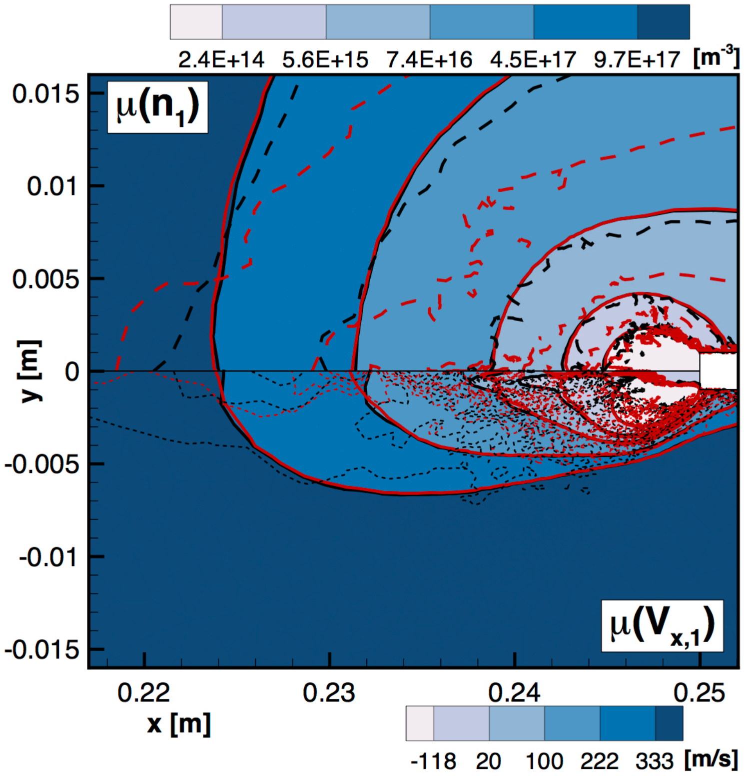


Figure 5.11: Sampled number density (top) and $V_{x,1}$ velocity (bottom) obtained with the adaptive (solid) and standard method (dashed) with A or S400 (black) and A or S100 (red).

5.5 Conclusions

An adaptive method for the DSMC time step, cell weight and species relative weights was introduced in this chapter. The aim of the procedure presented is to automatically update the time step, weight and relative weights of all cells as the simulation progresses to steady state. The value of the time step is maximized while still requiring it to satisfy DSMC requirements for all cells. The value of the weights and relative weights are chosen so as to obtain a specified average number of particles for all species in all cells. The formulation of the adaptive method makes its implementation in existing DSMC codes straightforward while its computational overhead, compared to that of the simulation is negligible. It furthermore reduces the need for user input by determining the optimum time step and weight fields automatically

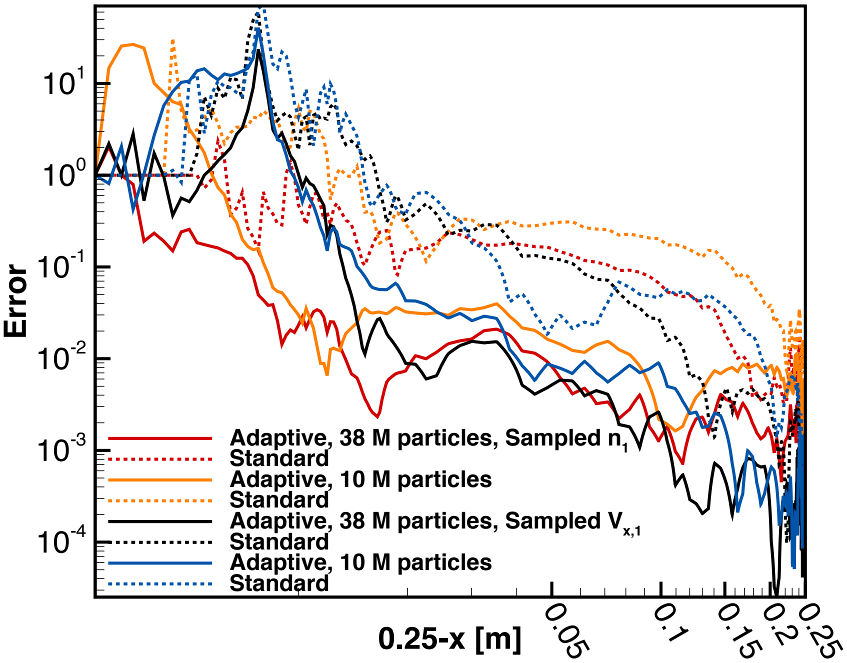


Figure 5.12: Numerical error for sampled n_1 and $V_{x,1}$ on the centerline computed with A4800 as an exact solution.

thereby eliminating the need to run multiple simulations so as to determine them. The potential computational savings allowed by method were illustrated by simulating the flow between two counterflowing jets at a Knudsen number of 0.015 with the new method on one hand and with a spatially varying time step and cell weight field on the other. For the same total number of particles, the use of the adaptive method allows a substantial reduction, i.e. an order of magnitude between the two jets, in the value of the error for the sampled number density and velocity. The increased computational efficiency is achieved by more uniformly distributing particles throughout the domain thereby preventing cells from containing an excessive or an insufficient number of particles. The adaptive method is thus expected to be of most benefit for multi-species flows with uneven distributions of the number of particles between cells and species. A deleterious consequence of the effect of particle cloning in the relative

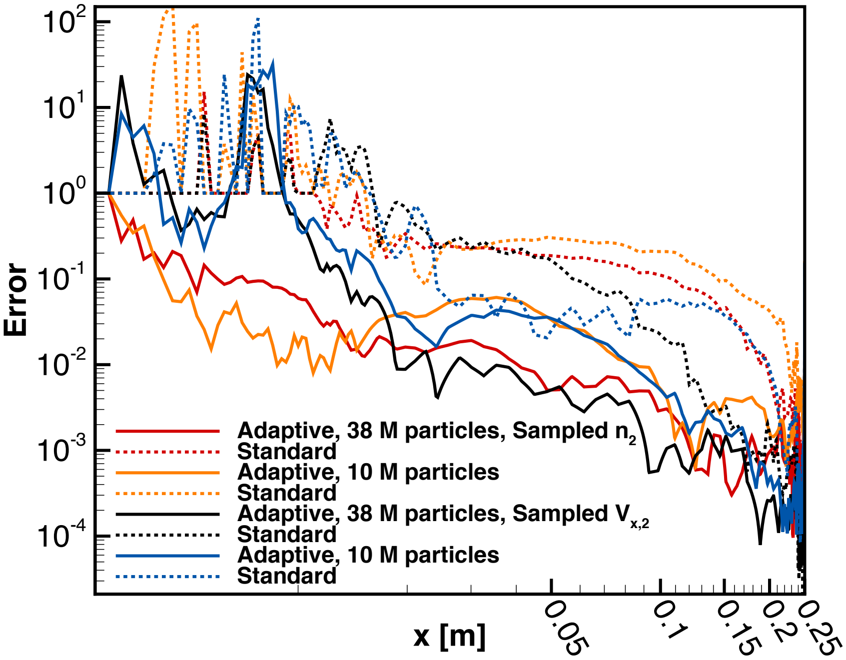


Figure 5.13: Numerical error for sampled n_2 and $V_{x,2}$ on the centerline computed with A4800 as an exact solution.

absence of collisions, the *avalanching* phenomenon, was examined and found to be mitigated by reducing the range of weights in the simulation while increasing the required average number of particles per cell. This work has furthermore highlighted the potential harmful effects introduced by the cloning of particles at weight interfaces. Alleviating those could clearly improve the accuracy of the present method and allow the use of even fewer particles.

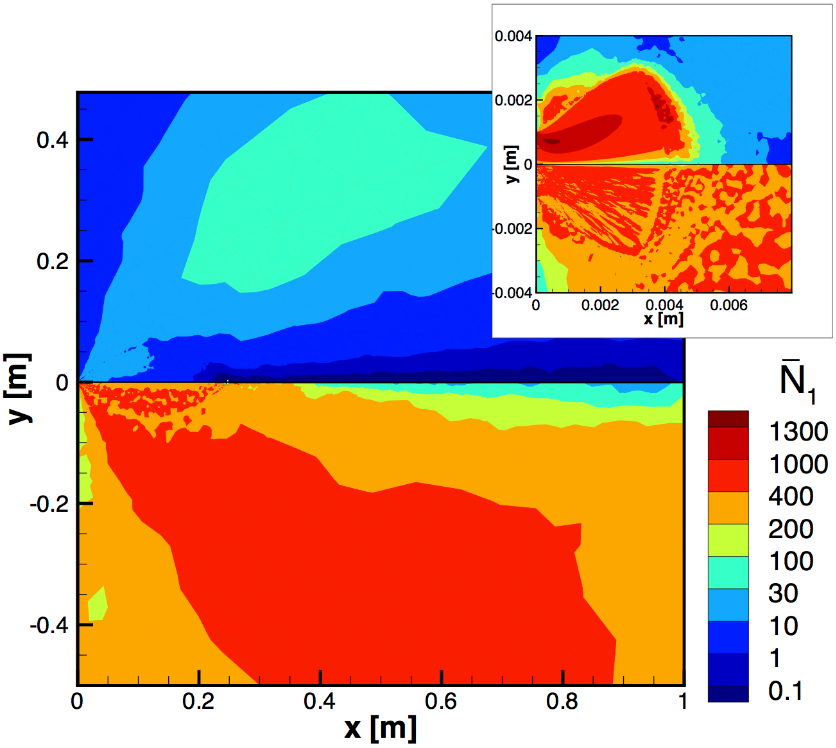


Figure 5.14: Comparison of the average number of particles for species 1, $\overline{N}_1$, obtained for test case S400 (top) and for test case A400 (bottom).

Designation	A400	A100	S400	S100
$W_{p,0}$	5.6×10^{12}	5.6×10^{12}	0.71×10^{12}	2.42×10^{12}
Δt_0 [s]	5.0×10^{-8}	5.0×10^{-8}	5.0×10^{-8}	5.0×10^{-8}
Adaptive $\widetilde{\Delta t}$	YES	YES	NO,$\widetilde{\Delta t}$ from A400	NO,$\widetilde{\Delta t}$ from A100
Adaptive $\widetilde{W_p}$	YES	YES	NO,$\widetilde{W_p}$ from A400	NO,$\widetilde{W_p}$ from A400
Adaptive W_{rel}	YES	YES	NO	NO
$N_{p,\mathrm{want},1,2,3}$	400	100	N/A	N/A
Total realized number of particles (M)	38.6	10.5	38.6	10.5
Realized number of particles for species (1,2,3) (M)	(12.9,13.2,12.3)	(3.7,3.7,3.1)	(18.6,19.7,0.19)	(5.1,5.3,0.05)
$\left[\widetilde{\Delta}_{t,\min}, \widetilde{\Delta}_{t,\max}\right]$	$\left[10^{-12}, 10^{12}\right] \left(3.3\ 10^{-3}, 109.7\right)$	$\left[10^{-12}, 10^{12}\right] \left(6.6\ 10^{-3}, 116.7\right)$	$\left(3.3\ 10^{-3}, 109.7\right)$	$\left(6.6\ 10^{-3}, 116.7\right)$
$\left[\widetilde{W}_{p,\min}, \widetilde{W}_{p,\max}\right]$	$\left[2^{-10}, 2^{50}\right] \left(1.9\ 10^{-3}, 1\right)$	$\left[2^{-10}, 2^{50}\right] \left(1.9\ 10^{-3}, 1\right)$	$\left(1.9\ 10^{-3}, 1\right)$	$\left(1.9\ 10^{-3}, 1\right)$
$W_{\mathrm{rel},\min,1,2,3}$	$\left[10^{-8}\right], (30, 1.05, 1)\, 10^{-8}$	$\left[10^{-8}\right], (34, 1.2, 1)\, 10^{-8}$	1	1
$W_{\mathrm{rel},\max,1,2,3}$	$(0.42, 0.87, 0.004)$	$(1, 1, 0.017)$	1	1
$\left\{\widetilde{W}_p\ W_{\mathrm{rel}}\right\}_{\min,1,2,3}$	$\left[10^{-8}\right], (2.0, 2.0, 1.8)\, 10^{-9}$	$\left[10^{-8}\right], (2.0, 2.0, 1.8)\, 10^{-9}$	$(2.0, 2.0, 1.8)\, 10^{-9}$	$(2.0, 2.0, 1.8)\, 10^{-9}$

Table 5.2: Parameters of different test cases presented. The set values of the bounds are shown between brackets while actually observed values are reported between parenthesis.

Designation	A4800	A100-1em10
$W_{p,0}$	5.6×10^{12}	5.6×10^{12}
Δt_0 [s]	5.0×10^{-8}	5.0×10^{-8}
Adaptive $\widetilde{\Delta t}$	YES	YES
Adaptive $\widetilde{W_p}$	YES	YES
Adaptive W_{rel}	YES	YES
$N_{p,\text{want},1,2,3}$	4800	100
Realized number of particles (M)	487	× (Not run to steady state)
Realized number of particles for species (1,2,3) (M)	(161,174,153)	×
$\left[\widetilde{\Delta}_{t,\min}, \widetilde{\Delta}_{t,\max}\right]$	$\left[10^{-12}, 10^{12}\right] \left(8.8\ 10^{-4}, 105.4\right)$	$\left[10^{-12}, 10^{12}\right] \left(11.1\ 10^{-4}, 126.5\right)$
$\left[\widetilde{W}_{p,\min}, \widetilde{W}_{p,\max}\right]$	$\left[2^{-10}, 2^{50}\right] \left(1.9\ 10^{-3}, 1\right)$	$\left[2^{-10}, 2^{50}\right] \left(1.9\ 10^{-3}, 1\right)$
$W_{\text{rel},\min,1,2,3}$	$\left[10^{-10}\right], (23.3, 1.01, 1.09)\, 10^{-10}$	$\left[10^{-10}\right], (1.19, 1.19, 1.19)\, 10^{-7}$
$W_{\text{rel},\max,1,2,3}$	$[1], (0.089, 0.087, 0.0009)$	$[1], (1, 1, 1)$
$\left\{\widetilde{W}_p\ W_{\text{rel}}\right\}_{\min,1,2,3}$	$\left[10^{-11}\right], (2.61, 2.39, 1.66)\ \ 10^{-11}$	$\left[10^{-10}\right], (2.33, 2.33, 2.33)\ \ 10^{-10}$

Table 5.3: Parameters of different test cases presented (Continued). The set values of the bounds are shown between brackets while actually observed values are reported between parenthesis.

Designation	S100	A100	S400	A400
Total number of particles (M)	10.5	10.5	38.6	38.6
Average number of collisions per time step (K)	114.5	107.6	396.8	348.4
Relative CPU time†	100	0.90	3.28	2.96

Table 5.4: Computational cost comparison between simulations.

† required to reach time step 400,000 (depicted in Fig. 5.1)

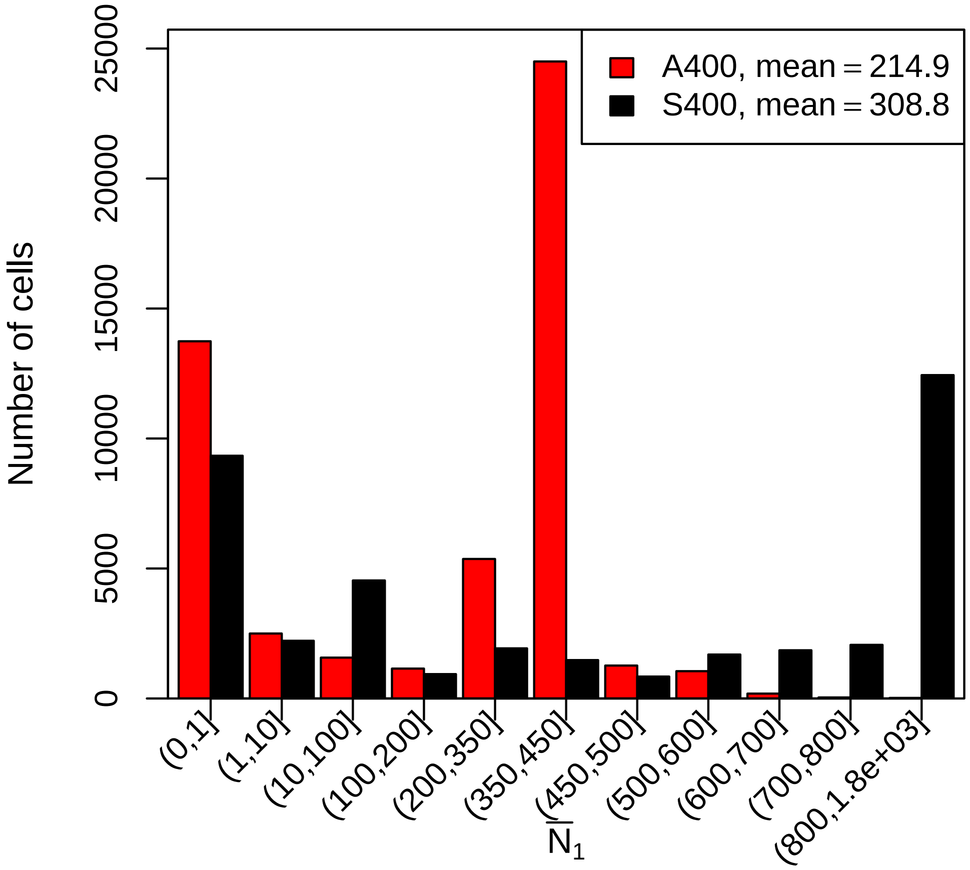


Figure 5.15: Comparison of the distribution of particles of species 1 amongst the cell of the domain for test cases A400 and S400.

CHAPTER VI

Simulation of rarefied ionized flows via DSMC

6.1 Introduction

6.1.1 Motivation

Whereas the work presented in the thesis up until this point only dealt with rarefied gas flows that only contained neutral particles, the focus of the remainder will be the simulation of weakly ionized rarefied gas flows. In this flow regime, three notable types of particles are present. The first two are heavy neutral and charged particles, which, as the flow is in the rarefied regime, are respectively governed by the Boltzmann equation without and with an electric force term. The third type of particles is the electrons for which motion is also governed by the Boltzmann equation with an electric field. Their dynamics are, however, very different than those of heavy particles due to their very light mass, an electron being 1836 times lighter than a proton [80]. In order to assess more quantitatively the various length and time scales at play, we consider as an example the flow conditions at the outlet of the plasma source that will be later simulated in Section 6.5 which are briefly summarized in Table 6.1.

The various time and length scales of the plasma at these conditions are reported in Tables 6.2 and 6.3 along with the respective methods/equations used for their cal-

Species	density $[\mathrm{m}^{-3}]$	Temperature	Mean x thermal velocity [m/s]	Bulk x velocity [m/s]
Ar	2.5×10^{21}	300 K	399[1]	322
Ar^+	2.5×10^{19}	300 K	399[1]	620
e^-	2.5×10^{19}	2 eV	5.9×10^5[2]	620

[1] $\overline{|\vec{v}'_x|} = \sqrt{\frac{k_B T}{m_{\mathrm{Ar}}}}$ [147]

[2] $\overline{|\vec{v}'_x|} = 4.19 \times 10^5 \left(T_e\,[\mathrm{eV}]\right)^{1/2}$ [m/s] [80]

Table 6.1: Plasma source flow conditions.

culations. Following the convention previously adopted in Chapter V, τ_{ij} designates the mean collision time of a particle of species i with all of those of species j, while λ_{ij} designates the average distance between two such events (i.e. the mean free path). In the following, the consequences that these plasma conditions entail in terms of computational method and cost will be examined.

Species	Ar	e
Ar	$\tau_{\mathrm{ArAr}} = 1.3 \times 10^{-6}$[1]	$\tau_{\mathrm{Ar}e} = 5.7 \times 10^{-7}$ [2]
e	$\tau_{e\mathrm{Ar}} = 5.7 \times 10^{-9}$ [3]	$\tau_{ee} = 3.9 \times 10^{-9}$[4]

[1] Obtained with the VHS model (see Chapter II).

[2] $k_{e-Ar} = \exp\left(-31.3879 + 1.609\ \log(T_e) + 0.0618\ \log(T_e)^2 - 0.1171\ \log(T_e)^3\right)$ [76] and $\tau_{Are} = \frac{1}{k_{e,Ar} n_e}$

[3] ibid with $\tau_{e,Ar} = \frac{1}{k_{e,Ar} n_{Ar}}$

[4] $\nu_{ee} = 2.91 \times 10^{-12}\ n_e \left[\mathrm{m}^{-3}\right] \log\Lambda \left(T_e\,[\mathrm{eV}]\right)^{-3/2}$, with $\log\Lambda = 10$ [80]

Table 6.2: Mean collision time between the species of the plasma source described in Table 6.1.

Species	Ar	e
Ar	$\lambda_{\mathrm{ArAr}} = 5.3 \times 10^{-4}$ [1]	$\lambda_{\mathrm{Are}} = 2.26 \times 10^{-4}$ [2]
e	$\lambda_{e\mathrm{Ar}} = 3.4 \times 10^{-3}$ [2]	$\lambda_{ee} = 2.3 \times 10^{-3}$ [2] $\lambda_D = 2.1 \times 10^{-6}$ [3]

[1] Obtained with the VHS model (see Chapter II).

[2] $\lambda_{ij} = \frac{v_{\mathrm{th},i}}{\nu_{ij}}$ with $i, j = e$ or Ar.

[3] $\lambda_D = 7.43 \times 10^3 \left(T_e\,[\mathrm{eV}]\right)^{1/2} \left(n_e\left[\mathrm{m}^{-3}\right]\right)^{-1/2}$ [80].

Table 6.3: Mean free paths of collisions between the species of the plasma source described in Table 6.1.

In order to obtain the most realistic results, one would ideally want to simulate the motion of electrons via a particle in cell method, a simulation method that will be discussed at greater length in Section 6.3.1. In this hypothetical ideal simulation, collisions between electrons, i.e. $e - e$ collisions (which collision rate is reported in Table 6.2), would be modeled via DSMC. The latter would also be used to model collisions between electrons and heavy species, i.e. $e - \text{Ar}$ and $e - \text{Ar}^+$ collisions. Due to the requirement of DSMC that the time step be less than the mean collision time (See Section 5.3.1) $\Delta t < \min_{ij} \tau_{ij} = \tau_{ee} = 3.9 \times 10^{-9}\,[s]$ as can be seen in Table 6.2. In a PIC simulation, the cell characteristic length scale $\langle \Delta x \rangle$ has to be less than the Debye length λ_D [78] while the simulation of collisions via DSMC requires that $\langle \Delta x \rangle$ be less than their mean free paths. Using the data reported in Table 6.3, the condition on the mesh size thus becomes $\langle \Delta x \rangle < \min_{ij} (\tau_{ij}, \lambda_D) = 2.1 \times 10^{-6}\,[\text{m}]$ due to the Debye length requirement. Furthermore, because of the CFL condition of both the PIC and DSMC method, already mentioned in Section 5.3.1, the time step Δt and characteristic length of each cell of the domain have to be chosen so as to satisfy

$$K_i = \frac{\langle v \rangle \Delta t}{\langle \Delta x \rangle} < 1, \tag{6.1}$$

which is most restrictive for the species with the greatest thermal or bulk velocity, i.e., here, the electrons with $v_{\text{th},e} = 5.9 \times 10^{-5}\,[\text{m/s}]$ as indicated in Table 6.1. With $\langle v \rangle \approx v_{\text{th},e}$[1], Eq. (6.1) respectively yields $v_{\text{th},e} \times \tau_{ee} = 2.3 \times 10^{-4} < \langle \Delta x \rangle$ or $\Delta t < 3.55 \times 10^{-12} = \frac{\lambda_D}{v_{\text{th},e}}$ when $\Delta t = \tau_{ee}$ or $\langle \Delta x \rangle = \lambda_D$ are used. The CFL condition thus furthers lower the permissible time step from τ_{ee} to $\frac{\lambda_D}{v_{\text{th},e}} = 3.55 \times 10^{-12}\,[\text{s}]$. The time step and cell size requirement of this ideal simulation are reported in Table 6.4 along with that of alternative simulation methods discussed in the following for comparison purposes. An alternative to the preceding approach, which was for instance adopted in [56], is to omit the simulation of electron-electron collisions. This can be justified by considering that, due to their high collision rates, i.e. $\tau_{ee} = 3.9 \times 10^{-9}\,[\text{s}]$,

---

[1] which, however, fails to account for highly energetic electrons in the tail of the velocity distribution function.

ee collisions only serve to thermalize the electron energy distribution function which is already near Maxwellian due to the high collision rate of electrons with heavy particles ($\tau_{e\mathrm{Ar}} = 5.7 \times 10^{-9}$ [s]). Within that simulation framework, only collisions with heavy particles, e.g. $e - \mathrm{Ar}$ and $e - \mathrm{Ar}^+$ here, are considered for the electrons. If the previous exercise is repeated without considering *ee* collisions, the exact same time step and cell size constraints as previously are obtained. The substantial computational cost reduction associated with that approach comes from not having to perform all the electron Coulomb collisions. As $\frac{\tau_{ee}}{\tau_{e\mathrm{Ar}}} \sim \frac{1}{100}$, the same reduction in the total number of collisions to perform can be expected. The time step and mesh size requirement, however, make such an approach prohibitively expensive to apply to two or three dimensional flows of practical interest as detailed in [56] which has led to a number of computational strategies to reduce the simulation costs. One can, for instance, increase the mass of the electron particles in the simulations [26] so as to reduce their thermal velocity which helps alleviates the constraint of Eq. (6.1) while also reducing the number of electron collisions that have to be considered. The use of the PIC approach and the associated $\langle \Delta x \rangle < \lambda_D$ requirement (cf. Table 6.4 with electron mass increased 1000 folds), however, still make such approach currently computationally intractable for the test case that will later be considered by requiring a very refined mesh and thus a great number of particles to have an adequate resolution. This need to reduce computational cost is the primary motivation behind the use of the present hybrid method. In a hybrid DSMC/fluid approach, electrons are not simulated as particles and their behavior is instead modeled by means of fluid equations. Because of the near Maxwellian character of the electrons that was previously discussed, the fluid equations provide a physically accurate description of the electrons in terms of only 3 variables (number density, momentum and kinetic energy). The equilibrium assumption also greatly facilitates the closure of the fluid equations system while removing the need for higher moments to be simulated. The cost of such a DSMC/PIC approach is that of the basic DSMC simulation with the added expense of solving the partial different equations of the fluid equation system. For the plasma condition of Table 6.1, the use of a hybrid approach allows a three order of magnitude reduction in the number of collisions that have to be performed as $\mathrm{e}-\mathrm{Ar}$ collisions are not performed (c.f. Table 6.2) and only $\mathrm{Ar}-\mathrm{Ar}$ and $\mathrm{Ar}^{+}-\mathrm{Ar}$ collisions are considered. The size of the mesh is furthermore not dictated by the Debye length but by the mean free path of heavy species, which allows the use of a much coarser (two orders of magnitude in each spatial direction) mesh along with a larger time step. A summary and comparison of the numerical parameters of the hybrid approach is presented in Table 6.4. The hybrid formulation that is used in this thesis was first introduced in [29] and subsequently used in [33], [44] and [81] to study the plasma plume produced by electric propulsion devices [25]. It represents an improvement over an earlier simplified fluid approach that relied on the Boltzmann relation to obtain the plasma potential that was initially introduced in [112] [113] and later used in [144] [67] [66]. Most weakly ionized rarefied flows are characterized by both large number density disparities between species in each individual cell of the domain as well as by large spatial variations throughout the domain. This is due to multiple factors, such as the occurrence of multiple chemical reactions that contribute to create trace species, or the low degree of ionization which de facto means that charged species have a low relative number density. Weakly ionized rarefied flows thus stand to gain from the use of the adaptive procedure presented in Chapter V. Its use for such flows would represent a good illustration of the application of the adaptive procedure within the framework of a complex DSMC flow. The main goal of this Chapter is thus to demonstrate the use of the adaptive procedure within the framework of a hybrid fluid electron/DSMC code. Anothe aim is to detail and formalize the coupling procedure between the fluid electron and DSMC models as it differs from the one presented in [33], [44] or [81].

Simulation method	Δt	$\langle \Delta x \rangle$	N_{coll}[1]
Ideal simulation (All DSMC)	$3.55 \times 10^{-12} \left(\frac{\lambda_D}{v_{\mathrm{th},e}}\right)$	$2.1 \times 10^{-6} \, (\lambda_D)$	$\propto 0.25 \times 10^{9} \left(\frac{1}{\tau_{ee}}\right)$
No *ee* collisions [57]	$3.55 \times 10^{-12} \left(\frac{\lambda_D}{v_{\mathrm{th},e}}\right)$	$2.1 \times 10^{-6} \, (\lambda_D)$	$\propto 0.17 \times 10^{9} \left(\frac{1}{\tau_{e\mathrm{Ar}}}\right)$
No *ee* collisions, $m_e \uparrow \times 1000$ [26]	$1.12 \times 10^{-10} \left(\frac{\lambda_D}{v_{\mathrm{th},e}}\right)$	$2.1 \times 10^{-6} \, (\lambda_D)$	$\propto 0.17 \times 10^{6} \left(\frac{1}{1000 \times \tau_{e\mathrm{Ar}}}\right)$
Hybrid DSMC/Fluid (Present)	$1.3 \times 10^{-6} \, (\tau_{\mathrm{ArAr}})$	$5.3 \times 10^{-4} \, (\lambda_{\mathrm{ArAr}})$	$\propto 0.72 \times 10^{6} \left(\frac{1}{\tau_{\mathrm{ArAr}}}\right)$

[1] Number of collisions to simulate

Table 6.4: Computation parameters comparison of various simulation approaches that all employ DSMC to simulate heavy particles. For each numerical parameter, the corresponding constraining physical parameter is indicated between parenthesis.

6.1.2 Outline

In the following, the fluid model that will be used for the electrons is presented in detail and the three governing equations (mass, momentum and energy) are derived from the Boltzmann equation. The particle in cell formulation that is used to calculate the charge on the computational grid and to move charged particles is then presented along with details about the implementation of the hybrid method. The computational test case used to investigate the hybrid method is then presented. It is fairly similar to the one used to showcase the adaptive method in Chapter V as it consists of two counterflowing jets and only three species. The geometry is, however, different as the spacing between the two jets is reduced, their diameter increased while the flow Knudsen number (based on the jet radii) is increased from 0.015 to 0.17 so as to place ourselves in the Knudsen number range where the hybrid method has previously been shown to produce "acceptable" simulation results [33], [44]. This particular flow, like most weakly ionized rarefied flows, is characterized by both large number density disparities between species in each individual cell of the domain as well as by large spatial variations throughout the domain. Because such type of flow stands most to gain from the use of the adaptive procedure presented in Chapter V, the latter is employed in the simulation. Finally, results obtained with the combined hybrid/adaptive methods (without energy equation) are presented.

6.2 Electron Fluid Model

In this section, fluid equations will be derived for a "simplified" electron fluid with the following characteristics:

1. No ionization and no other inelastic collisions with heavy species.
2. Undergoes elastic collisions with only one ion species, Ar^+, and one neutral species, Ar.

3. No magnetic field $\vec{B} = \vec{0}$.

4. In steady state.

5. In equilibrium at temperature $T_e\left(\vec{x}\right)$.

6.2.1 Fluid equations

In this section, we derive transport equations for the number density, momentum and energy of a generic gas that is governed by the Boltzmann equation which was previously introduced in Chapter II. The system of three equations that is in turn obtained is then specifically written for the electrons for the plasma regime under consideration and closed by a series of physical considerations. We consider the Boltzmann equation, Eq. (2.42), in ∇-notation where particles are subjected to an external force field $\vec{F}\left(\vec{x}, t\right)$

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f + \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}} f = Q\left(f, f\right) \tag{6.2}$$

We first seek to obtain transport equations for any general function $A\left(\vec{x}, \vec{v}, t\right)$ which only depends on time t, space $\vec{x}$ and velocity $\vec{v}$ from the Boltzmann equation. Multiplying Eq. (6.2) by A, and integrating over velocity space the following equation is obtained:

$$\overbrace{\int\limits_{\mathbb{R}^3} A\, \frac{\partial f}{\partial t} d\vec{v}}^{\text{(I)}} + \overbrace{\int\limits_{\mathbb{R}^3} A\ \vec{v} \cdot \nabla f d\vec{v}}^{\text{(II)}} + \overbrace{\int\limits_{\mathbb{R}^3} A\, \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}} f d\vec{v}}^{\text{(III)}} = \overbrace{\int\limits_{\mathbb{R}^3} A\ Q\left(f, f\right) d\vec{v}}^{\text{(IV)}} \tag{6.3}$$

Defining the velocity space averaging operator $\langle \cdot \rangle$ as follows:

$$\langle A \rangle \left(\vec{x}, t\right) \triangleq \int\limits_{\mathbb{R}^3} A\left(\vec{x}, \vec{v}, t\right) d\vec{v}, \tag{6.4}$$

and using the fact that

$$n \triangleq \langle 1 \rangle = \int\limits_{\mathbb{R}^3} f d\vec{v}, \tag{6.5}$$

the various terms of Eq. (6.3) can be rewritten as:

$$\text{(I)} = \int\limits_{\mathbb{R}^3} A \frac{\partial f}{\partial t} d\vec{v} = \frac{\partial}{\partial t} \left[n \langle A \rangle \right] - n \left\langle \frac{\partial A}{\partial t} \right\rangle, \tag{6.6}$$

$$\text{(II)} = \int\limits_{\mathbb{R}^3} A \ \vec{v} \cdot \nabla f d\vec{v} = \nabla \cdot \left[n \langle \vec{v} A \rangle \right] - n \langle \vec{v} \cdot \nabla A \rangle, \tag{6.7}$$

and

$$\text{(III)} = \int\limits_{\mathbb{R}^3} A \ \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}} f d\vec{v} = -n \left\langle \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}} A \right\rangle, \tag{6.8}$$

which can be found using an integration by parts if

$$\frac{\partial F_x}{\partial v_x} = \frac{\partial F_y}{\partial v_y} = \frac{\partial F_z}{\partial v_z} \tag{6.9}$$

which is indeed true for any force due to an electric field, which is independent of velocity, or perpendicular to it, such as a magnetic field ($\vec{v} \times \vec{B}$). Inserting the results of Eqs. (6.6), (6.7) and (6.8) into Eq. (6.3), the general transport equation for $A(\vec{x}, \vec{v}, t)$ is obtained:

$$\frac{\partial}{\partial t} \left(n \langle A \rangle \right) - n \left\langle \frac{\partial A}{\partial t} \right\rangle + \nabla \cdot \left[n \langle \vec{v} A \rangle \right] - n \langle \vec{v} \cdot \nabla A \rangle - n \left\langle \frac{\vec{F}}{m} \cdot \nabla_{\vec{v}} A \right\rangle = \int\limits_{\mathbb{R}^3} A \ Q(f, f) \, d\vec{v} \tag{6.10}$$

which will allow three equations (continuity, momentum and energy) to be derived for a function $A(\vec{v})$ that is respectively a zero, first or second order function of $\vec{v}$.

6.2.1.1 Continuity equation

This equation is derived by assuming that $A(\vec{v}) = 1$ so that Eq. (6.10) yields

$$\frac{\partial}{\partial t}(n) + \nabla \cdot (n\vec{u}) = \int_{\mathbb{R}^3} Q(f, f)\, d\vec{v} \tag{6.11}$$

where bulk velocity $\vec{u}$ is defined as

$$\vec{u} \triangleq \langle \vec{v} \rangle = \int_{\mathbb{R}^3} \vec{v} f d\vec{v}. \tag{6.12}$$

If no chemical reactions occur such as in previous chapters of this thesis, the number of particles over velocity space considered as a whole does not change so that the collision term of Eq. (6.13) is null. To present the most general result possible, we momentarily assume that ionization and recombination are present in the flow before applying the equation to the present electron fluid where both are absent. The frequency of ionization of species j is denoted by $\nu_{\text{ion},j}$ while the volumetric frequency of electron destruction by collision with species j is denoted by $\nu_{\text{r},j}$ so that

$$\int_{\mathbb{R}^3} Q(f, f)\, d\vec{v} = \sum_{j=1}^{N_{\text{spec}}} \left(\nu_{\text{ion},j} - \nu_{\text{r},j}\right) n \tag{6.13}$$

where the summation is made over all species for conciseness and rates are set to 0 when the reaction considered does not occur so that Eq. (6.13) becomes:

$$\frac{\partial}{\partial t}(n_e) + \nabla \cdot (n_e\vec{u}_e) = \sum_{j=1}^{N_{\text{spec}}} \left(\nu_{\text{ion},j} - \nu_{\text{r},j}\right) n_e \tag{6.14}$$

We now apply this general equation for the electron fluid of the plasma under consideration, assuming steady state and neglecting both ionization and recombination so that Eq. (6.14) becomes

$$\nabla \cdot (n_e\vec{u}_e) = 0. \tag{6.15}$$

6.2.1.2 Momentum equation

The transport equation for the particle momentum of a generic gas governed by the Boltzmann equation is obtained by setting A to $m\vec{v}$ in Eq. (6.10)[2]:

$$\frac{\partial}{\partial t}\left(mn\left\langle \vec{v}\right\rangle\right) + \nabla\cdot mn\left\langle \vec{v}\vec{v}\right\rangle - n\left\langle \vec{F}\right\rangle = \int\limits_{\mathbb{R}^3} m\vec{v}\, Q\left(f,f\right) d\vec{v} \tag{6.16}$$

The (total) velocity $\vec{v}$ is decomposed into its bulk $\vec{u}$ and peculiar $\vec{u}'$ parts

$$\vec{v} = \vec{u} + \vec{u}', \tag{6.17}$$

where

$$\vec{u}' \triangleq \left\langle \vec{v} - \left\langle \vec{v}\right\rangle\right\rangle. \tag{6.18}$$

The tensor product $\vec{v}\vec{v}$ can itself be decomposed as follows:

$$mn\ \left\langle \vec{v}\vec{v}\right\rangle = mn\ \vec{u}\vec{u} + \overline{\overline{\Psi}}, \tag{6.19}$$

where $\overline{\overline{\Psi}}$ is the kinetic part of the pressure tensor which only depends on thermal motion

$$\overline{\overline{\Psi}} \triangleq mn\left\langle \left(\vec{v} - \left\langle \vec{v}\right\rangle\ \right)\left(\vec{v} - \left\langle \vec{v}\right\rangle\right)\right\rangle = mn\left\langle \vec{u}'\vec{u}'\right\rangle, \tag{6.20}$$

so that using the continuity equation Eq. (6.16) can be rewritten as

$$mn\left(\frac{\partial}{\partial t} + \vec{u}\cdot\nabla\right)\vec{u} = -\nabla\cdot\overline{\overline{\Psi}} + n\left\langle \vec{F}\right\rangle + \int\limits_{\mathbb{R}^3} m\vec{v}\, Q\left(f,f\right) d\vec{v}. \tag{6.21}$$

This equation is now specialized to the simplified electron fluid under consideration, i.e., a fluid that follows a Maxwellian distribution, with bulk velocity $\vec{u}_e$ and temperature T_e, that undergoes elastic and inelastic collisions with heavy atomic species

---

[2] The dyadic product $\otimes$ will be omitted for conciseness in the following so that $\vec{v}\vec{v} \equiv \vec{v}\otimes\vec{v}$.

and which is subjected to an electric field $\vec{E}$ without magnetic field $\vec{B}$. Under such conditions, each particle of charge q_e is subjected to an external Lorentz force $\vec{F}$:

$$\vec{F} = q_e \vec{E} \tag{6.22}$$

This allows the simplification of Eq. (6.21) thus contributing to the closure of the system of equations. As the fluid is isotropic and in equilibrium, $\left\langle u_x^{'} \right\rangle = \left\langle u_y^{'} \right\rangle = \left\langle u_z^{'} \right\rangle = \frac{k_B T_e}{m_e}$, so that the pressure tensor can be rewritten as follows:

$$\overline{\overline{\Psi}} = n_e k_B T_e \; \bar{\bar{\mathrm{I}}} = p_e \; \bar{\bar{\mathrm{I}}} \tag{6.23}$$

where pressure p_e is thus defined as

$$p_e = n_e k_B T_e. \tag{6.24}$$

The collision term on the right for elastic collisions can be rigorously shown by integration to be given [70] by:

$$\int\limits_{\mathbb{R}^3} m\vec{v} \; Q_{\mathrm{elas}} \left(f, f\right) d\vec{v} = P_{ej} = \sum_{j=1}^{N_{\mathrm{spec}}} n_e \mu_{ej} \nu_{ej} \left(\vec{u}_j - \vec{u}_e\right) \tag{6.25}$$

where ν_{ej} designates the collision frequency between the electrons and species j and as $m_e \ll m_j$, the reduced mass μ_{ej} will be approximated by m_e in the following. This approximation of the collision rate can also be derived following a more physical approach as is done in [108] [140] [128] by interpreting it as a *friction term.* The term for inelastic collisions can further be shown to be identical to that for elastic collisions [70]. Thus using Eqs. (6.23) and (6.25), the momentum equation can be rewritten as

$$n_e m_e \left(\frac{\partial}{\partial t} + \vec{u}_e \cdot \nabla\right) \vec{u}_e = -\nabla p_e + q_e n_e \vec{E} + \sum_{j=1}^{N_{\mathrm{spec}}} n_e m_e \nu_{ej} \left(\vec{u}_j - \vec{u}_e\right) \tag{6.26}$$

which we proceed to further simplify by applying it to the plasma system at hand. As only 3 species are present (neutrals, ions and electrons), we only consider elastic collisions between electrons and neutrals and electrons and ions that is P_{en} and P_{ei}. Steady state is assumed, so that the time derivative terms drop out of the equation. The convective term $\vec{v}_e \cdot \nabla \vec{v}_e$ is further assumed to be negligible compared to the other terms of the equations, so that Eq. (6.26) becomes

$$0 = -\nabla p_e + q_e n_e \vec{E} + n_e \mu_{en} \nu_{en} \left(\vec{u}_n - \vec{u}_e\right) + n_e m_e \nu_{ei} \left(\vec{u}_i - \vec{u}_e\right). \tag{6.27}$$

We further neglect the bulk velocity of heavy particles u_i and u_n compared to that of the electrons $\vec{u}_e$ and obtain the final form of the momentum equation that will later be considered:

$$-\nabla \left(n_e k_B T_e\right) + q_e n_e \vec{E} - n_e m_e \left(\nu_{en} + \nu_{ei}\right) \vec{u}_e = 0 \tag{6.28}$$

The derivation of the equilibrium collision rates ν_{ei} and ν_{en} is rather involved and beyond the scope of this presentation. The interested reader is referred to [43] for a complete overview of the subject. The electron-ion collision rate ν_{ei} can be found, from basic scattering and plasma physics considerations, to be well approximated by the following [108] p. 58:

$$\nu_{ei} = n_i \frac{4\sqrt{2\pi}}{3} \left(\frac{m_e}{k_B T_e}\right)^{3/2} \left(\frac{e^2}{4\pi\varepsilon_0 m_e}\right)^2 \log \left[12\pi \left(\frac{T_e^3}{n_e}\right)^{1/2} \left(\frac{\varepsilon_0 k_B}{q_e^2}\right)^{1/2}\right], \tag{6.29}$$

whereas the electron-neutral collision rate can be obtained [108] p. 45 by approximating $\nu_{en} = n_n \overline{\sigma_{en} c_r}$ as

$$\nu_{en} = \frac{4\ n_n}{3} \left(\frac{8 k_B T_e}{\pi m_{en}}\right)^{1/2} \sigma_{en}\left(T_e\right) \tag{6.30}$$

where σ_{en} denotes the electron neutral elastic cross section. The electron-argon $\sigma_{e\mathrm{Ar}}$ elastic cross section used in this work is that compiled in [117].

6.2.1.3 Energy equation

The final equation of the system is obtained by writing a transport equation for particle kinetic energy by letting $A = \frac{1}{2}M\vec{v}^2$ in Eq. (6.10) which in turn yields

$$\frac{\partial}{\partial t}\left(\frac{1}{2}nm\left\langle \vec{v}^2\right\rangle\right) + \nabla\cdot\left[\frac{1}{2}nm\left\langle \vec{v}^2\vec{v}\right\rangle\right] - n\left\langle\frac{\vec{F}}{m}\cdot\vec{v}\right\rangle = \int\limits_{\mathbb{R}^3}\frac{1}{2}m\vec{v}^2\; Q\left(f,f\right)d\vec{v}. \tag{6.31}$$

The total kinetic energy flux vector $\nabla\cdot\left[\frac{1}{2}nm\left\langle\vec{v}\vec{v}^2\right\rangle\right]$ can be decomposed, c.f. [49] p. 46, by splitting velocity $\vec{v}$ into its bulk $\vec{u}$ and peculiar $\vec{u}'$ parts as done in Eq. (6.17):

$$\frac{1}{2}nm\left\langle\vec{v}\vec{v}^2\right\rangle = \left(\frac{1}{2}nm\vec{u}^2 + \frac{1}{2}\,\mathrm{Tr}\,\overline{\overline{\Psi}}\right)\vec{u} + \vec{u}\cdot\overline{\overline{\Psi}} + \vec{q}, \tag{6.32}$$

where the heat flux $\vec{q}$ is defined as

$$\vec{q} = \frac{1}{2}nm\left\langle\vec{v}'\vec{v}'^2\right\rangle \tag{6.33}$$

and obtained via Fourier's law

$$\vec{q} = -\kappa\nabla T, \tag{6.34}$$

with κ being the thermal conductivity of the fluid. We now proceed to specialize Eq. (6.31) to the case of the present electron fluid. Similarly as for the momentum equation, the elastic collision term can be explicitly calculated [70] by assuming all j species of the flow to be isotropic and in equilibrium with respective bulk velocity $\vec{v}_j$ and temperature T_j and if the collision frequency ν_{ej} is assumed to be independent of particle velocity:

$$\begin{aligned}\int\limits_{\mathbb{R}^3}\frac{1}{2}m\vec{v}^2\; C_{\mathrm{elas}}\left(f\right)d\vec{v} = 2\;\nu_{ej}\;n_j\frac{\mu_{ej}}{m_e+m_j}\Bigg[\frac{3k_B}{2}\left(T_j-T_e\right) + \frac{mv_j^2}{2} - \frac{mv_e^2}{2} + \\ + \frac{\left(m_e-m_j\right)}{2}\left(\vec{v}_e\cdot\vec{v}_j\right)\Bigg],\end{aligned} \tag{6.35}$$

which is valid for any species heavy species j of the flow. This expression may be further simplified [108] p. 51, if one assumes that, for all species, the ratio of diffusion to thermal speed is small, i.e. $\frac{|\vec{u}'|}{|\vec{u}|} \ll 1$, so that the temperature term dominates all the others on the right hand side of Eq. (6.35), so that the latter can thus be approximated as follows

$$R_{ej}^{\text{elas}} \approx \frac{2m_e}{m_j}\; n_j \nu_{ej} \left[\frac{3k_B}{2}\left(T_j - T_e\right)\right], \tag{6.36}$$

using $m_e \ll m_j$. The part of the collision integral dealing with inelastic collisions can be greatly simplified by assuming that particle kinetic energy is negligible compared to the energy lost or gained during the inelastic collision. The energy gained, viz. lost, can thus be approximated by the threshold energy of the inelastic collisions under consideration $\varepsilon_{\text{in,th}}$, i.e.

$$\int\limits_{\mathbb{R}^3} \frac{1}{2} m\vec{v}^2\; C_{\text{inel}}\left(f\right) d\vec{v} \approx R_{ei}^{\text{inel}} = -n_e\; \nu_{ej}\; \varepsilon_{\text{in,th}} \tag{6.37}$$

This assumption will certainly prove to be true for the ionization reaction (the only inelastic collision considered) in the flow that will later be simulated where $T_e \sim 2\text{eV}$ and $\varepsilon_{\text{in,th}} = 15.6$ eV. Thus inserting the expressions of Eqs. (6.32), (6.34), (6.35) and (6.37) into Eq. (6.31) written for a electron fluid, the following energy equation is obtained:

$$\begin{aligned}
\frac{\partial}{\partial t}\overbrace{\left(\frac{1}{2}n_e m_e \vec{u}_e^2\right)}^{\text{bulk KE}} + \frac{\partial}{\partial t}\overbrace{\left(\frac{3}{2}m_e k_B T_e\right)}^{\text{thermal KE}} &= -\nabla\cdot\overbrace{\left[\frac{1}{2}n_e m_e \vec{u}_e^2 \vec{u}_e\right]}^{\text{bulk KE flux}} - \nabla\cdot\overbrace{\left[\frac{3}{2}n_e k_B T_e \vec{u}_e\right]}^{\text{thermal KE flux}} \\
&\quad + \nabla\cdot\underbrace{\left(\kappa_e \nabla T_e\right)}_{\text{heat flux}} - \underbrace{p_e \nabla\cdot\vec{u}_e}_{\text{pressure work}} + \underbrace{q_e\; n_e\left(\vec{E}\cdot\vec{u}_e\right)}_{\text{Ohmic heating}} + \\
&\quad \underbrace{\sum_j \frac{m_e}{m_j}\; n_j \nu_{ej}\left[3k_B\left(T_j - T_e\right)\right]}_{R_{ej}^{\text{elas}}\text{(elastic collisions)}} + \underbrace{\sum_j n_e\; \nu_{ej}\; \varepsilon_{\text{ion,th}}}_{R_{ej}^{\text{inel}}\text{(inelastic collisions)}}.
\end{aligned} \tag{6.38}$$

The equation is further simplified by applying it to the electron fluid at hand where no inelastic collisions occur and only collisions with one ion and one neutral species at respective temperatures T_i and T_n are considered. Assuming steady state so that the time derivatives drop out of Eq. (6.38), neglecting bulk kinetic energy flux, using the continuity equation for n_e, and the definition of pressure of Eq. (6.24) the final form of the energy equation is obtained

$$\begin{aligned}\nabla \cdot \left[\frac{3}{2} n_e k_B T_e \vec{u}_e\right] &= \nabla \cdot \left(\kappa_e \nabla T_e\right) - \frac{k_B T_e}{m_e} \nabla \cdot \vec{u}_e + q_e\ n_e \left(\vec{E} \cdot \vec{u}_e\right) \\ &+ \left(\frac{3m_e}{m_i}\right)\ n_i \nu_{ei} \left[k_B \left(T_i - T_e\right)\right] + \left(\frac{3m_e}{m_n}\right)\ n_n \nu_{en} \left[k_B \left(T_n - T_e\right)\right] \quad (6.39)\end{aligned}$$

The electron thermal conductivity κ_e for the present three species partially ionized gas can be obtained by a mean free path type analysis [108], p. 94:

$$\kappa_e = \frac{2.4}{1 + \frac{\nu_{ei}}{\sqrt{2}\nu_e}} \frac{k_B^2 n_e T_e}{m_e \nu_e}, \tag{6.40}$$

where

$$\nu_e = \nu_{ei} + \nu_{en} \tag{6.41}$$

is the total electron heavy particle collision frequency, ν_{ei} and ν_{en} being respectively obtained with Eqs. (6.29) and (6.30).

6.2.2 Simplified fluid equation system

Three conservation equations were previously obtained for mass, Eq. (6.15) momentum, Eq. (6.28) and energy, Eq. (6.39):

$$\nabla \cdot \left(n_e \vec{u}_e\right) = 0 \tag{6.42}$$

$$-\frac{1}{n_e} \nabla \left(n_e k_B T_e\right) + q_e \vec{E} - n_e m_e \left(\nu_{en} + \nu_{ei}\right) \vec{u}_e = 0 \tag{6.43}$$

$$\nabla \cdot \left[\frac{3}{2} n_e k_B T_e \vec{u}_e\right] = \nabla \cdot (\kappa_e \nabla T_e) - \frac{k_B T_e}{m_e} \nabla \cdot \vec{u}_e + q_e \ n_e \left(\vec{E} \cdot \vec{u}_e\right)$$
$$+ \left(\frac{3m_e}{m_i}\right) \ n_i \nu_{ei} \left[k_B \left(T_i - T_e\right)\right] + \left(\frac{3m_e}{m_n}\right) \ n_n \nu_{en} \left[k_B \left(T_n - T_e\right)\right] \quad (6.44)$$

which form a closed system of equations with three unknowns $(n_e, \vec{u}_e, T_e)$. Because the system of equation is coupled and non-linear, its solution is non-trivial. Following the approach suggested in [108] and subsequently adopted by Boyd *et al* [29] [45], it can be recast in a different form to make its solution easier. This is done by assuming that the electron fluid follows potential flow and can thus be described by a potential function ψ such that

$$n_e \vec{u}_e = \nabla \psi, \quad (6.45)$$

and as $\frac{\partial \vec{B}}{\partial t} = \vec{0}$, by considering the electric potential ϕ instead of the electric field $\vec{E}$

$$\vec{E} = -\nabla \phi. \quad (6.46)$$

This allows Eqs. (6.42), (6.43) and (6.44) to be recast into three (easier to solve) Poisson equations. The first directly follows from the definition of ψ thru Eq. (6.45) whereby inserting it into Eq. (6.42) yields

$$\nabla^2 \psi = 0. \quad (6.47)$$

The second equation is obtained by taking the divergence of Eq. (6.43) and using (6.42) so as to obtain an equation for the potential ϕ defined by Eq. (6.46):

$$\nabla \cdot (\sigma \nabla \phi) = \frac{k_B}{q_e} \left[\sigma \nabla^2 T_e + \sigma T_e \left(\log n_e\right)\right] + \sigma \nabla \left(\log n_e\right) \nabla T_e + T_e \nabla \sigma \cdot \nabla \left(\log n_e\right) + \nabla \sigma \nabla T_e \quad (6.48)$$

where coefficient σ is defined as

$$\sigma = \frac{q_e^2 n_e}{m_e \nu_e} \quad (6.49)$$

which corresponds to the plasma conductivity [108] where ν_e is given by Eq. (6.41) The third and final equation of the system is obtained by simply rearranging Eq. (6.44) putting the Laplacian operator on the left hand side:

$$\begin{aligned}\nabla \cdot \left(\kappa_e \nabla T_e\right) = -\nabla \cdot \left[\frac{3}{2} n_e k_B T_e \vec{u}_e\right] - \frac{k_B T_e}{m_e} \nabla \cdot \vec{u}_e + q_e\ n_e \left(\nabla \phi \cdot \vec{u}_e\right) \\ + \left(\frac{3 m_e}{m_i}\right)\ n_i \nu_{ei} \left[k_B \left(T_i - T_e\right)\right] + \left(\frac{3 m_e}{m_n}\right)\ n_n \nu_{en} \left[k_B \left(T_n - T_e\right)\right]. \quad (6.50)\end{aligned}$$

The newly obtained system of equations thus consists of Eqs. (6.47), (6.48) and (6.50) which can be summarized as

$$\nabla \cdot \begin{bmatrix} \nabla \psi \\ \sigma \nabla \phi \\ \nabla T_e \end{bmatrix} = \begin{bmatrix} 0 \\ F\left(T_e, n_e\right) \\ G\left(T_e, n_e, \psi, \phi\right) \end{bmatrix}, \qquad (6.51)$$

where the two functions F and G on the right hand side are detailed in Eqs. (6.48) and (6.50). This completes the exposition of the governing equations of the full detailed fluid model such as was used in [29] and [45]. In the work presented in this thesis, the energy equation was, however, omitted and a constant electron temperature was used for the electron fluid. The energy equation was omitted because the finite element solver use to solve the system of equations cannot accurately deal with strong source terms with large spatial variations. The following system of equations is thus solved

$$\nabla \cdot \begin{bmatrix} \nabla \psi \\ \sigma \nabla \phi \end{bmatrix} = \begin{bmatrix} 0 \\ F\left(T_e, n_e\right) \end{bmatrix}. \qquad (6.52)$$

6.2.3 Solution method

6.2.3.1 Finite element solver

The calculation of the plasma potential using Eq. (6.52) requires the solution of two partial differential equations, i.e.

$$\nabla^2 \psi = 0 \tag{6.53}$$

and

$$\nabla \cdot \ \left[\ \sigma \left(n_e \left[x, y\right], n_n \left[x, y\right]\right) \nabla \phi\ \right] = F\left(T_e, n_e\left[x, y\right]\right). \tag{6.54}$$

In axisymmetric geometries, where variables are independent of the polar angle and where y is the radial direction, the divergence operator is defined, for an arbitrary $\vec{A} = A_x \hat{x} + A_y \hat{y}$ vector field, as

$$\nabla \cdot \vec{A} = \frac{1}{y} \frac{\partial}{\partial y} \left(y\ A_y\right) + \frac{\partial A_x}{\partial x}, \tag{6.55}$$

while no changes are required for the gradient operator compared to Cartesian geometries and the Laplacian operator of a scalar $f\left(x, y\right)$ is given by

$$f \equiv \nabla^2 f = \frac{1}{y} \frac{\partial}{\partial y} \left(\frac{\partial f}{\partial y}\right) + \frac{\partial^2 f}{\partial x^2}. \tag{6.56}$$

Using Eqs. (6.55) and (6.56), Eqs. (6.53) and (6.54) can respectively be written as

$$\frac{\partial \left(y\psi\right)}{\partial x^2} + \frac{\partial \left(y\psi\right)}{\partial y^2} = \nabla^2_{\mathrm{cart}} \psi = 0 \tag{6.57}$$

and

$$\frac{\partial^2 \left(y\sigma\phi\right)}{\partial x^2} + \frac{\partial^2 \left(y\sigma\phi\right)}{\partial y^2} = \nabla_{\mathrm{cart}} \cdot \left(y\sigma\nabla_{\mathrm{cart}}\phi\right) = yF \tag{6.58}$$

where ∇_{cart}, $\nabla_{\text{cart}}\cdot$ and $_{\text{cart}}$ are all the usual Cartesian differential operators. Eqs. (6.57) and (6.58) are both examples of a generalized Poisson equation, in that they can both be written as

$$\frac{\partial}{\partial x}\left[a\left(x,y\right)\frac{\partial u}{\partial x}\right]+\frac{\partial}{\partial y}\left[a\left(x,y\right)\frac{\partial u}{\partial y}\right]=b\left(x,y\right) \tag{6.59}$$

with $u = y\psi,\ a = 1,\ b = 0$ for Eq. (6.57) or $u = \phi, a\left(x,y\right) = y\sigma\left(x,y\right),\ b = yF$ in the case of Eq. (6.58). Eq. (6.59) is a fairly easy to solve elliptical partial differential equation [123] particularly amenable to the use of the finite element method. They were thus solved by using the generalized Poisson finite element solver detailed in [145] that was implemented by Cai [33]. It is first order (linear finite elements) for axisymmetric geometries which resultant linear system is solved via a biconjugate gradient method [122].

6.2.3.2 Differentiation operator

The application of the detailed fluid model requires the calculation of derivatives such as, for instance, to calculate the electric field $\vec{E}$ from the potential ϕ with $\vec{E} = -\nabla\phi$ after solving Eq. (6.52) for the potential, or to calculate the various derivatives terms such as $\nabla\sigma$ or $\nabla\log\left(n_e\right)$ on the right hand side of that same system. The value of the derivatives of variable $\{a_l\}_{l=1}^{N_l}$, defined at each node $\{\vec{x}_l\}_{l=1}^{N_l}$ are calculated for each of the three nodes of each cell $\vec{x}_l = (x_l, y_l)$ by assuming that

$$a\left(x,y\right)=a\left(\vec{x}_l\right)+\frac{\partial a}{\partial x}\left(x-x_l\right)+\frac{\partial a}{\partial y}\left(y-y_l\right) \tag{6.60}$$

so that the value of a at the two other nodes of cell i, $a\left(\vec{x}_{l+1}\right)$ and $a\left(\vec{x}_{l+2}\right)$ being known, the following system may be formed

$$\begin{bmatrix} x_{l+1}-x_l & y_{l+1}-y_l \\ x_{l+2}-x_l & y_{l+2}-y_l \end{bmatrix}\begin{bmatrix} \left(\frac{\partial a}{\partial x}\right)_{l,i} \\ \left(\frac{\partial a}{\partial y}\right)_{l,i} \end{bmatrix}=\begin{bmatrix} a\left(\vec{x}_{l+1}\right)-a\left(\vec{x}_l\right) \\ a\left(\vec{x}_{l+2}\right)-a\left(\vec{x}_l\right) \end{bmatrix}, \tag{6.61}$$

which solution provides the value of the derivatives of a at node l for cell i. The procedure is repeated for the two other nodes of the cell, nodes $l+1$ and node $l+2$ so that the values of $\frac{\partial a}{\partial x}$ and $\frac{\partial a}{\partial y}$ are known for the three nodes of the cell. As each node is shared between $N_{b,l}$ different cells, a total of $N_{b,l}$ different derivative values are thus obtained for a given node. The final value of the derivative is determined by taking the average value among the $N_{b,l}$ different cells to which node l belongs:

$$\left(\frac{\partial a}{\partial x}\right)_l = \frac{1}{N_{b,l}} \sum_{i'=1}^{N_{b,l}} \left(\frac{\partial a}{\partial x}\right)_{l,i'} \quad \text{and} \quad \left(\frac{\partial a}{\partial y}\right)_l = \frac{1}{N_{b,l}} \sum_{i'=1}^{N_{b,l}} \left(\frac{\partial a}{\partial y}\right)_{l,i'}. \tag{6.62}$$

6.3 DSMC/“PIC” model

Although the exact details of the formulation and the implementation of the detailed fluid model differs from that of the particle in cell method, both share many common features and challenges. For that reason, the PIC method is briefly introduced below and presented for a simple one-dimensional plasma. Following that presentation, the charge projection and force weighting schemes used for the detailed fluid model are presented in parallel with the corresponding PIC formulation. Finally, the integration of the fluid model within a standard DSMC code is presented.

6.3.1 The particle in cell method

The most commonly used particle simulation technique for ionized gas flows is the particle in cell method [78] [21]. A concise introduction to the method that inspired the presentation below can be found in [20], whereas a contemporary review of PIC can be found in [74] or [105]. In its purest form, PIC refers to the simulation of a gas for which motion is governed by the Boltzmann equation with the Lorentz force. The value of the electric and magnetic fields throughout the domain is obtained by solving field equations, usually one or several of Maxwell’s equation, such as for instance the Poisson equation. PIC is fairly similar to DSMC in its formulation in that it relies on the use of macroparticles to simulate a large number of actual physical particles and uses single particle motion to simulate the transport part of the Vlasov (Boltzmann in the case of DSMC) equation. The main difference from DSMC lies in the use of field equations and the use of the resultant fields to move particles and the treatment of collisions (or lack thereof in the case of PIC). The most simple example of the particle in cell method, and one that is often given to illustrate the technique, is to simulate a one-dimensional, uniform and unmagnetized plasma that only consists of electrons and ions each with an average number density of n_0. As ions are much heavier than electrons, they are assumed to be at rest providing a neutralizing background to the light fast moving electrons. The non-zero charge resulting from the motion of the electrons in the positively charged ion background results in the creation of an electric field $\vec{E} = -\nabla\phi$ via the Poisson equation:

$$\nabla \cdot \vec{E} = \frac{\rho}{\varepsilon_0} \overset{\text{1D}}{\Rightarrow} \frac{d^2\phi}{dx^2} = -\frac{q_e}{n_0}\left[n_0 - n\left(x\right)\right] \tag{6.63}$$

which in turn accelerates each individual electron with position $\vec{x}_j$ and velocity $\vec{v}_j$ according to:

$$m_e \frac{d^2\vec{x}_j}{dt^2} = q_e \vec{E}\left[\vec{x}_j\left(t\right)\right] \tag{6.64}$$

which in turn changes the charge distribution of the plasma by affecting the position of each individual particle in the domain and so on ad infinitum. Equivalently, one might also say that the velocity distribution function of the electrons $f\left(x, v_x, t\right)$ is governed by the Vlasov equation

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \nabla f + \frac{q_e \vec{E}}{m_e} \cdot \nabla_{\vec{v}} f = 0 \Rightarrow \frac{\partial f}{\partial t} + \vec{v_x}\frac{\partial f}{\partial x} + \frac{q_e E_x\left(x\right)}{m_e}\frac{\partial f}{\partial v_x} = 0 \tag{6.65}$$

where $\vec{E}$ is obtained by solving Eq. (6.63) and n, by definition is

$$n\left(x\right) = \int\limits_{\mathbb{R}^3} f\left(\vec{x}, \vec{v}, t\right) d\vec{v} \; = \int\limits_{-\infty}^{+\infty} f\left(x, v_x, t\right) dv_x. \tag{6.66}$$

The combined used of particles and field equations poses two main challenges which are at the heart of the formulation of the PIC method. The first is the matter of calculating the charge on the right hand side of the Poisson equation, i.e. $\rho\left(\vec{x}\right)$ in Eq. (6.63). This must evidently be done by using the position of the electrons inside the domain to reconstruct a continuous function $\rho\left(\vec{x}\right)$. The issue of calculating the continuous charge function from the position of (inherently discrete) particles is referred to in the literature as *charge weighting* or *charge interpolation*. Conversely, after the electric field charge is calculated by solving Eq. (6.63), it has to be applied to each individual particle of the domain. This step is often referred to as *force projection* or *force interpolation* in the literature. The difficulty lies in the fact that the electric field cannot be resolved at the exact location of each individual particle in the domain and is instead resolved on a discrete mesh so that the value of the electric field has to be interpolated to the location of each individual cell. Having briefly introduced the matter of charge and force interpolation, the two will now be discussed in detail within the context of the hybrid fluid/DSMC method.

6.3.2 Charge interpolation

We assume that the flow is quasi-neutral [104], i.e that the electron number density is at all times equal to the ion number density, so that all cells have no net charge at all time steps. As no electron particles are present in the simulation, the electron number density, a required input of the detailed fluid model, is directly obtain from the ion number density

$$n_e = n_i = n\left(\mathrm{Ar}^+\right). \tag{6.67}$$

In the following, the charge will always refer to the ion number density, bearing in mind that the net charge of all particles in a cell is always null. The problem of charge interpolation will thus consist in interpolating the ion number density to the grid. The problem of charge interpolation was previously introduced within the framework of the simple 1-D plasma example and will now be examined in greater detail. The unstructured triangular mesh is assumed to be composed of N_c cells with subscript $1 \leq i \leq N_c$ and N_l total cell vertices with subscript $1 \leq l \leq N_l$ located at $\{\vec{x}_l\}_{l=1}^{N_l}$. In the following, as triangular cells are used, each cell contains 3 vertices which are shared with other neighboring cells. We further assume that the simulation contains N_{tot} particles with subscript $1 \leq j \leq N_{tot}$ with positions $\vec{x}_j^k$ and velocities $\vec{v}_j^k$ at time step k. Because the charge density that is naturally obtained with discrete particle is too noisy:

$$\rho_N\left(\vec{x}, t\right) = q_e \int\limits_{\mathbb{R}^3} f_N\left(\vec{x}, \vec{v}, t\right) d\vec{v} = q \sum_{j=1}^{N} w_j \ \delta\left(\vec{x} - \vec{x}_j\right), \tag{6.68}$$

shape factors $S_l\left(\vec{x}\right)$ with compact support in $\mathbb{R}^2$, defined for each l node are used to calculate the interpolated charge $\rho_h\left(\vec{x}_l, t\right)$:

$$\rho_h\left(\vec{x}_l, t\right) = \int\limits_{\mathbb{R}^2} \rho_N\left(\vec{x}, t\right) S_l\left(\vec{x}\right) d\vec{x} = \frac{q}{\mathrm{V}\left(\vec{x}_l\right)} \sum_{j=1}^{N_p} S_l\left(\vec{x}_j\left(t\right)\right). \tag{6.69}$$

Commonly used shape factors for PIC include the 0th degree spline $S^{\text{2D},0} : \mathbb{R}^2 \to \mathbb{R}$ which, in the case of a two-dimensional cartesian mesh with uniform grid spacing Δx and Δy in the x and y directions, is given by:

$$S_l^{\text{2D},0}\left(x, y\right) = \frac{1}{\Delta x} S^{\text{1D},0}\left(\frac{x - x_l}{\Delta x}\right) \times \frac{1}{\Delta y} S^{\text{1D},0}\left(\frac{y - y_l}{\Delta y}\right) \tag{6.70}$$

with

$$S^{\text{1D},0}\left(\xi\right) = \begin{cases} 1 & \text{if } |\xi| \leq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}, \tag{6.71}$$

or the first order spline $S_l^{2\mathrm{D},1}\left(x,y\right)$ which is similarly defined using

$$S^{1\mathrm{D},1}\left(\xi\right)=\begin{cases}1-\left|\xi\right| & \text{if } \left|\xi\right|\leq 1\\ 0 & \text{otherwise}\end{cases}. \tag{6.72}$$

$S_l^{1\mathrm{D},0}$ and $S^{2\mathrm{D},1}$ are often referred to as nearest grid point (NGP) or cloud-in-cell (CIC) weighting, respectively. Details on other commonly used shape factors can be found in e.g. [141] or [21]. However, as our simulations are conducted with a triangular unstructured mesh, those commonly used shape factors cannot be utilized. Charge interpolation schemes for unstructured meshes are still an area of active research as evidenced by [85]. Because DSMC, by formulation, always employs a nearest grid point scheme to calculate any cell-based quantities, we have to resort to an original charge calculation scheme, which consists in two successive interpolation steps that are detailed below. In the first, the charge is calculated at the center of each DSMC cell while in the second that charge is interpolated to the vertices between DSMC cells.

In the first step, the charge is calculated at the center of each cell from the position of all charged particles. This formally corresponds to a particles → cell centers interpolation. This is done by adopting an NGP scheme on an unstructured mesh whereby each cell Ω_i is associated with the following shape factor:

$$S_i\left(\vec{x}\right)=\mathbf{1}_{\Omega_i}\left(\vec{x}\right), \tag{6.73}$$

where $\mathbf{1}_{\Omega_i}$ is the indicator function $\mathbf{1}_{\Omega_i}:\mathbb{R}^2\rightarrow\left\{0,1\right\}$ which is defined as follows for each cell Ω_i as

$$\mathbf{1}_{\Omega_i}(\vec{x}):=\begin{cases}1 & \text{if } \vec{x}\in\Omega_i\\ 0 & \text{if } \vec{x}\notin\Omega_i\end{cases}. \tag{6.74}$$

Following the definition of the charge interpolation of Eq. (6.69), the cell-centered charge function, denoted by $\rho_c(\vec{x}, t)$ is obtained by

$$\rho_c(\vec{x}, t) = \frac{W_p(\vec{x})}{\mathrm{V}(\vec{x})} \sum_{j=1}^{N_p} q_j \ \mathbf{1}_{\Omega_i}(\vec{x}_j(t)) \tag{6.75}$$

where $W_p(\vec{x})$ is the cellwise constant field of cell weights

$$W_p(\vec{x}) = \sum_{i=1}^{N_c} W_{p,i} \ \mathbf{1}_{\Omega_i}(\vec{x}) \tag{6.76}$$

while the cellwise constant volume function $\mathrm{V}(\vec{x})$ is similarly defined by using the volume of each individual cell, V_i, in Eq. (6.76) instead. A graphical depiction of the operator is shown in Fig. 6.1. One furthermore always requires that a charge interpolation scheme conserve total charge, i.e.

$$\sum_{i}^{N_c} V_i \ \rho_h(\vec{x}_i, t) = q \sum_{j}^{N_p} W_p(\vec{x}_j) \tag{6.77}$$

which, in the general case, by the definition of the charge of Eq. (6.69) can be accomplished by requiring that

$$\sum_{i}^{N_c} S_i(\vec{x}) = 1 \quad \forall \ \vec{x} \in \mathbb{R}^2 \tag{6.78}$$

which is indeed satisfied by the NGP and CIC schemes of Eqs. (6.71) and (6.72). Conservation of charge is, however, more difficult to achieve for axisymmetric geometries, even when cartesian grids are used, as the volume of cells scales linearly in the radial direction. For that reason, modifications have to be brought to standard interpolation schemes in the radial direction which are discussed for instance in [127]. This is, however, not a concern for the charge interpolation scheme used in present work as the shape function $S_l(\vec{x})$ of Eq. (6.73) does indeed satisfy Eq. (6.78).

In the second step of the charge interpolation procedures, the charge is interpolated from the cell centers to their vertices which are also designated as *nodes* in the

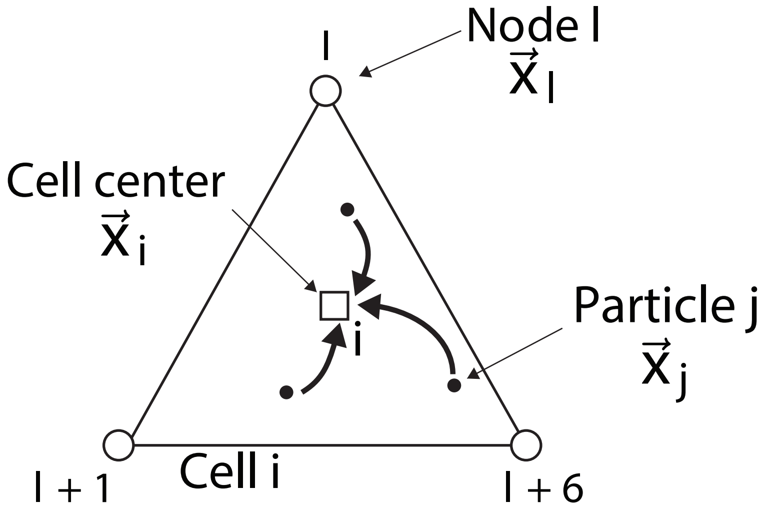


Figure 6.1: Schematic of the first step of the charge interpolation

following. It formally consists in applying interpolation operator P_n : cell centers $\hookrightarrow$ nodes which is defined as

$$P_n : \rho_c\left(\vec{x}\right) \rightarrow \rho_n\left(\vec{x}\right) \tag{6.79}$$

where

$$\rho_n\left(\vec{x}_l\right) = \frac{\sum_{i=1}^{N_c} \rho_c\left(\vec{x}_c\right) \mathbf{1}_{\Omega_i}\left(\vec{x}_l\right)}{\sum_{i=1}^{N_c} \mathbf{1}_{\Omega_i}\left(\vec{x}_l\right)}. \tag{6.80}$$

The value of the charge at a single node is thus obtained by averaging the value of the cell center charges of all the cells to which the node belongs as shown by the graphical depiction of Fig. 6.2. The overall charge interpolation operator, denoted by P_h, consists in the successive application of P_c and P_n:

$$P_h = P_n \circ P_c : \{\vec{x}\}_{j=1}^{N} \rightarrow \rho_h\left(\vec{x}\right) \tag{6.81}$$

which, by using their respective definitions given by Eqs. (6.75) and (6.80), can be explicitly written as:

$$\rho_h\left(\vec{x}_l, t\right) = \frac{\sum_{i=1}^{N_c} \left[\frac{W_p(\vec{x}_i)}{\mathrm{V}(\vec{x}_i)} \sum_j q_j \mathbf{1}_{\Omega_i}\left(\vec{x}_j\left(t\right)\right)\right] \mathbf{1}_{\Omega_i}\left(\vec{x}_l\right)}{\sum_{i=1}^{N_c} \mathbf{1}_{\Omega_i}\left(\vec{x}_l\right)}. \tag{6.82}$$

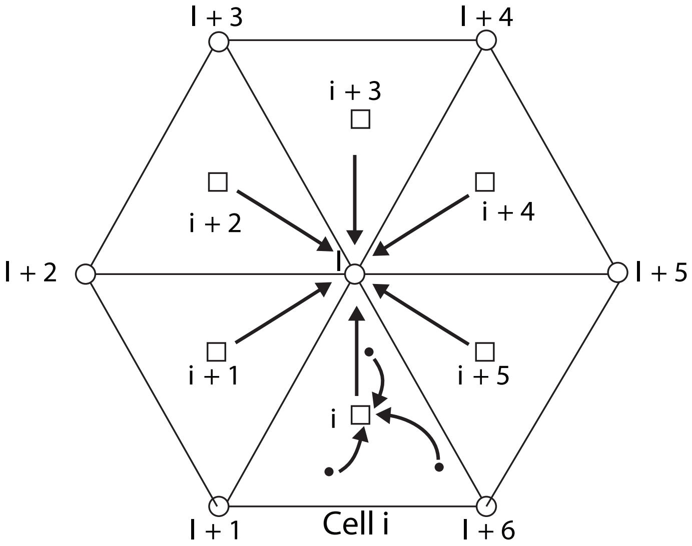


Figure 6.2: View of the second step of the charge interpolation

The instantaneous charge produced by the application of Eq. (6.82) is, however, inherently noisy due to the natural fluctuations of the number of particles in each individual cell of the DSMC simulation. This inherent noise makes it impossible to calculate the various quantities required for the application of the detailed fluid model, i.e. $n_e, \nabla n_e$ and Δn_e. The problem is further compounded by the need to calculate multiple derivatives of the charge, which further aggravates noise. The detailed fluid model was, however, derived with a steady state assumption (i.e. no time derivative), so that all quantities used in the model, such as the charge n_e should be independent of time. There is furthermore no physical basis for any unsteadiness in contract to PIC simulations, where the unsteadiness of the charge and fields are dictated by the governing equations so that it can for instance be used to simulate Landau damping [20], an inherently unsteady phenomenon. In order to reduce the unsteadiness of the charge, an exponential moving average is thus used [138]. It consists in calculating the time averaged charge $\overline{\rho}_h\left(\vec{x}_l, t\right)$ as follows

$$\overline{\rho}_h\left(\vec{x}_l, t^{k+1}\right) = R\,\rho_h\left(\vec{x}_l, t^{k+1}\right) + (1 - R)\,\overline{\rho}_h\left(\vec{x}_l, t^{k}\right). \tag{6.83}$$

where R is a relaxation factor such that $R \ll 1$. The relaxation factor should be chosen small enough so as to minimize noise but also large enough so as to allow the flow to develop reasonably fast from its initial conditions to steady state. The time-averaged charge $\overline{\rho}_h\left(\vec{x}_l, t\right)$ is in turn used to calculate the plasma potential according to Eq. (6.52). The detailed fluid model also requires that the number density of all neutrals be interpolated to the grid so as to calculate the plasma conductivity σ following Eq. (6.49). The exact same interpolation and averaging procedures, respectively given by Eqs. (6.82) and (6.83) as for the charge are used for the neutral number density.

Various charge interpolation schemes have been tested and all have been observed to fare no better than that of (6.82) which is admittedly fairly crude when compared to traditionally used charge interpolation schemes such as CIC or higher order schemes [21]. This situation is no doubt due to the fact that the charge interpolation/force interpolation presented here is not a “real” PIC procedure because of the time averaging of the charge and the use of fluid equations instead of Maxwell’s equations.

6.3.3 Force Interpolation

Solving the system of fluid equations of Eq. (6.52) yields the plasma potential ϕ at all the nodes of the computational domain $\{\vec{x}_l\}_{l=1}^{N_l}$, from which the electric field is obtained with

$$\vec{E}\left(\vec{x}_l\right) = -\nabla\phi. \tag{6.84}$$

using the discrete differentiation operator of Eqs. (6.61) and (6.62). This electric field in turn accelerates all the ions of the domain. As it varies through space, it has to be interpolated to the location of each individual ion of the simulation in a process analogous to the charge interpolation discussed previously. In an actual PIC simulation, this would be accomplished by using the same shape factor as used for the charge interpolation, so as to prevent the occurrence of non-physical self-forces on particles [78], with:

$$\vec{E}\left(\vec{x}_j, t\right) = \sum_{l}^{N_l} \vec{E}\left(\vec{x}_l, t\right) S_l\left(\vec{x}_j\left(t\right)\right). \tag{6.85}$$

If we were to use the exact same charge interpolation scheme that was previously presented in reverse, $\left\{\vec{E}\left(\vec{x}_l\right)\right\}_{l=1}^{N_l}$ would first have to be interpolated from the nodes to the cell centers to obtain $\left\{\vec{E}\left(\vec{x}_i\right)\right\}_{i=i}^{N_c}$. $\vec{E}\left(\vec{x}_i\right)$ would then be used to accelerate all particles inside cell Ω_i regardless of their position inside the cell which corresponds to the use of an NGP scheme in Eq. (6.85). In order to increase the accuracy of the force interpolation scheme, a different interpolation scheme than for the charge is utilized. It consists in weighting the value of the electric field of each cell node by the area described by the triangle obtained by joining the position of the particle to the two opposite nodes. Following the notations of the graphical depiction of the scheme in Fig. 6.3, the value of the electric field $\vec{E}$ at particle position $\vec{x}_j$ is obtained by

$$\vec{E}\left(\vec{x}_j, t\right) = \frac{1}{\Omega_i}\left(\Omega_{i,j,l_1}\ \vec{E}\left(\vec{x}_{l_1}, t\right) + \Omega_{i,j,l_2}\ \vec{E}\left(\vec{x}_{l_2}, t\right) + \Omega_{i,j,l_3}\ \vec{E}\left(\vec{x}_{l_3}, t\right)\right) \tag{6.86}$$

The choice of such a scheme is a posteriori justifiable by the better results that it is seen to produce compared to the two-step interpolation process described above, bearing in mind the occurrence of any self force is not an issue due to the time averaging of the charge of Eq. (6.83). Once the value of the electric field at the location of the particle has been determined, it is first accelerated at its initial location

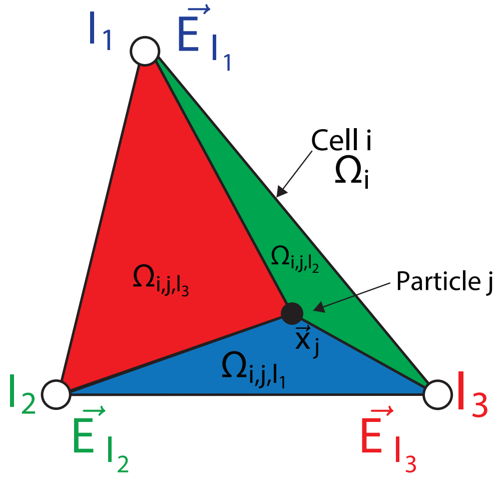


Figure 6.3: Graphical view of the force interpolation procedure

$\vec{x}_j^k$ using a first order Euler-explicit scheme to obtain its post-acceleration velocity $\vec{v}_j^{k+1/2}$,

$$\vec{v}_j^{k+1/2} = \vec{v}_j^k + \frac{q_e \Delta t}{m_j} \vec{E}\left(\vec{x}_j^k\right) \tag{6.87}$$

while its position remains unchanged and then moved following the standard DSMC particle move procedure for axisymmetric geometries described in [15] p. 370, again with a time step of Δt.

6.4 DSMC/Fluid model coupling

6.4.1 Coupling strategy

As previously mentioned, because of its derivation, the fluid model is not intended to capture any time dynamics of the coupling between the electron/ion motion and the electric field. To reduce possible effects of the inherent unsteadiness of DSMC on the value of the charge, the charge is averaged via a relaxation factor as indicated by Eq. (6.83). The same averaging procedure is also employed for the plasma potential which is calculated by solving the following equation (discussed in section 6.2.2)

$$\nabla \cdot (\sigma \phi) = F. \tag{6.88}$$

Every time this equation is solved at time step k, it provides a new value $\left\{\phi_l^{k+1}\right\}_{l=1}^{N_l}$ for the plasma potential for all nodes which is in turn used to update the value of the average potential $\left\{\overline{\phi}_l^{k+1}\right\}_{l=1}^{N_l}$ by using:

$$\overline{\phi}_l^{k+1} = R\ \phi_l^{k+1} + (1 - R)\ \overline{\phi}_l^{k} \tag{6.89}$$

The average potential thus obtained is then used to calculate the electric field via Eq. (6.84) which is itself used to move the ions following the force interpolation procedure discussed in Section 6.3.3.

6.4.2 Implementation

The overall implementation of the DSMC/fluid model is summarized in the following pseudo-code:

```
For time steps, $k = 0, ..., N_{\text{step}}$, do:
    Every $N_{\textbf{update}}$ steps do:
        i) Solve fluid equations of Eq. (6.52) using method of section 6.2.3.1 to calculate $\phi^k$
                                                                                   } Fluid Model
        ii) Update average plasma potential $\overline{\phi}^k$ with Eq. (6.89)
    end
    For all cells, $i = 1, ..., N_c$, do:
        i) Interpolate charge and neutral density with Eq. (6.82)
        ii) Update average charge and neutral density with Eq. (6.83)
        iii) CountParticles()
        iv) CalculateCollisionPairs()                                              } DSMC
        v) PerformCollisions()
        vi) Accelerate ions with Eq. (6.87)
        vii) MoveParticles()
        viii) ApplyBCs()
    end
end
```

All the steps related to the fluid model are indicated in bold fonts. “DSMC” refers to the standard DSMC implementation time step loop that was already mentioned in Section II. Following initial design choices [33], the fluid model and the DSMC procedure are relatively independent of one another. The computational overhead introduced by the used of the fluid model varies greatly depending on the value of N_{update}. This issue will be discussed when results are presented.

6.5 Test Case

6.5.1 Geometry and boundary conditions

The test case used to investigate the adaptive method is shown in Fig. (6.4). It consists of plasma source, *jet 1*, which emits neutral argon $\text{Ar}, 1$ (i.e. argon with a species tag of 1), argon ions and electrons. Assuming quasi-neutrality, the electrons and ions have the same number density and bulk velocity at the outlet of the plasma source. Opposite the plasma source, *jet 2*, emits neutral argon $\text{Ar}, 2$ (i.e. argon with a species tag of 2). The flow conditions for the two jets are reported in Table 6.5. All species are in equilibrium (Maxwellian VDF) with no bulk velocity in the y or z directions when emitted. The Knudsen numbers of the two jets based on their diameters and using the VHS model is 0.17. Species $\text{Ar}, 1$ and $\text{Ar}, 2$ are both emitted at $M = 1$, while ions (and electrons) are emitted with $M = 1.92$. A greater velocity in the part of the ions is fairly typical of what is observed at the outlet of plasma emitting devices such as hollow cathodes. Very often neutral gases are emitted by a nozzle-like geometry functioning in the choked regime so that the gas is at sonic conditions, as was chosen here. Charged particles, such as ions or electrons, can be accelerated before their emission with an electric field such as in hollow cathode devices [28] [69] and their speeds not limited by their gas dynamics behavior as is the case for neutral particles. Supersonic outflow boundary conditions are used for particles at the left, top and right of the computational domain which would correspond to the wall of the vacuum chamber if an experiment of this test case were conducted. Axisymmetry in the DSMC simulation is enforced by the particle move procedure (see [15] p. 370) while specular reflection boundary conditions are enforced for particles at the wall. Boundary conditions used for the fluid model are summarized in Table 6.6. The plasma potential at the source is set to 5 V while that of all other boundaries is set to 2V (except the centerline where $\frac{\partial \phi}{\partial y}(x, y = 0) = 0$ to enforce axisymmetry). The positive plasma potential at the exit of the plasma source corresponds to what is typically observed at the exit of many low temperature plasma sources, such as, e.g. hollow cathodes [69]. The drop in plasma potential between the plasma source exit and other boundaries combines with the gas expansion to further accelerate the ions inside the domain. From the quasi-neutrality assumption $n_e = n_i = n_{\mathrm{Ar}^+} = 2.5 \times 10^{19}$ m^{-3} at the exit of the plasma source while previous measurements and simulations [94] have shown the ratio of electron bulk to thermal velocities to be about 0.5-0.8 at the exit of a hollow cathode, so that assuming $\frac{u_{e,x}}{u_e'} = 0.5$ with $T_e = 2$eV, yields $u_{e,x} = 4.7 \times 10^5$ [m/s]. By definition of the electron stream function, $\frac{\partial \Psi}{\partial x} = n_e u_{e,x}$ so that using the previously mentioned values for n_e and $u_{e,x}$, a value of 1.175 m^{-1}s^{-1} is obtained at the exit of the plasma source which is reported in table 6.6. The jet centerline and wall are assumed to correspond to an electron streamline with $u_{e,y} = 0$ so that $\frac{\partial \Psi}{\partial y} = 0$ is used along those boundaries. The same zero normal velocity boundary condition is used for the left and top outflow boundaries and jet B while the value of Ψ at the right is outflow boundary chosen so as to obtain streamlines that are parallel to the x direction.

	Number density [m^{-3}]	x-Bulk velocity [m/s]	Temperature [K]
Jet 1			
$\mathrm{Ar}, 1$	2.5×10^{21}	322	300
Ar^+	2.5×10^{19}	620	300
Jet 2			
$\mathrm{Ar}, 2$	2.5×10^{21}	-322	300

Table 6.5: Jet 1 and 2 inflow conditions.

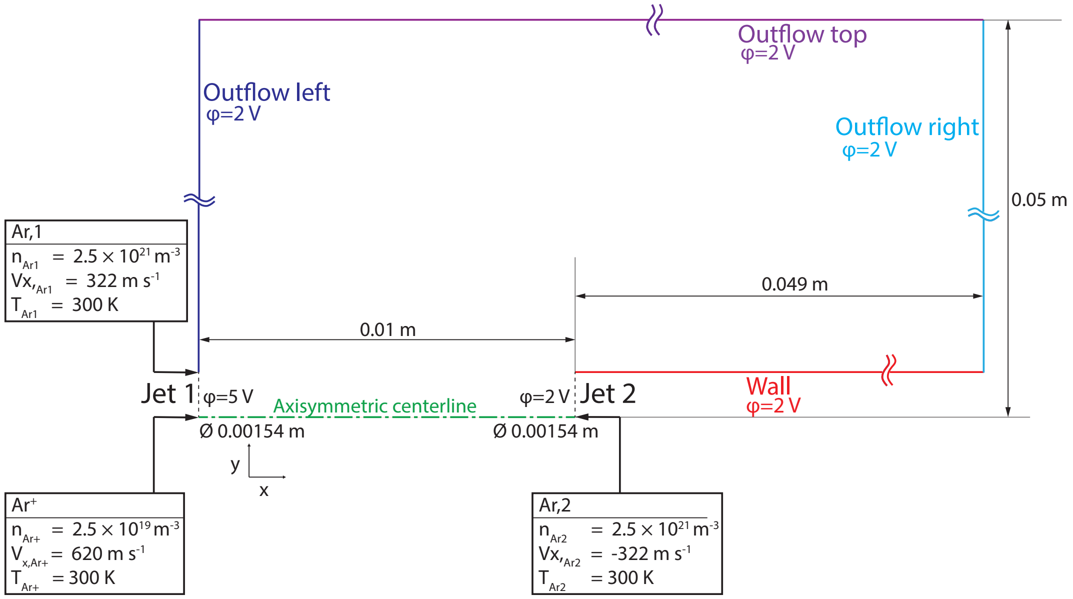


Figure 6.4: Schematic of test case with boundary conditions and dimensions.

Boundary	DSMC boundary condition	Potential	Stream function Ψ
Jet A	See Table 6.5	$\phi = 5$V	$\frac{\partial\psi}{\partial x} = -1.175 \times 10^{25}$ $\left[\mathrm{m}^{-2}\mathrm{s}^{-1}\right]$
Outflow left	Supersonic outflow	$\phi = 2$V	$\frac{\partial\psi}{\partial x} = 0$
Outflow top	Supersonic outflow	$\phi = 2$V	$\frac{\partial\psi}{\partial y} = 0$
Outflow right	Supersonic outflow	$\phi = 2$V	$\psi = -1.175 \times 10^{20}$ $\left[\mathrm{m}^{-1}\mathrm{s}^{-1}\right]$
Wall	Specular reflection	$\phi = 2$V	$\frac{\partial\psi}{\partial y} = 0$
Jet B	See Table 6.5	$\phi = 2$V	$\frac{\partial\psi}{\partial x} = 0$
Centerline	Axisymmetric symmetry	$\frac{\partial\phi}{\partial y} = 0$	$\frac{\partial\psi}{\partial y} = 0$

Table 6.6: Simulation boundary conditions.

6.5.2 Numerical method and parameters

The flow was simulated using both the detailed fluid model that was detailed in Section 6.2 and the adaptive procedure for the time step, cell weights and species relative weights that is the subject of Chapter V. The various numerical parameters of the fluid models are summarized in Table 6.7 while those of the adaptive technique are given in Table 6.8. The mesh consists of 5,028 triangular cells while the total number of particles for each species is reported in Table 6.8. Collisions between all particles, including the Ar^+ ions, are modeled via the variable hard sphere (VHS) model [15] for argon with a reference diameter of 4.17×10^{-10} m at 273 K and a temperature exponent of 0.81. The sophisticated DSMC pair selection scheme of Eq. (5.37) is used while further details concerning the implementation of the DSMC method can be found in [91].

Designation	
Relaxation factor R (Eq. (6.83))	10^{-4}
Frequency of update of electric field, N_{update}	5
Constant electron temperature, T_e	2 eV

Table 6.7: Parameters of detailed fluid model.

Designation	A800ION
$W_{p,0}$	$4,884,125$
Δt_0 [s]	5.0×10^{-9}
Adaptive $\widetilde{\Delta t}$	YES
Adaptive $\widetilde{W_p}$	YES
Adaptive W_{rel}	YES
$N_{p,\mathrm{want},1,2,3}$	800
Total realized number of particles (M)	16.5
Realized number of particles for species (1,2,3) (M)	(4.5,4.5,6.9)
$\left[\widetilde{\Delta}_{t,\min}, \widetilde{\Delta}_{t,\max}\right]$	$\left[10^{-12}, 10^{12}\right] \left(8.8\ 10^{-4}, 105.4\right)$
$\left[\widetilde{W}_{p,\min}, \widetilde{W}_{p,\max}\right]$	$\left[2^{-10}, 2^{50}\right] (1, 32)$
$W_{\mathrm{rel},\min,1,2,3}$	$\left[10^{-4}, 10^{-4}, 10^{-7}\right], \left(0.017, 0.004, 5.310^{-5}\right)$
$W_{\mathrm{rel},\max,1,2,3}$	$[1, 1, 0.0001], (1, 1, 0.0001)$

Table 6.8: Parameters of adaptive technique parameters used. The set values of the bounds are shown between brackets while actually observed values are reported between parenthesis.

6.6 Simulation results

The sampled results presented in this section were obtained with 5×10^5 samples and their state of convergence checked by plotting their time evolution. Figure 6.5 presents the sampled charge and plasma potential that were obtained for the entire domain, while Fig. 6.11 presents their values on the centerline. The plasma potential is observed to monotonically decrease between the two jets dropping from 5 V to 2 V. Because of the choice of axis scaling, it can be clearly seen on that plot that a Boltzmann-type relation:

$$\phi = \phi_0 + K \log\left(\frac{\rho}{\rho_0}\right) \tag{6.90}$$

does not describe well the relationship between the charge and the plasma potential between the two jets. The resultant electric field in the x-direction $E_x = -\frac{\partial \phi}{\partial x}$ is plotted in the top part of Fig. 6.7 which provides a positive, albeit non-uniform acceleration to particles in the x direction. The effects of that acceleration are visible in the bottom part of Fig. 6.7 which shows the increase of ion velocity in the x direction. As $E_y = -\frac{\partial \phi}{\partial y} > 0$, it drives ions away from the centerline which explains the shape of the charge number density isocontours in Fig. 6.5 which are much more spread out in the radial direction than those of neutral particles shown in Fig. 6.6. Two important intermediate quantities of the fluid model are plotted in Fig. 6.6, the neutral number density and the plasma conductivity. The neutral density corresponds to the sum of the number densities of species Ar,1 and Ar,2. The neutral particles demonstrate less spread in the radial direction compared to the ions as they are unaffected by the electric field. The electron streamline function Ψ is shown in Fig. 6.8 along with the electron velocity x-component which is obtained with $u_{e,x} = \frac{1}{n_e}\frac{\partial \Psi}{\partial x}$ following Eq. (6.45). Due to the formulation of the detailed fluid model, the electron velocity is different, and greater than that of the ions that is shown in Fig. 6.7 although both have the same number density from the quasi-neutrality assumption.

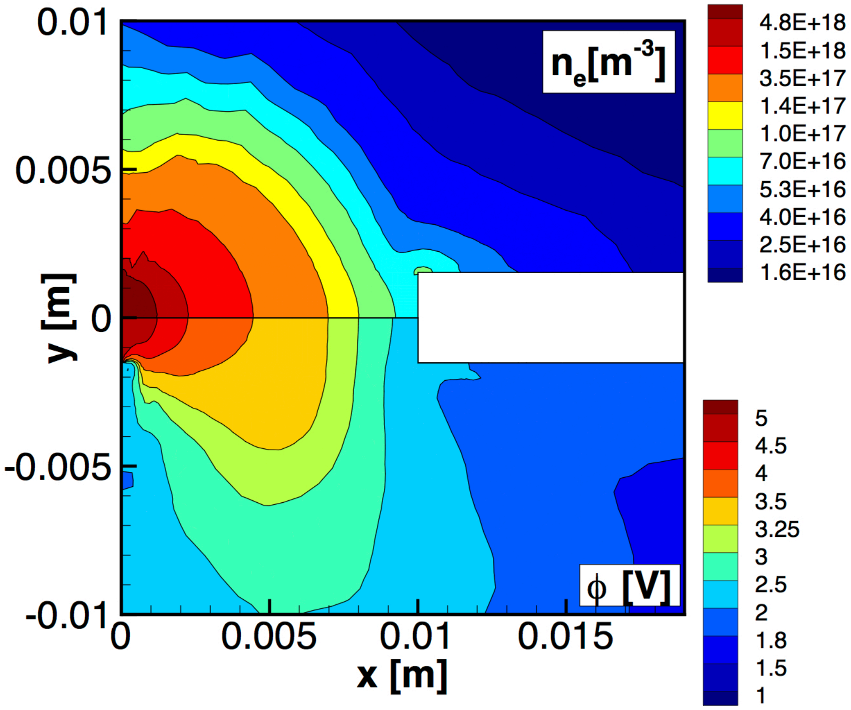


Figure 6.5: Sampled charge number density (top) and sampled plasma potential (bottom).

6.7 Conclusions

In this chapter, a thorough exposition of a hybrid DSMC/fluid electron model was presented. A detailed derivation of the electron fluid equations was given. It was followed by a discussion of the charge/force interpolation problem within a DSMC framework which is introduced by the need to both calculate a global, spatially dependent charge and to move charged particles following a spatially varying field. The implementation of the coupling between the fluid model and DSMC was finally presented. The DSMC/fluid model was then applied to the simulation of the flow between a plasma and a neutral jet. The particular flow that is studied in this Chapter was chosen because of its technological application and to allow the demonstration of the adaptive technique within the framework of a complex rarefied flow. Very

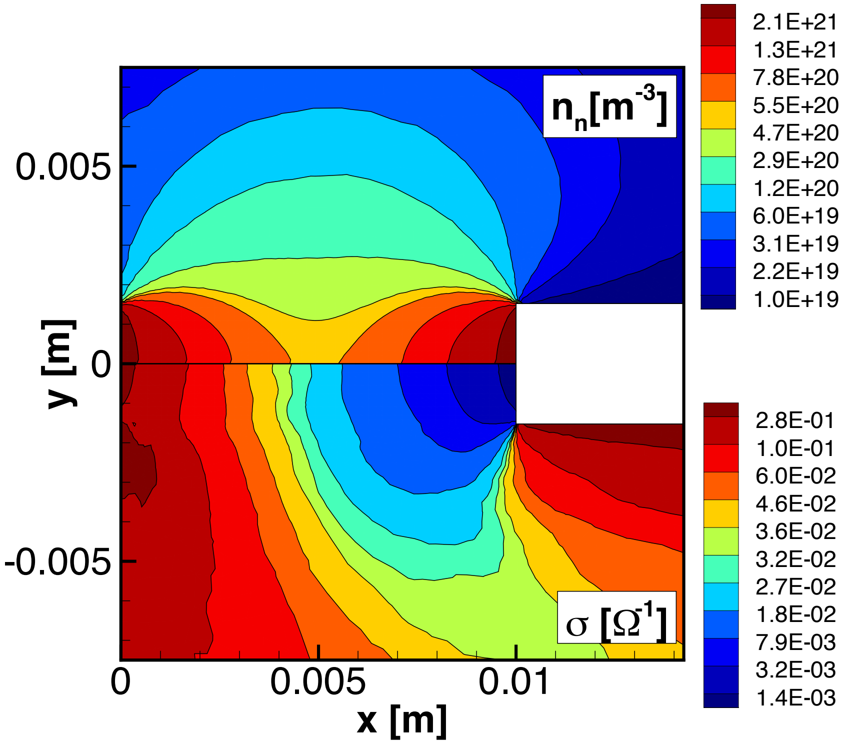


Figure 6.6: Sampled neutral number density (top) and plasma conductivity (bottom).

satisfactory results were obtained with the adaptive method within the framework of a *complex* flow which today cannot be simulated without a priori specifying relative weights, and cell weights. This further reinforces the case for the use of the adaptive procedure that was already made in Chapter V using a simpler rarefied flow. The results obtained in this chapter suggest that the hybrid electron fluid/DSMC framework presented in this Chapter benefits from the use of such a procedure. The latter should furthermore clearly be used if and when the energy equation is incorporated into the model.

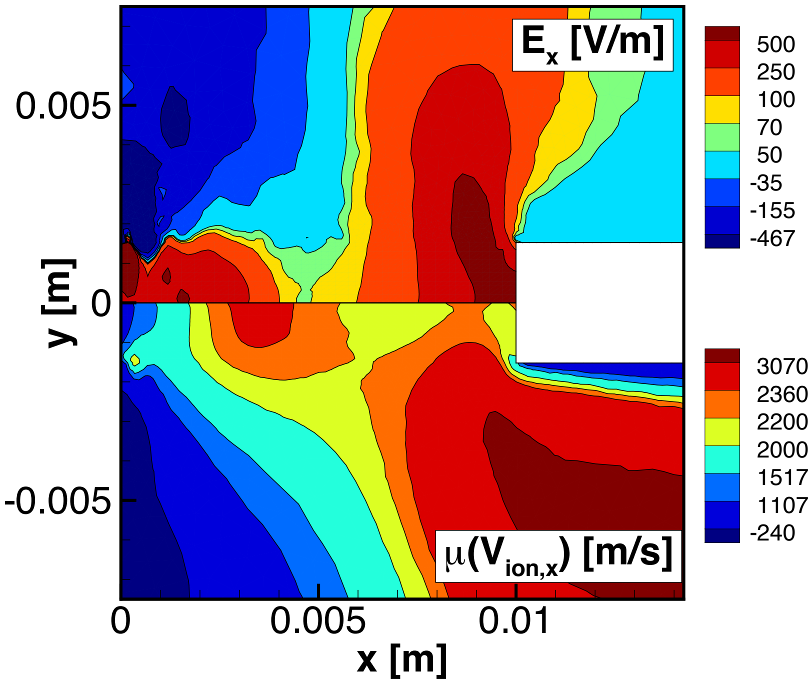


Figure 6.7: Sampled E_x field (top) and sampled V_x of Ar$^+$ ions (bottom).

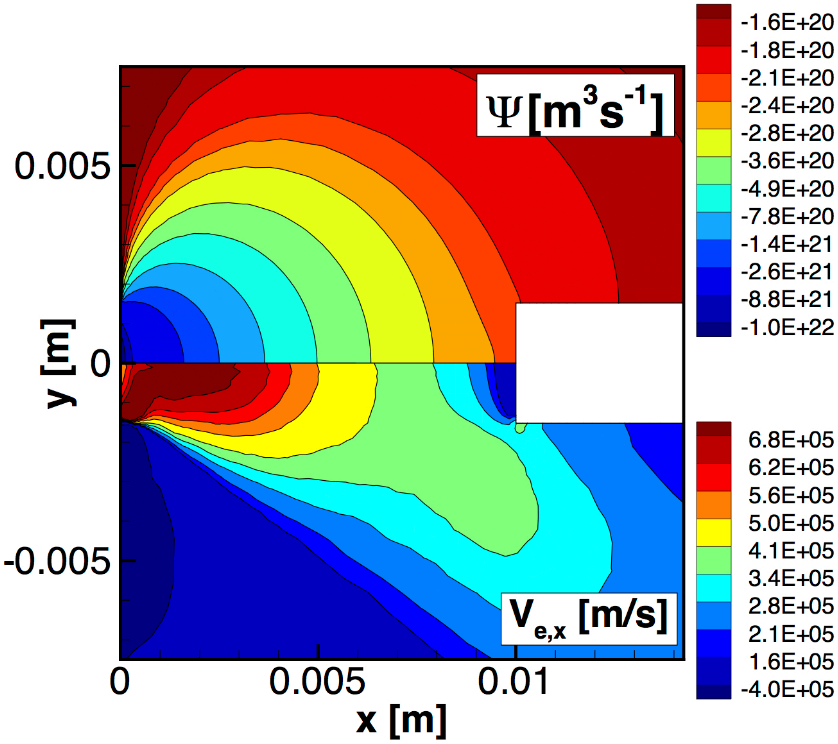


Figure 6.8: Electron streamline function Ψ (top) and electron x-velocity (bottom).

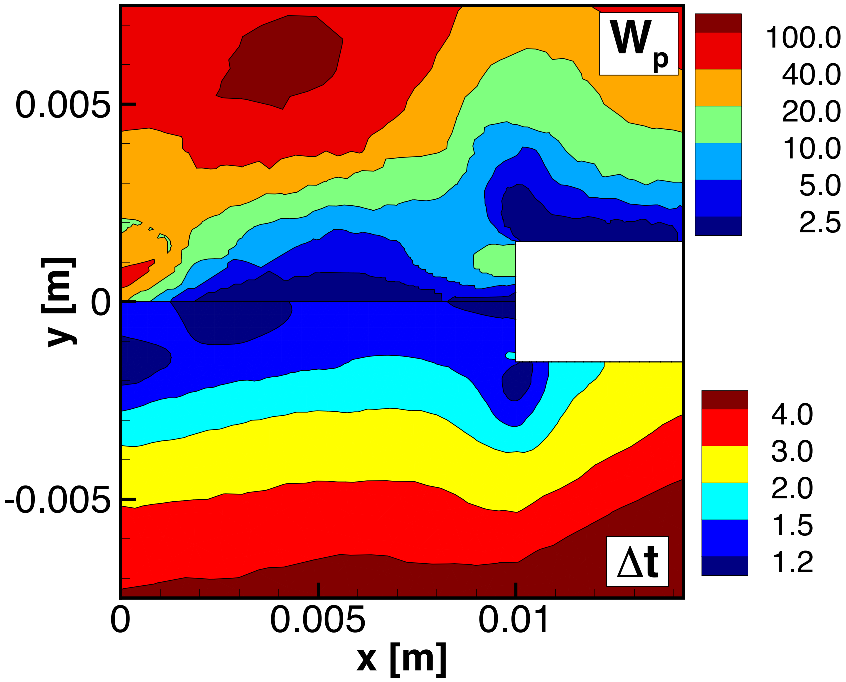


Figure 6.9: Weight (top) and time step (bottom) fields, as obtained with the adaptive procedure.

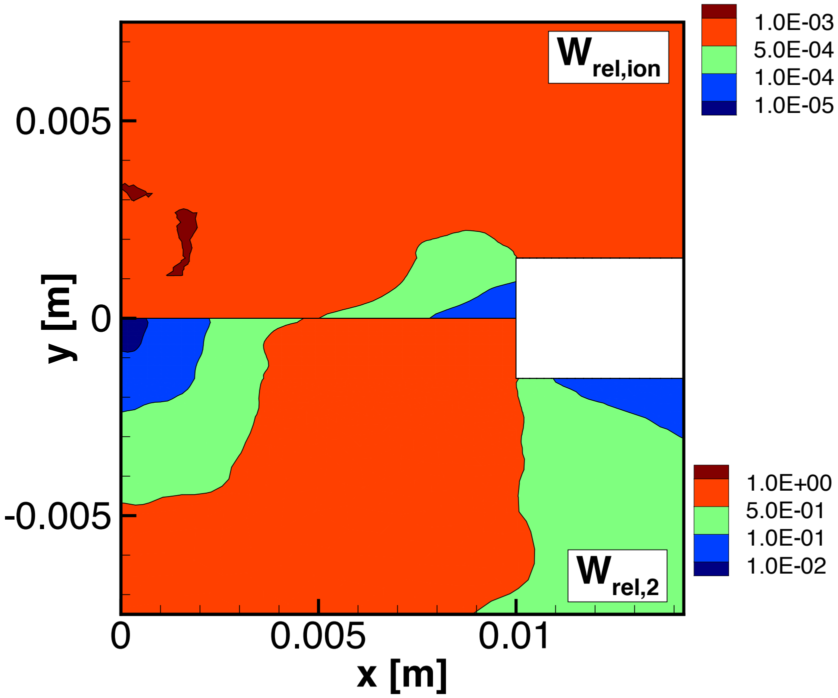


Figure 6.10: Relative weight of Ar^+ and species 2, as obtained with the adaptive procedure

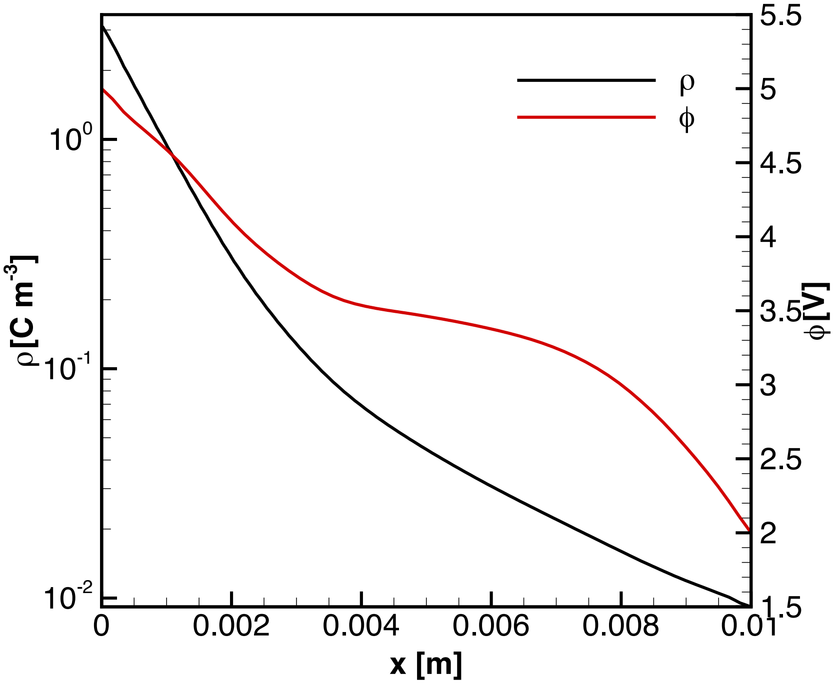


Figure 6.11: Sampled charge (log scale) and plasma potential on the centerline between the two jets.

CHAPTER VII

Conclusion

7.1 Summary

Chapter II introduced the Boltzmann equation by presenting its derivation from the Liouville equation. The formulation of the direct simulation Monte Carlo method was then introduced accompanied by a derivation of the collision step from a model homogeneous Boltzmann equation.

In Chapter III, A framework for the analysis of the convergence and accuracy of a DSMC simulation was presented and two corresponding error types, the *convergence* and *numerical* error introduced. The convergence error was studied by taking ensemble averages of sampled means during their convergence. It was found to vary as $\tilde{\sigma} k^{-1/2}$ for the sampled mean of the number density and velocity (k being the number of sampling steps). Time correlation between samples is such that the standard central limit theorem cannot be applied for sampled means. The autocorrelation function of sampled quantities was used to quantify the time correlation between samples. Particularly large time correlation was observed in zones where a small time step, relative to the mean collision time, was used. The autocorrelation function $\rho(k)$ was observed to be of very similar shape for N, V and very close to the correlation function between N and V . It was furthermore found to be invariant under particle count change and under $k\Delta t$ scaling.

A modified version of the central limit theorem that takes correlation into account for both particle and cell-based variables (e.g; respectively n and V) was introduced. It is then used to formulate a numerical method to a priori predict the value of the convergence error during the course of the simulation. It is found to accurately predict the observed standard deviation $\tilde{\sigma}$ for all cells of the domain that are examined.

In Chapter IV, the effect of the number of particles and the time step on the numerical error $\overline{\varepsilon}$ of DSMC was investigated within the framework of an axisymmetric jet that was previously used to study the convergence error in Chapter III. The cellwise numerical error was found to vary for both number density and velocity following a power law that is, as $C \times N^{-\alpha}$ with $0 < \alpha < 1.5$, where C and α are cell dependent and vary between n and V. The normalized cellwise numerical error of V is observed to be consistently an order of magnitude smaller than that of n. Large spatial variations are observed for the value of the numerical error throughout the domain, with low error zones having consistently smaller α values. The rather complicated spatial patterns of the error can be explained by the trajectory of particles in the simulation domain and the location of highly collisional zones. The average number of particles in a local cell gives no indication as to the value of the error in the cell although increasing it by raising the total number of particles in the simulation systematically reduces it. No direct correspondence is observed between the value of the cellwise normalized numerical error and the average number of particles in a particular cell. Accuracy for sampled quantities in a particular cell should not be tied to the average number of particles (e.g. 20) that it contains but rather to the total number of particles employed in the simulation, i.e. W_p. In practical terms, this means that the accuracy of a simulation should be gauged by comparing the results obtained for multiple particle counts. As was noted, advantage can be taken of the general power law scaling that was observed for the error to a priori determine the number of particles required for a desired level of accuracy. For relatively small particle counts, an increase in the time step results in an increase or decrease of the cellwise numerical error depending on the location in the flow. The effect of an increase in the time step is observed to result in a more monotonic increase of the value of the numerical error as the particle count is increased. No general scaling of the cellwise numerical error with Δt could be consistently observed although a power law type dependency is usually admissible for $\mu_1 \left(V \right)$ at relatively high particle counts.

In Chapter V, an adaptive method for the DSMC time step, cell weight and species relative weights was introduced. The aim of the procedure presented is to automatically update the time step, weight and relative weights of all cells, as the simulation progresses to steady state. The value of the time step is maximized while still requiring it to satisfy DSMC requirements for all cells. The value of the weights and relative weights are chosen so as to obtain a specified average number of particles for all species in all cells. The formulation of the adaptive method makes its implementation in existing DSMC codes straightforward while its computational overhead, compared to that of the simulation, is negligible. It furthermore reduces the need for user input by calculating the optimum time step and weight fields automatically thereby eliminating the need to run multiple simulations so as to determine them. The potential computational savings allowed by method were illustrated by simulating the flow between two counterflowing jets at a Knudsen number of 0.015 with the new method on one hand and with a spatially varying time step and cell weight field on the other. For the same total number of particles, the use of the adaptive method allows a substantial reduction, i.e. an order of magnitude, in the value of the error for the sampled number density and velocity. The increased computational efficiency is achieved by more uniformly distributing particles throughout the domain, thereby preventing cells from containing an excessive or an insufficient number of particles.

The adaptive method is thus expected to be of most benefit for multi-species flows with uneven distributions of the number of particles between cells and species.

In Chapter VI, a thorough exposition of a hybrid DSMC/fluid electron model was presented. It includes a detailed derivation of the electron fluid equations as well as the charge/force interpolation procedures The DSMC/fluid model, without the energy equation, and using the adaptive technique presented in Chapter V, was then applied to the simulation of the flow between a plasma and a neutral jet. Satisfactory results were obtained for the plasma potential and charge.

7.2 Contributions

To this author's best knowledge and review of the literature, the work contained in this thesis has permitted the following contributions:

1. Chapter III presents the first study of the correlation between samples in a DSMC simulation.

2. Chapter III contains one of the first use of the central limit theorem to study the convergence of sampled quantities in a DSMC simulation.

3. Chapter IV presents the first study of the cellwise numerical error in a spatially non-homogeneous DSMC simulation. In particular, it is the first work that suggests a power law behavior (with a spatially varying slope) for the cellwise numerical error in a DSMC simulation.

4. Chapter IV clearly indicates that the often quoted guideline of needing "20 particles per cell" for accurate DSMC simulation results, does not hold for a spatially inhomogeneous simulation.

5. Chapter IV highlights for the first time, the absence of any scaling for the cellwise numerical error with the time step contrary to what is observed for the

global error.

6. Chapter V contains the first published exposition of an adaptive procedure for the DSMC cell weights and/or relative weights.
7. Chapters V and VI detail the first published use of spatially varying relative weights to maximize DSMC simulation efficiency.

7.3 Future Work

Convergence Study

The a priori convergence determination method in Chapter III should be incorporated into an automated procedure that halts the simulation whenever a specified level of convergence has been reached which implementation should be rather straightforward. It is also this author's opinion that the time series provided by instantaneous cell quantities, e.g. the instantaneous number of particles in a cell as a function of time, is an area that should receive greater attention in the spirit of [92]. It is very likely that some properties of those time series might provide a way to determine a priori the accuracy of the simulation in a particular cell.

Accuracy Study

More detailed accuracy studies of DSMC such as was performed here should be undertaken to gain a greater understanding of the accuracy of method. Not having any a priori method to quantify the value of the numerical error may ultimately make DSMC less viable than exact methods [95] as the availability of computational resources increases with time. Future work concerning the numerical error may include performing the detailed analysis conducted in the present work for other flows, such as two dimensional jets, to see whether similar trends or error patterns are observed with the ultimate goal of being able to a priori predict them. Alongside this proposed “numerical experiment” route which is necessarily limited by the specificities of each particular test case, an alternative path might be a more theoretical study based on the interpretation of DSMC as a Markov jump process. Any possible insights thus obtained might first be checked in a homogeneous DSMC simulation with only one cell and then extrapolated to one or two dimensions.

In parallel to that, the accuracy of the particle move procedure and the possible effect of boundary conditions could be investigated by repeating similar numerical experiments without collisions. It might also be interesting to study the propagation of the error by, for instance, perturbing the flow, thereby introducing error in selected cells and investigating their effect on the rest of the flow.

It would also be interesting to repeat the numerical error analysis for a two-dimensional jet to isolate the contribution of the axisymmetric geometry on the observed error patterns.

Adaptive Procedure

The effect of spatially varying weights and time step on accuracy should clearly be investigated using the error analysis framework introduced in Chapter IV. This would help quantify and maybe explain what was remarked when spatially varying weights are present, i.e. that having $\overline{N}_0$ particles on average when the simulation contains spatially varying weights is not identical to having the same number in their absence. The performance of the adaptive procedure is limited by the deleterious effects of particle cloning at weight interface which are essentially caused by the creation of multiple identical particles. One strategy, proposed in [15] to specifically address this issue is to store cloned particles at weight interfaces so as to then reuse them in subsequent time steps. Implementing such a scheme could allow the use of even fewer particles when using the adaptive technique.

Hybrid Fluid/DSMC technique

Having proven the basic viability of the hybrid scheme for this test case and flow conditions for a constant electron temperature, the remaining energy equation should next be incorporated into the model. A natural next step would the solution of the complete set of fluid equations without any simplifying assumptions (i.e. potential flow, no time derivatives, no convective terms) using a finite volume solver. This would allow a true time-dependent coupling between the electron fluid and the particles which is really necessary due to the inherent unsteadiness of all cell averaged quantities such as the charge (i.e. the ion number density under the quasineutrality assumption) in a DSMC simulation.

APPENDIX

APPENDIX A

Markov Processes

In this appendix, rigorous definitions of a few concepts related to Markov processes that are used in Chapter II are given. A much more thorough presentation can be found, amongst many others, in [97] [98] [54].

Definition A.1. Considering a probability space $(\Omega, \mathcal{A}, P)$, with sample space Ω, σ-algebra $\mathcal{A}$ and probability measure P, a *continuous time stochastic process* is a family of random variables $\{Z(t); t \geq 0\}$ with values in a measurable *state space* $(S, \mathcal{S})$, i.e. $Z(t) : \Omega \to S$.

$Z(t)$ designates the *state* of the system at time t, while for a given outcome $\omega \in \Omega$, $\{t \geq 0\}$, is called a *trajectory* or *realization* of the stochastic process Z associated with ω. The state space $(S, \mathcal{S})$ comprises both the space in which Z takes its values S and an associated σ-algebra $\mathcal{S}$. In this thesis, only $S = \mathbb{R}^d$ with $d = N$ is considered, as it is used to describe the velocity of all N particles in a cell. In that case, $\mathcal{S}$ is the Borel σ-algebra for $\mathbb{R}^d$. By abuse of language in Chapter II, S is, however, referred to as state space.

Definition A.2. A continuous time stochastic process process $\{Z(t); t \geq 0\}$ with values in state space $(S, \mathcal{S})$ is said to have the Markov property (and is thereby referred to as a *Markov process*) if and only if for all $0 < s < t$ and for

$$t_0 = 0 < t_1 < ... < t_n < s \qquad \text{and} \qquad z_0, z_1, ..., z_n, z \in S,$$

$$P\left[Z\left(t\right) \in B \mid Z\left(t_s\right) = z, Z\left(t_n\right) = z_n, ..., Z\left(t_1\right) = z_1, Z\left(0\right) = z_0\right] = \\ P\left[Z\left(t\right) \in B | Z\left(t_s\right) = z\right] \qquad \forall B \in \mathcal{S} \quad \text{(A.1)}$$

Only *homogeneous* Markov processes are considered, i.e. processes such that $P\left[Z\left(t\right) \in B | Z\left(t_s\right) = z\right]$ only depends on t and s through their difference $t - s$.

Definition A.3. A *Markov process with jumps* is a Markov process $\left\{Z\left(t\right), t \geq 0\right\}$ for which trajectories are constant and right continuous between jumps which occur at random jump times $T^1\left(\omega\right) \leq T^2\left(\omega\right) \leq ... \leq T^k\left(\omega\right) \leq ....$ It can be expressed as:

$$Z\left(t\right) = \sum_{k \geq 0; T_k < \infty} \overline{z}^k \mathbf{1}_{\left[T^k, T^{k+1}\right[}\left(t\right) \tag{A.2}$$

where $\overline{z}^k \in S$ is the state of the system for $t^k \leq t < t^{k+1}$, and where the indicator function $\mathbf{1}_E\left(\cdot\right)$ is defined as

$$\mathbf{1}_E(t) := \begin{cases} 1 & \text{if } t \in E \\ 0 & \text{if } t \notin E \end{cases}. \tag{A.3}$$

Definition A.4. A Markov *transition function* (or transition kernel) is a function $\mu : S \times \mathcal{S} \to \mathbb{R}^+$ such that *i)* $\forall B \in \mathcal{S}$, $\mu\left(x, B\right)$ is $\mathcal{S}$-measurable, *ii)* For any $x \in S$, $\mu\left(x, y\right)$ is a probability measure on $\left(S, \mathcal{S}\right)$ such that $\mathrm{P}\left(Z^{k+1} \in B | Z^k = x\right) = \mu\left(x, B\right)$, $\forall B \in \mathcal{S}$ and $\forall x \in S$.

An integral of a function $f : S \to \mathbb{R}$ with respect to measure μ can be written as $\int_S f(y)\,\mu(x, dy)$ which is a notation frequently utilized in chapter II. It, for instance, allows the concise writing of expressions such as

$$\mathbb{E}\left[f\left(Z^{k+1}\right)|Z^k = x\right] = \int_S \mu(x, dy)\, f(y)\,. \tag{A.4}$$

Proposition A.5. Conditionally, at each step $k \neq 0$, $\overline{z}^{k+1}$ and T^{k+1} are independent. By definition A.4, given $\overline{z}^k$, $\overline{z}^{k+1} \sim \mu\left(\overline{z}^k, \cdot\right)$. The waiting time at time step k, $\Delta T^k = T^{k+1} - T^k$ follows an exponential distribution with parameter $\lambda\left(\overline{z}^k\right)$, i.e. $\Delta T^k \sim \text{Exp}\left[\lambda\left(\overline{z}^k\right)\right]$.

Proof. A Markov jump process is continuously memoryless between jumps. From the definition of Eq. (A.2) this means that for all $0 < s < t$

$$P(\Delta T > t + s|\Delta T > s) = P(\Delta T > t) \tag{A.5}$$

$$\begin{aligned} P(\Delta T > t + s) = P(\Delta T > t + s \cap \Delta T > s) &= P(\Delta T > t + s|\Delta T > s)\, P(\Delta T > s) \\ &= P(\Delta T > s)\, P(\Delta T > t) \end{aligned}$$

from Eq. (A.5). Defining $P(\Delta T > \zeta) = \int_\zeta^\infty f_{\Delta T}(\eta)\, d\eta$ where $f_{\Delta T}$ is the distribution function of Δt

$$P(\Delta T > t + s) = P(\Delta T > s)\, P(\Delta T > t) \Rightarrow f_{\Delta T}(t + s) = f_{\Delta T}(t) + f_{\Delta T}(s) \tag{A.6}$$

which can only be satisfied if $f_{\Delta T}(\eta) = C \exp(\eta)$ with $C \in \mathbb{R}$, which means that $f_{\Delta T}$ can only be the exponential distribution. □

Definition A.6. For all homogeneous Markov Processes $\{Z(t), t \geq 0\}$, we can define a *transition semi-group*, $\{Q_t, t \geq 0\}$, $Q_t : S \times \mathcal{S} \to [0, 1]$, such that $\forall z \in S$ and $\forall B \in \mathcal{S}$

$$Q_t(z, B) = P\left(Z\left(t\right) \in B | Z\left(0\right) = z\right) \tag{A.7}$$

$Q_t(z, B)$ is clearly a probability measure for $\forall z \in S$

A semi-group Q_t defines a linear operator Q_t such that

$$Q_t f\left(z\right) = \mathbb{E}\left[f\left(Z\left(t\right)\right) | Z\left(0\right) = z\right] \tag{A.8}$$

for all functions $f : S \to \mathbb{R}$, which, following the notation of Eq. (A.4), is usually written

$$Q_t f\left(z\right) = \int\limits_S f\left(z'\right) Q_t\left(z; dz'\right) \tag{A.9}$$

Definition A.7. The *infinitesimal generator* A of semi-group Q (and of the associated Markov process), is the operator that is the derivative at the origin of Q_t, that is

$$Af\left(z\right) = \lim_{h \to 0} \frac{1}{h} \left[Q_h f\left(z\right) - Q_0 f\left(z\right)\right] = \lim_{h \to 0} \frac{1}{h} \left(\mathbb{E}\left[f\left(Z\left(h\right)\right) | Z\left(0\right) = z\right] - f\left(z\right)\right). \tag{A.10}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] R.K. Agarwal, K. Yun, and R. Balakrishnan. Beyond Navier-Stokes: Burnett equations for flows in the continuum-transition regime. *Physics of Fluids*, 13(10):3061–3085, 2001.

[2] B.J. Alder and T.E. Wainwright. Phase transition for a hard sphere system. *The Journal of Chemical Physics*, 27(5):1208–1209, 1957.

[3] A.A. Alexeenko, C. Galitzine, and A.M. Alekseenko. High-order discontinuous Galerkin method for Boltzmann model equations, AIAA Paper 2008-4256, 2008.

[4] A.A. Alexeenko, A. Ganguly, and S.L. Nail. Computational analysis of fluid dynamics in pharmaceutical freeze-drying. *Journal of Pharmaceutical Sciences*, 98(9):3483–3494, 2009.

[5] A.A. Alexeenko, S.F. Gimelshein, D.A. Levin, A.D. Ketsdever, and M.S. Ivanov. Measurements and simulation of orifice flow for micropropulsion testing. *Journal of Propulsion and Power*, 19(4):588–594, 2003.

[6] V.V. Aristov. *Direct Methods for Solving the Bolzmann Equation and Study of Nonequilibrium Flows.* Kluwer Academic Publishers, 2001.

[7] A.A. Arsen'yev. On the approximation of the solution of the Boltzmann equation by solutions of the Itō stochastic differential equations. *USSR Computational Mathematics and Mathematical Physics*, 27(2):51–59, 1987.

[8] C. Baranger, J. Claudel, N. Hérouard, and L. Mieussens. Locally refined discrete velocity grids for stationary rarefied flow simulations. *Journal of Computational Physics*, 257, Part A:572–593, 2014.

[9] M. Bartel, J.E. Johannes, and T.R. Furlani. Trace chemistry modelling with DSMC in chemically reacting plasmas, AIAA Paper 98-2753, 1998.

[10] O.M. Belotserkovskii and V.E. Yanitskii. The statistical particles-in-cells method for solving rarefied gas dynamics problems. *USSR Computational Mathematics and Mathematical Physics*, 15(5):101–114, 1975.

[11] J.J. Bertin and R.M. Cummings. Critical hypersonic aerothermodynamic phenomena. *Annual Review of Fluid Mechanics*, 38(1):129–157, 2006.

[12] G.A. Bird. Approach to translational equilibrium in a rigid sphere gas. *Physics of Fluids*, 6(10):1518–1519, 1963.

[13] G.A. Bird. *Molecular Gas Dynamics*. Clarendon Press, 1976.

[14] G.A. Bird. Perception of numerical methods in rarefied gasdynamics. *Progress in Astronautics and Aeronautics, AIAA*, 117:211–226, 1989.

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

[16] G.A. Bird. Recent advances and current challenges for DSMC. *Computers and Mathematics with Applications*, 35(1):1–14, 1998.

[17] G.A. Bird. Forty years of DSMC, and now? In *Proceedings of the 22nd International Symposium on Rarefied Gas Dynamics, AIP, 2001*, pages 372–380, 2001.

[18] G.A. Bird. *The DSMC Method*. Create Space Independent Publishing Platform, 2013.

[19] G.A. Bird, M.A. Gallis, J.R. Torczynski, and D.J. Rader. Accuracy and efficiency of the sophisticated direct simulation Monte Carlo algorithm for simulating noncontinuum gas flows. *Physics of Fluids*, 21(1):017103, 2009.

[20] C.K. Birdsall. Particle-in-Cell charged particle simulations, plus Monte Carlo collisions with neutral atoms, PIC-MCC. *IEEE Transactions on Plasma Science*, 19(2):65–85, 1991.

[21] C.K. Birdsall and A.B. Langdon. *Plasma Physics via Computer Simulation*. Taylor & Francis, 2004.

[22] L. Boltzmann. Weitere Studien über das Wärmegleichgewicht unter Gasmolekülen. *Sitzungsberichte Akademie der Wissenschaften*, (66):275–370, 1872.

[23] I.D. Boyd. Conservative species weighting scheme for the direct simulation Monte Carlo method. *Journal of Thermophysics and Heat Transfer*, 10(4), 1996.

[24] I.D. Boyd. Monte Carlo simulation of nonequilibrium flow in a low-power hydrogen arcjet. *Physics of Fluids*, 9(10):3086, 1997.

[25] I.D. Boyd. Numerical modeling of spacecraft electric propulsion thrusters. *Progress in Aerospace Sciences*, 41(8):669–687, 2005.

[26] I.D. Boyd. Modeling of associative ionization reactions in hypersonic rarefied flows. *Physics of Fluids*, 19(9), 2007.

[27] I.D. Boyd. Direct simulation Monte Carlo for atmospheric entry. 2. Code development and application results. NATO Conf. Proc. RTO-EN-AVT-162, 2009.

[28] I.D. Boyd and M.W. Crofton. Modeling the plasma plume of a hollow cathode. *Journal of Applied Physics*, 95(7):3285–3296, 2004.

[29] I.D. Boyd and J.T. Yim. Modeling of the near field plume of a Hall thruster. *Journal of Applied Physics*, 95(9):4575–4584, 2004.

[30] S. Brenner and R.J. Scott. *The Mathematical Theory of Finite Element Methods*. Springer, 2008.

[31] S.G. Brush. *The Kinetic Theory of Gases: An Anthology of Classic Papers with Historical Commentary*. Imperial College Press, 2003.

[32] D. Bushnell, M. Garneau, J.M. Logsdon, R. Sagdeev, E. Lu, M. Mountain, and N. Stephenson. NASA: what now? *Nature*, 472(7341):27–29, 2011.

[33] C. Cai. *Theoretical and Numerical Studies of Plume Flows in Vacuum Chambers*. PhD thesis, University of Michigan, 2005.

[34] R. Campargue. Progress in overexpanded supersonic jets and skimmed molecular beams in free-jet zones of silence. *The Journal of Physical Chemistry*, 88(20):4466–4474, 1984.

[35] S. Caprino, M. Pulvirenti, and W. Wagner. Stationary particle systems approximating stationary solutions to the Boltzmann equation. *SIAM Journal on Mathematical Analysis*, 29(4):913–934, 1998.

[36] G. Casella and R.L. Berger. *Statistical Inference*. Duxbury, 2001.

[37] A. Cedilnik, K. Kosmelj, and A. Blejec. The distribution of the ratio of jointly normal variables. *Metodoloski zvezki*, 1(1):99–108, 2004.

[38] C. Cercignani. *The Boltzmann Equation and its Applications*. Springer, 1988.

[39] C. Cercignani, R. Illner, and M. Pulvirenti. *The Mathematical Theory of Dilute Gases*. Springer, 1994.

[40] S. Chapman and T.G. Cowling. *The Mathematical Theory of Non-uniform Gases*. Cambridge University Press, 1991.

[41] G. Chen and I.D. Boyd. Statistical error analysis for the direct simulation Monte Carlo technique. *Journal of Computational Physics*, 126(2):434–448, 1996.

[42] H. Chen, S. Kandasamy, S. Orszag, R. Shock, S. Succi, and V. Yakhot. Extended Boltzmann kinetic equation for turbulent flows. *Science*, 301(5633):633–636, 2003.

[43] G. Chernyi, S. Losev, S. Macheret, and B. Potapkin. *Physical and Chemical Processes in Gas Dynamics: Cross Sections and Rate Constants*. AIAA, 2002.

[44] Y. Choi. *Modeling an Anode Layer Hall Thruster and its Plume*. PhD thesis, University of Michigan, 2008.

[45] Y. Choi, M. Keidar, and I.D. Boyd. Particle simulation of plume flows from an anode-layer Hall thruster. *Journal of Propulsion and Power*, 24(3):554–561, 2008.

[46] M.R. Combi. Time-dependent gas kinetics in tenuous planetary atmospheres: The cometary coma. *Icarus*, 123(1):207–226, 1996.

[47] G.E. Cook. Satellite drag coefficients. *Planetary and Space Science*, 13(10):929–946, 1965.

[48] R. Dautray. *Méthodes Probabilistes pour les Équations de la Physique*. Eyrolles, 1989.

[49] J.-L. Delcroix. *Plasma Physics*. Wiley, 1965.

[50] 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.

[51] R.J. DiPerna and P.-L. Lions. Global weak solutions of kinetic equations. *Rendiconti del Seminario Matematico Università e Politecnico di Torino*, 46(3):259–288, 1988.

[52] D.J. Economou, T.J. Bartel, R.S. Wise, and D.P. Lymberopoulos. Two-dimensional direct simulation monte carlo (DSMC) of reactive neutral and ion flow in a high density plasma reactor. *IEEE Transactions on Plasma Science*, 23(4):581–590, 1995.

[53] K.T. Edquist, P.N. Desai, and M. Schoenenberger. Aerodynamics for Mars Phoenix entry capsule. *Journal of Spacecraft and Rockets*, 48:713–726, 2011.

[54] S.N. Ethier and T.G. Kurtz. *Markov Processes: Characterization and Convergence*. Wiley, 2005.

[55] M.A. Fallavollita, D. Baganoff, and J.D. McDonald. Reduction of simulation cost and error for particle simulations of rarefied flows. *Journal of Computational Physics*, 109(1):30–36, 1993.

[56] E.D. Farbar. *Kinetic Simulation of Rarefied and Weakly Ionized Hypersonic Flow Fields*. PhD thesis, University of Michigan, 2010.

[57] E.D. Farbar and I.D. Boyd. Modeling of the plasma generated in a rarefied hypersonic shock layer. *Physics of Fluids*, 10(22):106101, 2010.

[58] T.S. Ferguson. *A Course in Large Sample Theory*. Chapman and Hall, 2004.

[59] H. Flyvbjerg and H.G. Petersen. Error estimates on averages of correlated data. *Journal of Chemical Physics*, 91(1):461, 1989.

[60] C. Galitzine and I.D. Boyd. Simulation of the interaction between two counterflowing rarefied jets. In *Proceedings of the 26th International Symposium on Rarefied Gas Dynamics, AIP, 2011*, pages 545–550, 2011.

[61] C. Galitzine and I.D. Boyd. Development of an adaptive weighting scheme for DSMC and its application to an axisymmetric jet. In *Proceedings of the 27th International Symposium on Rarefied Gas Dynamics, AIP, 2012*, pages 587–594, 2012.

[62] M.A. Gallis, J.R. Torczynski, D.J. Rader, and G.A. Bird. Convergence behavior of a new DSMC algorithm. *Journal of Computational Physics*, 228(12):4532–4548, 2009.

[63] D. Gao, C. Zhang, and T.E. Schwartzentruber. Particle simulations of planetary probe flows employing automated mesh refinement. *Journal of Spacecraft and Rockets*, 48(3):397–405, 2011.

[64] A.L. Garcia. Direct simulation Monte Carlo: Theory, methods, and open challenges, Proceedings of the Models and Computational Methods for Rarefied Flows NATO Conf. Proc. RTO-EN-AVT-194 , 2011.

[65] A.L. Garcia and W. Wagner. Time step truncation error in direct simulation Monte Carlo. *Physics of Fluids*, 12(10):2621, 2000.

[66] N.A. Gatsonis and A. Spirkin. A three-dimensional electrostatic particle-in-cell methodology on unstructured Delaunay-Voronoi grids. *Journal of Computational Physics*, 228(10):3742–3761, 2009.

[67] N.A. Gatsonis and X. Yin. Hybrid (particle-fluid) modeling of pulsed plasma thruster plumes. *Journal of Propulsion and Power*, 17(5):945–958, 2001.

[68] C. Gleason-Gonzalez, S. Varoutis, V. Hauer, and C. Day. Simulation of neutral gas flow in a tokamak divertor using the direct simulation Monte Carlo method. *Fusion Engineering and Design*, In Press, 2014.

[69] D.M. Goebel and I. Katz. *Fundamentals of Electric Propulsion: Ion and Hall Thrusters*. Wiley, 2007.

[70] V.E. Golant, A.P. Zhilinsky, and I.E. Sakharov. *Fundamentals of Plasma Physics*. Wiley, 1980.

[71] T.I. Gombosi. *Gaskinetic Theory*. Cambridge University Press, 1994.

[72] H. Grad. The profile of a steady plane shock wave. *Communications on Pure and Applied Mathematics*, 5(3):257–300, 1952.

[73] C. Graham and S. Méléard. Probabilistic tools and Monte-Carlo approximations for some Boltzmann equations. *ESAIM: Proc.*, 10:77–126, 2001.

[74] Y.N. Grigoryev, V.A. Vshivkov, and Fedoruk M.P. *Numerical Particle-in-Cell Methods: Theory and Applications.* de Gruyter, 2002.

[75] X.J. Gu and D.R. Emerson. A computational strategy for the regularized 13 moment equations with enhanced wall-boundary conditions. *Journal of Computational Physics*, 225(1):263–283, 2007.

[76] J.T. Gudmundsson. Notes on the electron excitation rate coefficients for argon and oxygen discharge. Technical Report RH-21-2002, Science Institute, University of Iceland, 2002.

[77] N.G. Hadjiconstantinou, A.L. Garcia, M.Z. Bazant, and G. He. Statistical error in particle simulations of hydrodynamic phenomena. *Journal of Computational Physics*, 187(1):274–297, 2003.

[78] R.W. Hockney and J.W Eastwood. *Computer Simulation Using Particles.* Taylor & Francis, 1989.

[79] K. Huang. *Statistical Mechanics.* Wiley, 1987.

[80] J.D. Huba. *NRL Plasma Formulary.* Naval Research Laboratory, 2013.

[81] T.D. Huismann. *Improving Hall Thruster Plume Simulation through Refined Characterization of Near-field Plasma Properties.* PhD thesis, University of Michigan, 2011.

[82] M. Ivanov, G. Markelov, and S. Gimelshein. Statistical simulation of reactive rarefied flows - Numerical approach and applications, AIAA Paper 1998-2669, 1998.

[83] M. Ivanov and S. Rogasinsky. Analysis of numerical techniques of the direct simulation Monte Carlo method in the rarefied gas dynamics. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 3(6):453–466, 1988.

[84] M.S. Ivanov and S.F. Gimelshein. Computational hypersonic rarefied flows. *Annual Review of Fluid Mechanics*, 30(1):469–505, 1998.

[85] G.B. Jacobs and J.S. Hesthaven. High-order nodal discontinuous Galerkin particle-in-cell method on unstructured grids. *Journal of Computational Physics*, 214(1):96–121, 2006.

[86] A. Jameson. Re-engineering the design process through computation. *Journal of Aircraft*, (1):36–50.

[87] J. Jeans. *An Introduction to the Kinetic Theory of Gase.* Cambridge University Press, 1982.

[88] E. Josyula and J. Burt. Review of rarefied gas effects in hypersonic applications. NATO Conf. Proc. RTO-EN-AVT-194, 2011.

[89] M. Kac. Foundations of Kinetic Theory. In J. Neyman, editor, *Third Berkeley Symposium on Mathematical Statistics and Probability*, pages 171–197, 1956.

[90] K. Kadau, J.L. Barber, T.C. Germann, B.L. Holian, and B.J. Alder. Atomistic methods in fluid simulation. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 368(1916):1547–60, 2010.

[91] K.C. Kannenberg and I.D. Boyd. Strategies for efficient particle resolution in the direct simulation Monte Carlo method. *Journal of Computational Physics*, 157(2):727–745, 2000.

[92] C.R. Kaplan and E.S. Oran. Nonlinear filtering for low-velocity gaseous microflows. *AIAA Journal*, 40:82–90, 2002.

[93] G. Karniadakis, A. Beskok, and N. Aluru. *Microflows and Nanoflows: Fundamentals and Simulation.* Springer, 2005.

[94] I. Katz, I. Mikellides, and D.M. Goebel. Model of the plasma potential distribution in the plume of a hollow cathode, AIAA Paper 2004-4108, 2004.

[95] V.I. Kolobov, R.R. Arslanbekov, V.V. Aristov, A.A. Frolova, and S.A. Zabelok. Unified solver for rarefied and continuum flows with adaptive mesh and algorithm refinement. *Journal of Computational Physics*, 223(2):589–608, 2007.

[96] R. Kumar, Z. Li, and D.A. Levin. Modeling of CO_2 condensation in the high pressure flows using the statistical BGK approach, AIAA Paper 2010-818, 2010.

[97] B. Lapeyre. *Markov Processes and Applications: Algorithms, Networks, Genome and Finance.* Wiley, 2009.

[98] B. Lapeyre, E. Pardoux, and R. Sentis. *Introduction to Monte-Carlo Methods for Transport and Diffusion Equations.* Oxford University Press, 2003.

[99] M. Laux. Local time stepping with automatic adaptation for the DSMC method, AIAA Paper 1998-818, 1998.

[100] E.L. Lehmann. *Elements of Large-Sample Theory.* Springer, 2004.

[101] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2002.

[102] D.W. Levy, K.R. Laflin, E.N. Tinoco, J.C. Vassberg, M. Mani, B. Rider, C.L. Rumsey, R.A. Wahls, J.H. Morrison, O.P. Brodersen, S. Crippa, D.J. Mavriplis, and M. Murayama. Summary of data from the fifth computational fluid dynamics drag prediction workshop. *Journal of Aircraft*, In Press, 2014.

[103] R.L. Liboff. *Kinetic Theory Classical, Quantum, and Relativistic Descriptions.* Springer, 2003.

[104] M.A Lieberman and A.J. Lichtenberg. *Principles of Plasma Discharges and Materials Processing; 2nd ed.* Wiley, 2005.

[105] A.S. Lipatov. *The Hybrid Multiscale Simulation Technology: An Introduction with Application to Astrophysical and Laboratory Plasmas.* Springer, 2010.

[106] M.N. Macrossan and C.R. Lilley. Viscosity of argon at temperatures greater than 2000 K from measured shock thickness. *Physics of Fluids*, 15(11):3452, 2003.

[107] H. P. McKean. Fluctuations in the kinetic theory of gases. *Communications on Pure and Applied Mathematics*, 28(4):435–455, 1975.

[108] M. Mitchner and C.H. Krueger. *Partially Ionized Gases.* Wiley, 1973.

[109] R. Monaco and L. Preziosi. *Fluid dynamic applications of the discrete Boltzmann equation.* World Scientific, 1991.

[110] E.P. Muntz. Rarefied gas dynamics. *Annual Review of Fluid Mechanics*, 21(1):387–422, 1989.

[111] K. Nanbu. Direct simulation scheme derived from the Boltzmann equation. i. Monocomponent gases. *Journal of the Physical Society of Japan*, 49(5):2042–2049, 1980.

[112] D.Y. Oh. *Computational Modeling of Expanding Plasma Plumes in Space using a PIC-DSMC Algorithm.* PhD thesis, Massachusetts Institute of Technology, 1997.

[113] D.Y. Oh and D.E. Hastings. Experimental verification of a PIC-DSMC model for Hall thrusters plumes AIAA Paper 96-3196, 1996.

[114] T. Ohwada. Higher order approximation methods for the Boltzmann equation. *Journal of Computational Physics*, 139(1):1–14, 1998.

[115] D.P. Olynick, H.A. Hassan, and J.N. Moss. Grid generation and adaptation for the direct simulation Monte Carlo method. *Journal of Thermophysics and Heat Transfer*, 3(4):368–373, 1989.

[116] E.S. Oran, C.K. Oh, and B.Z. Cybyk. Direct simulation Monte Carlo: Recent advances and applications. *Annual Review of Fluid Mechanics*, 30(1):403–441, 1998.

[117] A.V. Phelps. Cross sections and swarm coefficients for nitrogen ions and neutrals in N_2 and argon ions and neutrals in Ar for energies from 0.1 eV to 10 keV. *Journal of Physical and Chemical Reference Data*, 20:557–573, 1991.

[118] S.D. Piersall and J.B. Anderson. Direct Monte Carlo simulation of bimolecular reactions of chemical reaction systems: Simple bimolecular reactions. *Journal of Chemical Physics*, 95(2):971–978, 1991.

[119] M.Y. Plotnikov and E.V. Shkarupa. Estimation of the statistical error of the direct simulation Monte Carlo method. *Computational Mathematics and Mathematical Physics*, 50(2):335–344, 2010.

[120] M.Y. Plotnikov and E.V. Shkarupa. Selection of sampling numerical parameters for the DSMC method. *Computers and Fluids*, 58:102–111, 2012.

[121] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. R.T. Edwards, 2005.

[122] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Springer, 2000.

[123] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*. Springer, 1994.

[124] D.J. Rader, M.A. Gallis, J.R. Torczynski, and W. Wagner. Direct simulation Monte Carlo convergence behavior of the hard-sphere-gas thermal conductivity for Fourier heat flow. *Physics of Fluids*, 18(7):077102, 2006.

[125] S. Rjasanow and W. Wagner. A temperature time counter scheme for the Boltzmann equation. *SIAM Journal on Numerical Analysis*, 37(6):1800–1819, 2000.

[126] S. Rjasanow and W. Wagner. *Stochastic Numerics for the Boltzmann Equation*. Springer, 2005.

[127] W.M. Ruyten. Density-conserving shape factors for particle simulations in cylindrical and spherical coordinates. *Journal of Computational Physics*, 105(2):224–232, 1993.

[128] D.G. Samaras. *Theory of Ion Flow Dynamics*. Prentice-Hall, 1962.

[129] H. Schlichting and E.A. Truckenbrodt. *Aerodynamics of the Aeroplane*. McGraw Hill, 1979.

[130] V.V. Serikov, S. Kawamoto, and K. Nanbu. Particle-in-cell plus direct simulation Monte Carlo (PIC-DSMC) approach for self-consistent plasma-gas simulations. *IEEE Transactions on Plasma Science*, 27(5):1389–1398, 1999.

[131] C. Shen. *Rarefied Gas Dynamics: Fundamentals, Simulations and Micro Flow*. Springer, 2005.

[132] A.V. Skorohod. *Stochastic Equations for Complex Systems*. D. Reidel Publishing Company, 1988.

[133] S.N. Smirnov. Justification of a stochastic method for solving the Boltzmann equation. *USSR Computational Mathematics and Mathematical Physics*, 29(1):187–192, 1989.

[134] S.K. Stefanov. On DSMC calculations of rarefied gas flows with small number of particles in cells. *SIAM Journal on Scientific Computing*, 33(2):677–702, 2011.

[135] A. Steland. *Financial Statistics and Mathematical Finance: Methods, Models and Applications.* Wiley, 2012.

[136] H. Struchtrup. *Macroscopic Transport Equations for Rarefied Gas Flows: Approximation Methods in Kinetic Theory.* Springer, 2005.

[137] H. Struchtrup and M. Torrilhon. Regularization of Grad's 13 moment equations: Derivation and linear analysis. *Physics of Fluids*, 15(9):2668–2680, 2003.

[138] Q. Sun and I.D. Boyd. Evaluation of macroscopic properties in the direct simulation Monte Carlo method. *Journal of Thermophysics and Heat Transfer*, 19(3):329–335, 2005.

[139] Z. Sun, Z. Tang, Y. He, and W. Tao. Proper cell dimension and number of particles per cell for DSMC. *Computers and Fluids*, 50(1):1–9, 2011.

[140] G.W. Sutton and A. Sherman. *Engineering Magnetohydrodynamics.* McGraw-Hill, 1965.

[141] T. Tajima. *Computational Plasma Physics: with Applications to Fusion and Astrophysics.* Westview Press, 2004.

[142] E.V. Titov, D.A. Levin, and S.V. Rogazinsky. Analyses of numerical errors in the kinetic modeling of microthruster devices. *Journal of Thermophysics and Heat Transfer*, 21(3):616–622, 2007.

[143] N.G. Van Kampen. *Stochastic Processes in Physics and Chemistry.* North Holland, 2007.

[144] D.B. VanGilder, I.D. Boyd, and M. Keidar. Particle simulations of a Hall thruster plume. *Journal of Spacecraft and Rockets*, 37(1):129–136, 2000.

[145] R.V. Vichnevetsky. *Computer Methods for Partial Differential Equations.* Prentice-Hall, 1981.

[146] C. Villani. A review of mathematical topics in collisional kinetic theory. *Handbook of Mathematical Fluid Dynamics*, 1:71–305, 2002.

[147] W.G. Vincenti and C.H. Kruger. *Introduction to Physical Gas Dynamics.* Krieger, 1975.

[148] W. Wagner. A convergence proof for Bird's direct simulation Monte Carlo method for the Boltzmann equation. *Journal of Statistical Physics*, 66(3-4):1011–1044, 1992.

[149] M. Wang and Z. Li. Gas mixing in microchannels using the direct simulation Monte Carlo method. *International Journal of Heat and Mass Transfer*, 49(9-10):1696–1702, 2006.

[150] J. Wu, W. Hsiao, Y. Lian, and K. Tseng. Assessment of conservative weighting scheme in simulating chemical vapour deposition with trace species. *International Journal for Numerical Methods in Fluids*, 43(1):93–114, 2003.

[151] J. Wu and K. Tseng. Analysis of micro-scale gas flows with pressure boundaries using direct simulation Monte Carlo method. *Computers and Fluids*, 30(6):711–735, 2001.

[152] J. Yvon. *La Théorie Statistique des Fluides et l'Equation d'Etat.* Hermann et Cie, Paris, 1935.