Finite element solution of the Boltzmann equation for rarefied macroscopic gas flows.

Evans, Benjamin James

Swansea University
Prifysgol Abertawe

Finite Element Solution of the Boltzmann Equation for Rarefied Macroscopic Gas Flows

by

Benjamin James Evans

M.Eng. M.A. CANTAB

Thesis submitted to Swansea University
in candidature for the degree of Doctor of Philosophy

03/10/2008

Civil and Computational Engineering Centre

School of Engineering

Swansea University

DECLARATION

This work has not previously been accepted in substance for any degree and is not being currently submitted in candidature for any degree.

Signed...(candidate)
Date........................15/11/08.......................

STATEMENT 1

This thesis is the result of my own investigation, except where otherwise stated. Other sources are acknowledged by footnotes giving explicit references. A bibliography is appended.

Signed...(candidate)
Date........................15/11/08.......................

STATEMENT 2

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organizations.

Signed...(candidate)
Date........................15/11/08.......................

Acknowledgements

Firstly, I would like to thank Prof Nigel Weatherill for convincing me to return to Swansea to continue in postgraduate study, a decision that I will never regret!

Next, I would like to express sincere gratitude and thanks for the wisdom and patience shown by my supervisors Prof Ken Morgan and Prof Oubay Hassan, the indispensable support from Dr Jason Jones and the unfailing supply of Friday morning cups of coffee and chat provided by Prof Zienkiewicz. I would also like to mention the many great friends that have been made and that have supported me over my PhD years at Swansea making the whole experience unforgettable; Rhodri, Matt, Tim O, Tim T, Emily, Alice, Ash, Clare, Antonio, Sam .... the rest of you know who you are!

And finally I would like to acknowledge the uncountable number who have influenced the journey through my times at Jesus College, Cambridge, Gorseinon College, Bishopston Comprehensive, Newton and Eastern Primaries .... what a fantastic education experience!

'So many of the properties of matter, especially when in the gaseous form, can be deduced from the hypothesis that their minute parts are in rapid motion, the velocity increasing with temperature, that the precise nature of this motion becomes a subject of rational curiosity'.

James Clark Maxwell (1860)

I would like to dedicate this thesis to mum, dad and Dan.
Thank you for putting up with me!

Outline

This thesis presents research carried out at The Civil and Computational Engineering Centre at Swansea University between September 2004 and December 2007. The focus of the research was the application of modern finite element solution techniques to the governing equations of molecular gas dynamics in order to solve macroscopic gas flow problems. The journey of research began by considering and comparing various finite difference and finite element formulations in the solution of a simple scalar convection equation. This formed the basis for developing a solver for a variety of forms of the Boltzmann equation of molecular gas dynamics, and application of these solvers to a range of subsonic, transonic and supersonic gas flow problems. The merits and drawbacks of the molecular approach, particularly when compared with more traditional continuum CFD solvers, are identified along with possible extensions to the work presented here.

Contents

List of Figures

List of Tables

Nomenclature

In this section, some of the standard notation used throughout this thesis is summarised. Uniqueness of semantics has been attempted wherever possible. However, wherever a symbol is used with a different meaning to that defined below, it will be clearly indicated. The units of variables has also been included where relevant.

Kn	Knudsen number	*dimensionless*
λ	mean free path of molecules	m
L	length scale	m
n	molecular number density	m^{-3}
N	total number of molecules	*dimensionless*
m	molecular mass	Kg
$f(\boldsymbol{r}, \boldsymbol{c}, t)$	molecular distribution function	*dimensionless*
$f_s(\boldsymbol{c})$	single particle distribution function	*dimensionless*
$f_{sp}(\boldsymbol{r}, \boldsymbol{c}, t)$	single particle distribution function in phase space	*dimensionless*
$f_N(\boldsymbol{c}_1, \boldsymbol{r}_1, ..., \boldsymbol{c}_N, \boldsymbol{r}_N, t)$	N-particle distribution function	*dimensionless*
$f_0(\boldsymbol{c})$	Maxwellian equilibrium distribution function	*dimensionless*
$\boldsymbol{r}(x, y, z)$	position in physical space	m
$\dot{\boldsymbol{r}}(x, y, z)$	velocity vector in physical space	ms^{-1}
$\ddot{\boldsymbol{r}}(x, y, z)$	acceleration vector in physical space	ms^{-2}
$\boldsymbol{c}(u, v, w)$	velocity vector (position vector in velocity space)	ms^{-1}
$\boldsymbol{c}^*(u, v, w)$	post-collision velocity vector	ms^{-1}
$f^*(\boldsymbol{c}, \boldsymbol{r}, t)$	post-collision distribution function	*dimensionless*
$\boldsymbol{c}^{'}$	thermal/peculiar/perturbation velocity	ms^{-1}
$\boldsymbol{F}(\boldsymbol{r})$	force field vector	N
$Q(f, f)$	collision term in Boltzmann equation	
Ω_r	physical space domain	
Ω_c	velocity space domain	
Γ_r	physical space domain boundary	
$\sigma d\Omega$	differential cross section of a molecular collision	m^2
σ_T	total collision cross section	m^2
χ	deflection angle of the relative velocity vector	
b	distance of closest approach of undisturbed molecular trajectories	m
$\nu(\boldsymbol{r}, t)$	BGK coefficient ($\propto$ molecular collision frequency)	s^{-1}
X_∞	free stream value of a macroscopic flow parameter	
$\boldsymbol{v}(u, v, w)$	bulk (macroscopic) flow velocity	ms^{-1}
$\rho(\boldsymbol{r})$	density	Kgm^{-3}
$p(\boldsymbol{r})$	static pressure	Pa
$T(\boldsymbol{r})$	temperature	K
$T_k(\boldsymbol{r})$	kinetic temperature	K
k	Boltzmann constant	$(1.380630524 \times 10^{-23} J/kg)$
R	gas constant	kJ/kgK
α	absorption coefficient	*dimensionless*
$\eta(\boldsymbol{r}, t)$	flux conservation parameter	*dimensionless*
Rf	specular component of reflected distribution function	*dimensionless*
Mf	diffuse component of reflected distribution function	*dimensionless*
Q	molecular quantity	
$\bar{Q}$	mean molecular quantity	
(η, ζ)	coordinates for spectral discretisation of velocity space	*dimensionless*
Δt	timestep size	s
$X^m_{\boldsymbol{r},\boldsymbol{c}}$	X defined at timestep m, p-space coord $\boldsymbol{r}$, v-space coord $\boldsymbol{c}$	
N_k	linear finite element shape function associated with node k	
$\boldsymbol{F}(X)$	flux of variable X	
$\boldsymbol{F}_n(X)$	normal flux of variable X at a wall	
$M_L]_{re}$	lumped (3x3) f.e. mass matrix for linear, triangular elements	
A_e	element area	m^2
r_v	radial extent of velocity space	ms^{-1}
w_i	quadrature weighting of sampling point i in spectral v-space	*dimensionless*
J	transformation Jacobian	

Chapter 1

Introduction

1.1 General Background

The concept of air as a fluid exerting mechanical pressure on a surface dates back to the early 17th century. Torricelli, Pascal and Boyle are the earliest known scientists to make attempts at establishing the physical nature of air. They managed to persuade their colleagues that Earth is surrounded by a 'sea' of air that exerts a pressure much in the same way that a body of water does. Their work also convinced scientists that it was this 'air pressure' that was responsible for many of the phenomena previously attributed to 'nature's abhorrence of a vacuum'. This change in approach was part of the transition lead by Galileo, Boyle and Newton to a 'mechanico-corpuscular' view of nature which replaced the postulation of 'occult forces' or teleological principles, the philosophical 'form follows function' approach, to explain natural phenomena. It was believed that all natural phenomena would eventually be explained simply in terms of matter and motion.

But what did this 'matter' consist of? It wasn't long before scientists and philosophers started taking this question very seriously. From a philosphical standpoint, there was the ancient view that a body could not fill space continuously (in the mathematical sense), but rather it must consist of discrete parts. These discrete parts were unobservable however because of their small size. The philosophical reasoning for this proposal was straight forward: 'An actual continuum must consist of an infinite number of parts; and yet an infinite number is undefinable' [5]. Newton even talks about the importance of this question in his groundbreaking 'Philosophiae Naturalis Principia Mathematica' (1687). Alongside his famous universal laws of motion, he mentions briefly the consequences of various hypotheses about the forces between the building blocks of matter, which by now have been given the name 'atoms', on the relation between pressure and temperature.

Progress slowed in the 18th century, as the majority of physicists and mathematicians directed their energy towards the study of celestial bodies, rather than freely moving atoms. However, one contribution from this period can be regarded as the birth of The Kinetic Theory of Gases; Daniel Bernoulli's derivation of the gas laws from a 'billiard ball' model, much like the one still used in elementary textbooks today! The idea that atoms were freely moving and could travel through space until colliding with another atom was truly radical and viewed by many as too drastic. What had happened to the 'ether'?

Laplace and Poisson worked on theories of repulsion between atoms in the late 18th century, but it took a long time for these ideas to catch on. It wasn't until Lord Rayleigh took up this new way of thinking, in the late 19th century, that kinetic theory really took hold. Also taking place in the 19th century was Joule's work on the equivalence of heat, work and other forms of energy (the birth of classical thermodynamics) and Maxwell worked on theories of diffusion, viscosity and heat conduction. It was upon the shoulders of these giants that Boltzmann was able to make the discoveries that lead to the development of his famous 'transport equation' in 1872, now more popularly known simply as 'The Boltzmann Equation'.

Classical thermodynamics is an axiomatic and macroscopic science dealing with 'lumps of matter'. It presupposes no microscopic structure for these 'lumps' and it begins with a number of accepted laws, which have no definite proofs, and all results flow from a completely logical and mathematically rigorous process. These results are of very general validity, but to apply them to specific cases requires the use of empirical data or relationships, such as specific heat capacity or the gas equation of state. This empirical approach is required to 'close' and solve the governing equation system.

This macroscopic, continuum approach is satisfactory for the majority of engineering applications. However, many people do not find the empirical nature of the method completely satisfying and wish to acquire a physical understanding of how the macroscopic theory relates to the microscopic behaviour of the molecules. Also, there are a growing number of applications in fluid mechanics where the continuum hypothesis is not justified, because the empirical relations break down, and it is essential to model the fluids at the microscopic level. For example, combustion and reactive flows, multi-phase flow, nano-scale flows, rarefied gas dynamics and hypersonics.

There are two quite different theories of the molecular behaviour of fluids; statistical mechanics and kinetic theory. Statistical mechanics has been the most successful so far in analysing equilibrium situations, i.e. gases in thermodynamic equilibrium. It makes no attempt to describe the motion of molecules or their behaviour during collisions. Instead, a counting procedure is employed to enumerate all possible microscopic configurations of the molecules, the microstates, comprising a system. It is then assumed that the random motion of the molecules causes the system to pass through all possible microstates and the observed equilibrium macrostate is that state associated with the most abundant type of microstate.

In contrast, kinetic theory applies the laws of mechanics to the motion of molecules both in 'free flow' and during collisions. Macroscopic results then follow from statistical averages, obtained by integrating over all the molecules. Unlike statistical mechanics, kinetic theory can handle non–equilibrium situations. In fact, it is the only practical approach currently available for analysing certain types of non–equilibrium flow. Boltzmann's transport equation is his attempt to give this method a solid mathematical foundation. It is an integro–differential equation describing the evolution of a scalar variable, called the *velocity distribution function*, which gives the probability of finding a molecule at a given location in space in a given velocity state at a given time. If the Boltzmann equation can be solved for a particular molecular model and boundary conditions, then the velocity distribution function is known everywhere and all thermodynamic gas properties can be calculated, without making any further assumptions about the nature of the flow. Transport properties, such as the viscosity and thermal conductivity, appear naturally as part of the solution and it is, therefore, not necessary to introduce them via empirical equations. Much of modern research in kinetic theory is based upon attempts to solve The Boltzmann Equation or similar equations for other kinds of distribution functions.

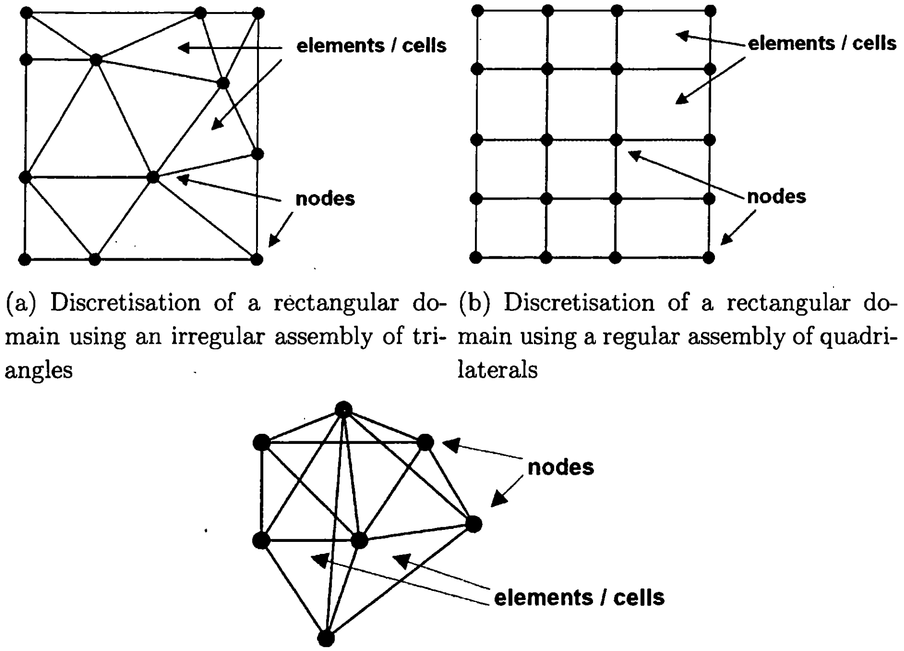
It is interesting to note that all of Boltzmann's work took place before any kind of proper establishment of the atomic and molecular structure of matter. In fact, Boltzmann committed suicide in 1906, fearing that all his work had been in vain, just a few years before Brownian motion experiments were explained by Perrin [81], confirming Einstein's theoretical explanations [28], and the scientific community came to an agreed acceptance of the existence of atoms. This must surely be one of the great ironies in the history of modern science!

1.2 Background to Computational Fluid Dynamics

Most engineering fluid flow problems are now solved, at least in part, by means of computational fluid dynamics (CFD); the use of computers to aid in the solution of the governing equations of fluid dynamics by numerical methods. The governing equations for the majority of practical fluid flow problems are partial differential equations, as is the case for many naturally occurring phenomena. It is, therefore, of no surprise that the development of CFD has paralleled the development of the mathematical tools of numerical methods for partial differential equations. Numerical methods have been known since the time of Newton in the 1700s. At this time, solution strategies were conceived, but only on paper. Without the aid of the computer, the full exploitation of these techniques was impossible.

Modern CFD has its roots in the 1950s, with the advent of the digital computer. However, the basic numerical tools of the finite difference method and the finite element method have different origins. Richardson presented the first paper on the finite difference method in 1910 at the Royal Society of London [84]. He applied the technique manually to the governing equation for the stress analysis of a masonry dam. In contrast, the first finite element technique was presented in a paper published in the Aeronautical Science Journal by Turner, Clough, Martin and Topp in 1956 [98]. They applied the technique to aircraft stress analysis. Since then, both methods have been developed extensively and have generated a massive number of variations and 'add-ons' to the original formulations. One of the newer techniques which combines the ideas of the finite difference method and the finite element method is the finite volume method. This method began its life in U.S. weapons research during the 1950s and 1960s at Lawrence Livermore National Laboratory [114], but significant development in more general applications did not occur until the 1970s ([16],[43]).

At the heart of all CFD numerical schemes is the fundamental consideration of how one should represent a continuous function in discretised form, i.e. how should one accurately store a function defined for an infinite number of points in some finite way? The most popular method of achieving this is to discretise the solution domain into a finite number of small cells/elements forming a mesh or grid, and to then apply a suitable algorithm to values stored at the intersections of the mesh



(a) Discretisation of a rectangular domain using an irregular assembly of triangles

(b) Discretisation of a rectangular domain using a regular assembly of quadrilaterals

(c) Discretisation of a 3D domain using an irregular assembly of tetrahedra

Figure 1.1: Discretisation methods in 2 and 3 dimensions

(nodes) to solve the governing equations.

The finite difference method (FDM) is based on transforming the derivatives in a partial differential equation into their discrete analogs. This approach results in a set of coupled equations with all unknowns prescribed at discrete mesh nodes across the entire solution domain. The FDM is relatively straightforward to formulate and program and hence became the dominant force in CFD in the 1960s and 70s. However, it is restricted to uniform/structured grids and therefore is difficult to apply to problems involving complex geometries.

The finite volume method (FVM) begins with the integral form of the governing equation, as does the finite element method. The unknowns are usually stored as constants across the volume, although higher order FVM methods do exist with the solution being discontinuous at the volume interfaces. A flux solver must then be applied to the discontinuous solution at each volume interface. The FVM has become popular because of its simplicity, combined with the ability to apply it to unstructured meshes, allowing a discretisation around complex geometries.

Although the finite element method was established in the 1950s, it has taken considerably longer to be taken up as a serious tool for use in CFD. In fact, even by the early 1980s its penetration was still relatively small. There are many variations of the FEM, but the most popular of these is the Galerkin method. This involves discretising the solution by means of shape/basis functions, of arbitrary order, which are defined for each element. The residual of the governing PDE is then multiplied, or 'weighted', by these basis functions and integrated over the entire solution domain, to spread and minimize the solution error. Methods based on FEM have become increasingly attractive, despite their more complex formulation because they lead to general purpose computer codes which can be applied to aribitrary mesh structures. In addition, the FEM is based on a solid theoretical foundation which has resulted in rigorously formulated procedures for elliptic [20] and parabolic [97] equations, while much is now understood about the application to hyperbolic equations [82].

A more recent development in the FEM, that has received significant recent interest is the discontinuous Galerkin FEM. In this discretisation approach, the solution is allowed to be discontinuous at element interfaces and the elements communicate by means of appropriately derived fluxes across these interfaces. There are parallels between this method and high order finite volume methods and some would argue that the discontinuous Galerkin FEM is somewhere in between finite element and finite volume methods. The key difference between the two being that in the discontinuous FEM, unknowns are still stored at nodes and interpolated across the element from these nodes, whereas, in high order finite volume schemes, a single unknown is prescribed per element / cell and the solution is 'reconstructed' across the cell using information from surrounding cells in some way. However, both require some kind of flux solver at element interfaces to deal with the discontinuities present.

It is also worth mentioning that discretisation techniques also exist which do not require a mesh (known collectively as meshless methods) for example, smoothed particle hydrodynamics. This method involves a division of the fluid into a set of discrete 'fluid elements' whose interactions are governed by means of, so called, 'kernel functions' [72].

The intense research activity of the last 50 years in the area of CFD has largely been driven by the aerospace industry, with its requirements for highly accurate solutions at minimum cost. At first, potential flow problems were the main focus but as computational resources have improved in the last few decades, the CFD community has largely focused its attention on finding solutions to the Euler and Navier–Stokes equation sets. One of the principle limiting factors in CFD has been available computing power, both in terms of processor speed and memory. One of the features of fluid flows that results in accurate solution strategies being extremely intensive numerically is turbulence. Turbulence is chaotic in nature and works at both very small and very large length scales. Therefore, to truly capture a turbulent flow accurately requires a large and highly discretised mesh. For example, it is estimated that approximately 10^{18} unknowns are required for accurate prediction of turbulent flow around a typical aircraft configuration and today we can solve problems involving approximately 10^8 unknowns at best. However, there is no need to be pessimistic about the potential for CFD. Moore's law states that available computing power doubles every 18 months and research into more and more efficient numerical techniques is currently taking place. As computing power expands exponentially and algorithms of ever increasing efficiency are developed, who know where the possibilities in CFD will end?

1.3 Why Kinetic Theory and CFD?

A gas flow may be modelled at either the macroscopic or microscopic level. However, at the macroscopic level, we must make the assumption that the gas can be regarded as a continuous medium. If this is possible, then the Navier–Stokes equations provide the appropriate mathematical description of the physics of the flow. Contrary to popular belief, this is not always the case! If we cannot 'close' the equation set by assuming some empirical relationships between the macroscopic flow properties, e.g. the ideal gas equation and relationships for shear stress and heat flux, then the Navier–Stokes equations are intractable.

We can classify a flow in terms of a nondimensional parameter called the Knudsen number (Kn), defined as

$$Kn = \frac{\lambda}{L} \tag{1.1}$$

where λ is the mean free path of molecules in the flow, i.e. the average distance travelled between collisions, and L is some typical length scale in the flow. The traditional rule–of–thumb is that Kn should be less than 0.1 for validity of the continuum assumption. However, this can be misleading if L is chosen to be some overall dimension of the flow in order to define a single global Kn for the flow. The limit can be defined more precisely if we define a local Kn with L

$$L = \frac{\rho}{d\rho/dx} \tag{1.2}$$

as the scale length of the macroscopic gradients where ρ is the fluid density.

The Knudsen number limits on the conventional mathematical formulations to describe gas flows is shown schematically in Figure 1.2 (taken from [5]).

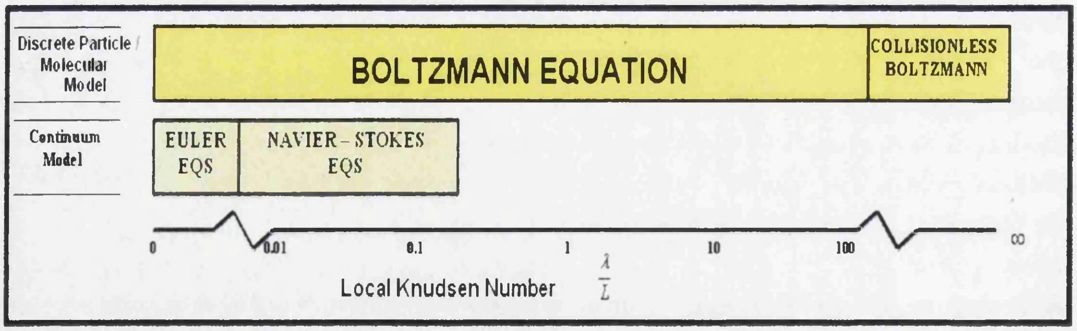


Figure 1.2: The Knudsen Regime

It is clear from Figure 1.2 that there is a significant range in the Knudsen regime within which the traditional continuum equations are no longer valid. There are many hypersonic applications which involve low density / high Kn flow, generally involving flight at high altitudes. For example, it has been noted in [74] that flow in the nose region of the space shuttle above 92km (about 300,000ft) cannot be treated properly by purely continuum assumptions and, for any given flight, as altitude increases, and hence the density decreases and λ increases, the continuum assumption becomes tenuous. An altitude will be reached where the conventional viscous no–slip condition begins to fail and the flow velocity at a surface takes on some finite value. This is termed the velocity slip condition. In a similar way, a temperature differential may be set up between the surface material and flow at the surface. This is called the temperature slip condition. At this point, the continuum equations may still be applied, with modifications to the boundary conditions, and successful solutions achieved.

However, as altitude continues to increase, a point is reached where we can no longer cling onto the continuum equations. Then we must seek solutions to the fundamental Boltzmann equation and think in terms of the underlying molecular kinetics. Finally, the air density can become low enough for the collisions between molecules to be ignored. For the space shuttle, this free molecular regime begins at an altitude of about 150km (500,000ft).

Therefore, in a very simplified sense, we can think of a spacecraft re–entry passing through various distinct phases; first the free molecular regime in which we must seek a solution to the collisionless Boltzmann Equation, next the spacecraft will enter the kinetic regime, in which we must seek a solution to the full Boltzmann Equation, then the transition regime is entered and we can model the flow using the continuum equations with modified boundary conditions, and finally we enter the continuum regime, in which the standard continuum equations apply. This is an overly simplistic view because the regime is based on a local Knudsen number which means that in practice, different regions of any given flow may actually have a different local value of Kn.

There is a further, more subtle, reason for using kinetic theory and the governing Boltzmann Equation as the starting point for analysing hypersonic flows. Continuum theory assumes that a gas is always in thermodynamic equilibrium, i.e. the distribution of molecular velocities is always described by a Maxwellian distribution function. This, in turn, is based on the assumption that there have been a sufficient number of molecular collisions, in a sufficiently small frame of time, to return any perturbations from equilibrium back to the equilibrium state. However, there are many problems in high–speed gas dynamics where the gas is not given the luxury of the necessary time to come to equilibrium. A typical example is flow across a shock wave, where the pressure and temperature are rapidly increased within the shock front. When the pressure and temperature of a gas are changed, its vibrational and chemical equilibrium properties will change. The fluid element will start to 'seek' these new properties but this requires collisions and, hence, time. By the time sufficient collisions have occured, the fluid element will have moved some distance downstream of the shock front, resulting in a region downstream of the shock in which equilibrium conditions do not prevail. We must, therefore, study such regions using kinetic theory, including the vibrational and chemical degrees of freedom of the gas.

It will be useful, at this stage, to introduce the Boltzmann equation, for which a derivation will be provided in the following chapter, to allow a discussion on the solution strategy,

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}} = \frac{1}{Kn}Q(f,f) \tag{1.3}$$

where $f = f(\boldsymbol{r}, \boldsymbol{c}, t)$ is the distribution function, n is the molecular number density, $\boldsymbol{F}$ describes any force fields (gravitational, electrostatic etc) that might be present, $Q(f, f)$ is the term accounting for molecular collisions.

The term $Q(f, f)$ is a five–fold integral, in 3 space dimensions, and is the source of many of the difficulties in solving this equation. The Boltzmann equation is an integro–differential equation and analytical solutions are restricted to extremely simple applications ([70],[1]). We must, therefore, adopt a numerical solution approach.

The numerical approaches for solving the Boltzmann equation can be broadly categorized into two classes of methods. The first are direct Boltzmann CFD methods whereby the governing equation is discretised and solved directly. The second are based on the simulation of the physics of the flow and do not depend on a mathematical model. The situation has been complicated by efforts to link some of the simulation methods to the Boltzmann equation with sufficient rigour that they may be regarded as 'solutions' of that equation. The situation is summarised in Figure 1.3

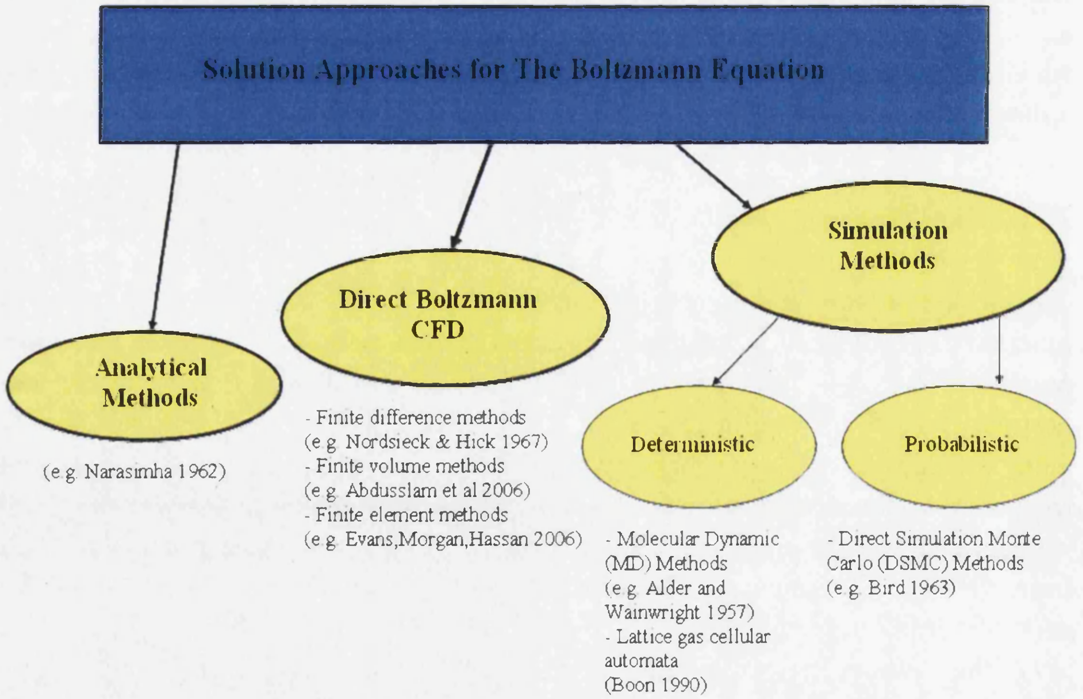

Solution Approaches for The Boltzmann Equation
Analytical Methods
(e.g. Narasimha 1962)
Direct Boltzmann CFD
- Finite difference methods
(e.g. Nordsieck & Hick 1967)
- Finite volume methods
(e.g. Abdusslam et al 2006)
- Finite element methods
(e.g. Evans,Morgan,Hassan 2006)
Simulation Methods
Deterministic
- Molecular Dynamic (MD) Methods
(e.g. Alder and Wainwright 1957)
- Lattice gas cellular automata
(Boon 1990)
Probabilistic
- Direct Simulation Monte Carlo (DSMC) Methods
(e.g. Bird 1963)

Figure 1.3: Categorisation of Solution Approaches

Today, CFD has a particularly strong impact in hypersonic aerodynamics research, in comparison with other strands of fluid dynamics, mainly because of the limited availability of hypersonic wind tunnels and other hypersonic ground–test facilities. Thus, in the modern world of hypersonics, CFD serves as a powerful tool for research, development and design.

1.4 Background to Hypersonics

1.4.1 What is Hypersonic Aerodynamics?

The previous section has shown that one area of fluid dynamics in which a kinetic approach is certainly appealing, if not essential, is hypersonics. The first question one must ask when approaching the topic of hypersonic aerodynamics is 'What is hypersonic flow?' Sometimes this question is answered with an overly simplistic answer; 'whenever the freestream Mach number is greater than 5'. But this is nothing more than a rule–of–thumb. In reality, a flow truly becomes 'hypersonic' when one or more of the following phenomena, highlighted by Anderson [47], dominate:

1. Thin Shock Layers

At high Mach numbers, the shock waves supported by a vehicle are very strong and, hence, invoke a massive density increase. The mass flow behind a shock can therefore 'squeeze' through a small area, resulting in the traditional bow shock 'wrapping around' the vehicle to form a thin 'shock layer'.

2. Entropy Layers

At hypersonic Mach numbers, as well as the shock layer being very thin, the shock detachment distance is very small and, in a blunt body nose region, the shock wave is highly curved. This high curvature results in large gradients of entropy in the blunt body shock layer. This 'entropy layer' flows downstream and can 'wet' the body a large distance downstream from the nose. This 'entropy layer' affects the way in which we treat the body's boundary layer.

3. Viscous Interaction

At hypersonic Mach numbers, boundary layers can get very thick. In fact, the boundary layer can become so thick that it merges with the shock wave resulting in a 'merged shock layer'. When this happens, the entire shock layer must be treated as fully viscous and conventional boundary layer analysis must be completely abandoned.

4. High Temperature Effects

A high velocity flow contains large amounts of kinetic energy, which gets partly dissipated in the boundary layers and shock waves as thermal energy. Hypersonic flows are, therefore, prone to very high temperatures and high temperature effects, such as vibrational excitation, dissociation, ionization and chemical reactions may require consideration.

5. Rarefied Gas Effects

As the mean free path between molecular collisions increases towards typical length scales of the flow, the ratio of these two being characterised by the Knudsen number, the 'transition regime' is entered whereby the continuum theory, on which the previous discussion in this section has been assumed, breaks down and we must revert to studying the flow at the molecular level. This transition regime occurs when the Knudsen number increases. This can be caused either by the mean free path increasing, e.g. in the outer atmosphere during spacecraft re–entry, as discussed in the previous section, or in regions of the flow with small length scales, e.g. very thin shocks. A typical transient hypersonic flow may pass through a range of Knudsen regimes and, at any time, feature different regions of the flow in different regimes.

1.4.2 The History of Hypersonic Research

Many would view the 'glory days' of hypersonic research as the late 50s and 60s, when aerodynamic research was dominated by hypersonics. The motivation for this research is clear; earlier emphasis on re–entry design for intercontinental ballistic missiles was now greatly reinforced by the fledgling Apollo program, whose goal was to place man on the moon by the end of the 60s. All aspects of hypersonic flow were under intense investigation. This research was, however, in the most part experimental, featuring large and expensive hypersonic wind tunnels and shock tubes. These facilities were appearing at all major aerodynamic laboratories in government, industry and even in universities. By the 1970s, however, the situation could not have been more different. The Apollo missions had now finished, the design of the new space shuttle fixed and the design for warheads somewhat standardised. In fact, in the 1970s, we are hard pushed to find any obvious signs of hypersonic research at all. Many hypersonic wind tunnels and shock tubes had been decomissioned and some even cut up to be used for scrap metal! However, some hypersonic research continues under the guise of CFD which is undergoing an explosive growth period. Hypersonic applications were being used in CFD to validate new codes and techniques. This transition in research emphasis, from experimential to computational, has also been observed in other fields of fluid dynamics in recent decades. The rise in CFD research has been largely due its relatively low cost, high speed, repeatability and versatility. It is important to note that CFD has not, and probably will not ever, completely replace experimental research in fluid dynamics, as experimental data will always be required to validate new CFD codes. In fact, experimental research at NASA Langley is still going strong, making use of their impressive 8-Foot High Temperature Tunnel. This tunnel has the capabilities for experimental testing at speeds of up to Mach 7 through a range of altitudes from 50,000 to 120,000 ft.

This brief study of the history of hypersonics research does now beg the question 'why should we study hypersonics today?' Since the Wright brothers' pioneering first flight in 1903, humans have always had an urge to fly faster, higher and further and so it is to this day. Although funding for research into high speed flight has ebbed and flowed over time, this underlying drive has always existed and shows no sign of disappearing. NASA have recently announced new plans for man's return to the moon and also plans for manned flights to Mars and, along with the ESA (European Space Agency), have missions planned, and undergoing, for interstellar and planetary probes, pushing further and further into our Solar System and beyond. The current model space shuttle will probably reach the end of its lifespan within the next decade, leading to new developments in reusable spacecraft technology requiring the most efficient re–entry design. Also, with the launch of Virgin Galactic in 2004 and the company's more recent highly publicised unveiling of 'SpaceShipOne', the 21st century may witness a whole new commercial industry in space travel. Design concepts have been launched for hypersonic passenger aircraft, cruising at between Mach 7 and Mach 12, with the capability of carrying people from New York to Tokyo in less than two hours! Many of these ideas are increasing in viability and turning from pipe dreams into reality largely due to developments in SCRAMjet (supersonic combustion ramjet) technology, which is now capable of providing extreme high speed propulsion. A wide range of hypersonic applications currently under development have already been mentioned here, without even acknowledging developments within the military sector. It seems clear that there will be a trend towards faster aircraft and spacecraft in the future and many exciting challenges for the hypersonic aerodynamicist are being posed [27].

1.5 Thesis Outline

This thesis is divided into five distinct chapters entitled:

1. Introduction
2. Problem Formulation
3. The Solution Algorithm
4. Examples
5. Concluding Remarks

In the remainder of Chapter 1, the objectives and history of the project are summarised and a review of the literature is presented. This provides sufficient basis to move onto Chapter 2; Problem Formulation.

Chapter 2 begins with an introduction to the single dependent variable of the Boltzmann equation; the velocity distribution function. Also, a review of binary molecular collision theory is provided before a derivation of the Boltzmann equation from first principles. The connection of the Boltzmann equation to the Liouville equation is discussed and the BGK model equation introduced. Finally, the crucial step of regaining macroscopic data from the velocity distribution function by taking moments of the Boltzmann equation is described.

Chapter 3 begins by looking at the methodology employed for discretisation of the solution domain of the Boltzmann equation; phase space. The discussion of the solution procedure begins with a look at a standard discontinuous Taylor–Galerkin algorithm for the scalar convection equation as this forms the basis of the Boltzmann solution procedure. The discussion then moves on to the solution algorithm employed for the Boltzmann equation in a variety of forms; the collisionless Boltzmann equation, the BGK–Boltzmann equation and the full Boltzmann equation. The procedure for implementation of the boundary conditions is then provided, along with the methodology for computing the moments of the Boltzmann equation.

Chapter 4 begins by comparing a variety of techniques for solving the scalar convection equation beginning with standard finite difference procedures before moving on to a comparison of the Taylor–Galerkin procedure using continuous and discontinuous elements. Example problems are then studied to benchmark the Boltzmann solver.

Initially, the collisionless Boltzmann solver is applied to high Knudsen number problems including a shock tube problem and flow over a vertical plate in 2D. These solutions are compared with data in the literature. A highly rarefied hypersonic example of Mach 25 flow over a double ellipse geometry, simulating a generic re–entry configuration, is then shown.

The BGK solver is then applied to flow over a vertical plate at a variety of Knudsen number below the free molecule limit and again results are compared with data in the literature. The BGK solver is then applied to a range of aerofoil flow problems including subsonic, transonic and supersonic conditions. The failure of the BGK solver at high Mach number is also demonstrated.

Finally, a validation example for the full Boltzmann solver is presented. It shows the ability of the full collision term to restore a non–Maxwellian distribution function to equilibrium. The full Boltzmann solver is then applied to a number of flow problems at a range of Mach numbers and, critically, the limitations of this methodology are highlighted and discussed.

In Chapter 5, the main conclusions derived from the work in this thesis are identified and discussed. Possible extensions to the work in this thesis are then highlighted.

1.6 Objectives of the Project

The objectives of this project were set out in the summer of 2004. At this time, the author was completing a Masters degree program in which he became interested in the field of molecular gas dynamics. This interest, coupled with the high level of expertise in CFD at the Civil and Computational Engineering Centre at the University of Wales, Swansea, led to the initial concept for the project of using traditional numerical techniques to find solutions to flow problems based on the Boltzmann equation of molecular gas dynamics. Also, at this time there was (and continues to be) significant interest in the discontinuous Galerkin finite element procedure. It was, therefore, agreed that a discontinuous Galerkin finite element solution procedure would be applied to the Boltzmann equation.

It was clear from an early stage that using the Boltzmann equation to study flow phenomena would be significantly more expensive than using standard CFD techniques. The specific goals of the project were therefore set out as follows:

To develop a 2D Boltzmann equation solver using a discontinuous Galerkin finite element procedure in order to

- have a flow solver for non–equilibrium gas flow problems such as highly rarefied gas flows and hypersonic flows.

- gauge the competitiveness of a full Boltzmann solver over standard CFD techniques under other flow conditions (i.e. transition regime flows, continuum regime flows).

1.7 Project History

The timescale for this project was three years and there are some clear phases of the project that can be identified.

Year one began with a literature review (detailed in the following section) and a review of computational fluid dynamics techniques, specifically the finite element method. It became clear that at the heart of the Boltzmann solver, an accurate scalar convection equation solver would be necessary. Time was, therefore, invested in the early stages of the project into analysing different scalar convection equation solution approaches including both finite difference methods and finite element methods. An existing C2EC Taylor–Galerkin code was used as a basis for the finite element work. This was then adapted for discontinuous elements to allow a comparison of the discontinuous and continuous finite element techniques.

Year two consisted of the development of the discontinuous Taylor–Galerkin scalar convection solver into a collisionless Boltzmann equation solver. At this stage, significant time was invested into an effective velocity space discretisation method. Also, boundary condition application procedures required much development time. On achieving a working collisionless Boltzmann equation solver, a variety of free-molecule flow examples were analysed using the solver. The code was parallelised at this stage as the allowable problem size became limited due to memory constraints.

In year three the main thrust of the project turned to dealing with the right–hand side of the Boltzmann equation. First, the simplified Boltzmann–BGK equation was considered and tested. Once the limitations of the Boltzmann–BGK solver became apparent, the full Boltzmann right–hand side was tackled and a Monte Carlo approach developed. Some test cases were attempted and the limitations of using the full Boltzmann solver for realistic flow problems identified.

1.8 Literature Review

In this section, we begin by reviewing the literature directly related to the content of the aforementioned introductory theory. Books, papers, articles concerned with the Boltzmann equation, relevant numerical methods, specifically discontinuous Galerkin procedures and hypersonics research are first highlighted. Next, material concerned with 'non–finite element' solution approaches for the Boltzmann equation are identified. These include lattice Boltzmann techniques, discrete velocity models, Monte Carlo methods and gas kinetic schemes. We then examine other finite element solution approaches to the Boltzmann equation that have been attempted and some analytical solutions.

Two books that provide much of the theory for the essential groundwork required to understand the characteristics of the Boltzmann equation are 'The Boltzmann Equation and its Applications' by Carlo Cercignani [18] and 'Molecular Gas Dynamics and the Direct Simulation of Gas Flows' by Graham Bird [5]. In his book, Cercignani effectively demonstrates the necessity of a statistical approach to problems in kinetic theory. He states that it is impossible to specify the absolute initial conditions for a specific problem in kinetic theory as this requires knowledge of the positions and velocities of every particle relevant to the problem, which in principle could be every particle in the universe! We, therefore, must revert to a method in which we are 'averaging over our ignorance' and it is this very statistical philosophy that leads us to the Boltzmann equation. He considers in detail probability density functions and the connection between the Boltzmann equation and the Liouville equation and Liouville's theorem. Cercignani also provides a rigorous derivation of the Boltzmann equation using the Liouville equation as his start point. Bird also provides a derivation of the Boltzmann equation, but rather than beginning with the Liouville equation, his derivation is from first principles. He, therefore, has to provide a lengthy discussion on the binary molecular collision theory necessary for deriving the collision term on the right–hand side of the Boltzmann equation. The second half of the book focuses on the direct simulation Monte Carlo method for analysis of the Boltzmann equation. This is a technique pioneered by Bird himself. A collection of Boltzmann's own work is contained in the book 'Lectures on Gas Theory' [7]. In this collection of work, we find a huge range of topics related to Boltzmann's expertise in the field of kinetic theory ranging from derivations of mean free path to his famous H–theorem. Much of the work is discussed from a philosophical standpoint. In [69], Bhatnagar, Gross and Krook demonstrate the usefulness of transforming the full collision integral right–hand side of the Boltzmann equation into a simpler form, resulting in their BGK–Boltzmann model equation. Analytical solutions to the Boltzmann equation are limited, due to the complexity of the equation. However, one famous analytical approach is provided by Mott–Smith [75], in which an analysis of the density and velocity profiles through a shock wave is provided by assuming a 'bimodal' distribution function.

Morgan and Peraire provide an excellent overview of the Galerkin finite element method for problems in fluid mechanics in [79]. This review places a particular emphasis on adaptive grid procedures, but also covers the history of computational fluid dynamics, details of the Galerkin method, stabilization, explicit and implicit timestepping, discontinuity capturing and TVD schemes.

Since the advent of the multiple processor supercomputer, parallel algorithms have played a crucial role in allowing the analysis of more complex, CPU intensive, high memory problems especially in three dimensions. The benefits of parallel simulations are clearly demonstrated in [51],[30],[92],[110].

This thesis deals largely with a technique based around the Taylor–Galerkin finite element method. The Taylor–Galerkin method is the finite element equivalent of the Lax–Wendroff method in finite differences. The Lax–Wendroff method was introduced by Lax and Wendroff in a paper looking at solutions of hyperbolic equations with high order of accuracy in 1964, [112]. Leith also adopted this approach in his 'Methods in Computational Physics' published in 1965, [53]. The Taylor–Galerkin approach was introduced by Donea [25] and has since been used widely ([96],[73],[111]) in both its one–step and two–step formats as an alternative to Runge–Kutta timestepping. Qiu et al. provide a good comparison of Taylor–Galerkin timestepping (or 'Lax–Wendroff Type Timestepping') with the Runge–Kutta approach in the context of the discontinuous Galerkin method [90].

The term 'discontinuous Galerkin' (DG) is first encountered in the literature in 1978 in a paper by Delfour and Trochu [101], but did not become widely used until the early 1980s. Zienkeiwicz and Taylor provide a thorough, albeit sceptical, overview of the method [101], in which they examine the various methods of 'linking' the discontinuous subdomains for convective and diffusive problems. They also examine the relationship between discontinuous Galerkin methods and finite volume methods. It is observed that finite volumes, as frequently used, are simply discontinuous Galerkin elements wherein very low order approximations are assumed to represent the variable in the cell. In [91], Zhang and Shu also note the similarities between discontinuous Galerkin and finite volume methods. They make comparisons between a discontinuous Petrov–Galerkin procedure and a spectral finite volume method for analysing 1D, linear, hyperbolic conservation laws. Bernardo Cockburn has been a staunch advocate of discontinuous methods in recent times. He provides a brief introduction to the method [22] and in a plenary lecture presented at the 80th Annual GAMM Conference in 2002, he emphasises the benefits of the discontinuous methods in terms of the significances of the block–diagonal mass matrices generated by the method in terms of parallelizability, as well as the significance of the local conservativity of such schemes at the element level. He also stresses the benefits of being able to vary the order of the polynomial approximation on each element which makes the method ideal for adaptivity algorithms. In [21], he identifies a wide variety of problems (1D and 2D) in which discontinuous methods 'perform well'. An important point raised by Cockburn is the importance of the definition of, what he terms, the numerical trace, this will later be referred to as the inter–element flux in this thesis, in terms of the consistency, stability and accuracy of the particular DG scheme. Cockburn also, importantly, identifies that the 'jumps' in the approximate solution that results from the DG method are the source of the stabilising artificial viscosity of the scheme. The scheme is thus stabilised without the loss of accuracy resulting from traditional artificial viscosity methods. This is particularly important when resolving shocks in the solution as the DG method will tend to 'trap' any spurious oscillations in the elements in the vicinity of the shock rather than allowing them to propagate throughout the domain. Other techniques in the literature that have been used to overcome this problem include the flux corrected transport (FCT) method and kinetic flux vector splitting method ([24],[103],[9],[60]). The flux vector splitting method will be examined in more detail below as it has a significant relationship with a direct Boltzmann scheme. Recently, the DG method has been used in a wide range of applications . For example, Persson and Peraire [80] demonstrate the shock capturing characteristic of the DG method when applied to the Euler and Navier–Stokes equations . Qiu, Dumbser and Shu [90] develop a Taylor–Galerkin (Lax–Wendroff) variation of the DG method. Brubeau, Sagaut and Bruneau [11] look at limiting for the DG method and Grooss and Hesthaven [38] look at a DG method for free surface flows.

All of the theory of hypersonic aerodynamics relevant to this thesis is contained in a book by John D. Anderson Jr. entitled 'Hypersonic and High Temperature Gas Dynamics', [47]. This book covers some important ground, including the definition of hypersonic flow, rarefied hypersonic flow and examines a wide variety of viscous and inviscid hypersonic flow examples. He also provides a survey of hypersonics research as things stood in the mid 1980s in [46]. Hypersonic flow has been analysed by a wide range of methodologies ranging from the analytical techniques such as the Lighthill–Freeman model ([57],[31]) and those of Anderson [47] and the experimental techniques of Harvey et al in [32], through standard CFD techniques [71] to the innovative direct simulation Monte Carlo (DSMC) of Bird [5] used, for example, by Yamaguchi et al [64]. This thesis details a further approach to the study of rarefied hypersonic flows, that of direct numerical solution of the Boltzmann equation. A comparative study of some of the aforementioned methodologies is provided by Chen et al [8].

A class of methods that has the Boltzmann equation at its heart, although cannot be truly regarded as providing solutions of the Boltzmann equation, are lattice Boltzmann methods (LBMs), [26]. The LBM is a mesoscopic particle–based approach to simulate fluid flows and has become a serious alternative to traditional methods in CFD for certain applications. The LBM is often derived from the simplified Boltzmann–BGK equation (see section 2.3). In lattice gases, particles live on the nodes of a discrete lattice. Particles 'jump' from one node of the lattice to another according to their discrete velocities. This highly simplified view of the propagation of molecules differentiates it from the Boltzmann solver detailed in this thesis. Particles may collide and acquire new discrete velocities in the 'collision phase'. This method is applied to three dimensional flows in pipes and around a sphere at moderate Reynolds numbers by Rossi et al [93] and an example of the treatment of boundaries is provided by Yu, Mei and Shyy [113]. There are many cases of the LBM being applied to microchannel flows, thermal problems and mesoscale/nanoscale flows in the literature ([56],[109],[58]). However, the author is unaware of any successful applications to rarefied hypersonic flows to date.

A class of methods known as discrete velocity models (DVM) ([68],[107],[6]) exist that have much in common with the LBM. Succi [94] provides a comparison of the LBM and DVM pointing out that even though they are close relatives in mathematical terms (since they are both based on grid–bound particles moving with a set of discrete speeds), the DVM remains more faithful to the underlying kinetic theory whereas the LBM is more concerned with 'capturing hydrodynamic phenomena', i.e. it is a more phenomenological methodology.

As was mentioned in Section 1.3, there are molecular interaction 'simulation' methodologies that have been linked with the Boltzmann equation rigourously enough that they have been regarded as 'solutions' of the equation [99]. One such technique is the direct simulation Monte Carlo (DSMC) method pioneered by Bird [5]. In the strictest sense, the DSMC method is merely a phenomenological model, simulating the gas molecules rather than being derived from fundamental theory. The DSMC method is, however, extremely popular for simulating rarefied and highly non–equilibrium gas flows ([13],[76],[83]) because of its efficiency and parallelisability [102] and will almost certainly be the major competitor to a direct Boltzmann solver for the type of applications in which the Navier–Stokes equations become invalid. The method uses a probabilistic Monte Carlo approach in the tracking of simulation molecules, representing a large number of real molecules, through physical space and in the modelling of inter–molecular collisions and molecule–surface collisions. However, even though historically Monte Carlo simulations have been the usual method of choice to tackle such problems because of the inability to resolve the phase space on available computers, recently, work ([33],[77]) has demonstrated that efficient deterministic methods can be competitive with Monte Carlo methods.

A family of numerical schemes which may be generically described as 'kinetic schemes' exist ([19],[44],[104],[45],[95],[59])that are designed specifically for the analysis of high Knudsen number flows. The method involves a Chapman–Enskog expansion analysis [23] of (usually) the Boltzmann–BGK equation to form a set of equations, similar in form to the Navier–Stokes equations, but including additional terms to account for the inter–molecular interactions and molecule–surface interactions. This removes the need for any kind of 'special' boundary treatment for high Knudsen number flows using the standard Navier–Stokes equations. One popular variant of such types of scheme is the Kinetic Flux Vector Splitting method ([29],[63],[55]). This scheme employs a particular type of upwinding based on the flow physics at the Boltzmann level rather than the Euler/Navier–Stokes level. It is important to note that such methods are not direct solutions of the Boltzmann equation. Nevertheless, as with the DSMC method, kinetic schemes will be a significant competitor to direct Boltzmann solution schemes.

There are a limited number of direct numerical Boltzmann solvers detailed in the literature at the time of writing. Aristov [3] provides a general overview of discretisation approaches, especially focussing on the discretisation of velocity space. However, in physical space, he is limited to a finite difference approximation. Analysis of the literature on direct numerical approaches makes it clear that one of the principal difficulties in solution of the Boltzmann equation lies in the right–hand side collision term. A 'spectrally accurate approximation of the collision operator' is presented in [88] and [87]. A high order method for approximating the distribution function in velocity space based on ideas from these papers is detailed in this thesis. In [78], a second order upwind finite difference scheme is applied to the simplified BGK and higher moment models of the Boltzmann equation. Successful application in both rarefied and continuum hypersonic applications is achieved. Other direct Boltzmann solution procedures do exist ([77],[86],[15]). However, the majority of applications are in micro–scale flows. Application to larger scale flows places very different requirements on the discretisation of phase space. The only other Boltzmann solver available at the time of writing (to the author's knowledge) that employs a discontinuous Galerkin procedure in physical space is by Gobbert and Cale [14]. They present a 3D kinetic transport equation solver for reactive, rarefied flows and apply it to chemical vapor decomposition (nano–scale) problems. They outline a methodology similar to that presented in this thesis whereby the Boltzmann equation is reduced to a 'large number of conservation laws', each solved by means of a discontinuous Galerkin scheme. A method similar to this was first introduced by Reed and Hill [39] for solving the neutron transport equation.

The idea behind the approach presented in this thesis was first outlined by Evans et al [37]. In this paper, the general methodology was applied to the collisionless form of the Boltzmann equation (valid for high Knudsen number flows) and used to analyse a rarefied shock tube example and rarefied subsonic flow over a vertical plate. A possible extension to the work in this thesis is hinted at by le Tallec and Mallinger [52] in which they identify a strategy for coupling Navier–Stokes solution domains to Boltzmann solution domains.

1.9 Publication List

A Discontinuous Taylor–Galerkin Finite Element Approach for Solution of the 2D Collisionless Boltzmann Equation
B. Evans, K. Morgan, O. Hassan
Proceedings of a Joint Conference of The Association for Computational Mechanics in Engineering (UK) and The Irish Society for Scientific and Engineering Computation
pp 135-138 (2006)

A Discontinuous Taylor–Galerkin Finite Element Approach for Solution of the Boltzmann-BGK Equation
B. Evans, K. Morgan, O. Hassan
Proceedings of the 11th International Conference on Civil, Structural and Environmental Engineering Computing
(2007)

Chapter 2

Problem Formulation

2.1 The Velocity Distribution Function

In a classical sense, a gas flow would be completely described if we could list the positions, velocities and internal states of all its constituent molecules at any instant in time. The number of molecules in a real gas is, however, far too large for this type of description to be possible and we, therefore, resort to a statistical description in terms of probability distributions. There are a number of distinct velocity distribution functions employed in kinetic theory and to avoid any confusion, especially when referencing other work in this field, the relationship between them will be reviewed.

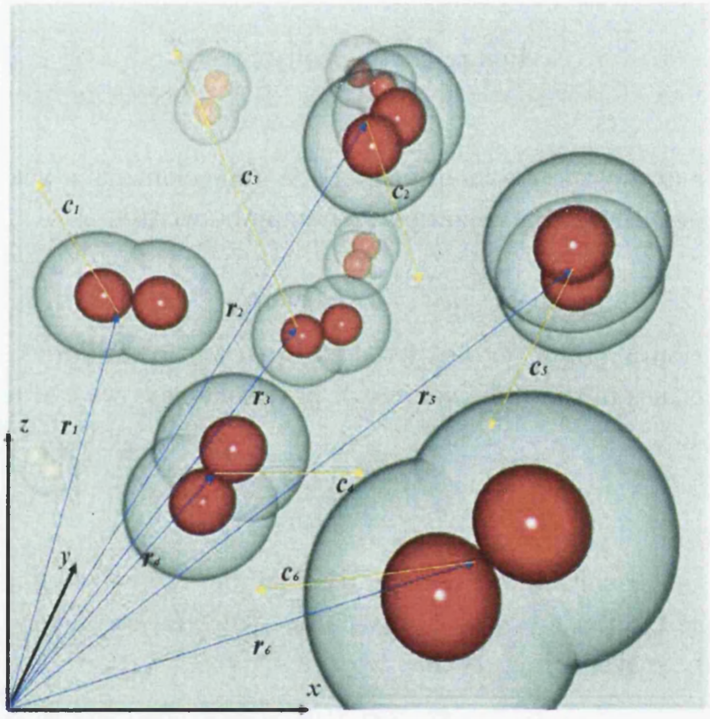


Figure 2.1: Positions and Velocities of a Group of Molecules

2.1.1 The Single Particle Distribution Function

The single particle distribution function, $f_s(\boldsymbol{c})$ for a group of N particles is defined to be such that

$$\mathrm{d}N = N f_s(\boldsymbol{c}) \mathrm{d}u \, \mathrm{d}v \, \mathrm{d}w \tag{2.1}$$

where $\mathrm{d}N$ is the number of molecules in the sample with velocity components in the range u to $u + \mathrm{d}u$, v to $v + \mathrm{d}v$,w to $w + \mathrm{d}w$ where $\mathrm{d}u, \mathrm{d}v, \mathrm{d}w$ are small incremements in u, v, w. We call these class $\boldsymbol{c}$ molecules, where $\boldsymbol{c} = (u, v, w)$.

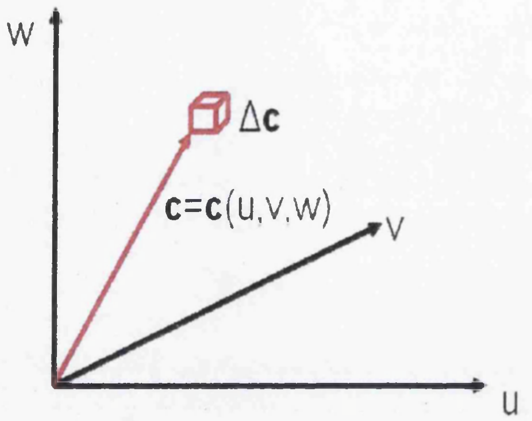


Figure 2.2: Velocity Space

The product $\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}w$ may be identified as a volume element in velocity space and can be denoted as $\mathrm{d}\boldsymbol{c}$, so that equation (2.1) may be written as

$$dN = Nf_s(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} \tag{2.2}$$

and this definition is no longer restricted to Cartesian coordinates. Since $\mathrm{d}N$ and N refer to molecules in the same volume of physical space, we can rewrite equation (2.2) in the form

$$\frac{dn}{n} = f_s\,\mathrm{d}\boldsymbol{c} \tag{2.3}$$

where n is the molecular number density (the number of molecules per cubic metre) and $\mathrm{d}n/n$ is the fraction of class $\boldsymbol{c}$ molecules. Since all molecules must occupy a point in velocity space, it follows that

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f_s\mathrm{d}u\,\mathrm{d}v\,\mathrm{d}w = \int_{-\infty}^{\infty} f_s\mathrm{d}\boldsymbol{c} = N/N = 1. \tag{2.4}$$

This is known as the normalisation condition for the single particle distribution function and provides a good check in any numerical scheme that one is computing a valid distribution function. It is worth noting that, since f_s is a probability function, it can never take on a negative value and must either have finite bounds in velocity space or tend to zero as $|\boldsymbol{c}|$ tends to infinity.

2.1.2 The Single Particle Distribution Function in Phase Space

So far, we have only considered a statistical description of the velocity distribution of molecules at a fixed point in physical space, i.e. f has been a function of $\boldsymbol{c}$ alone. However, it is clear that, just as the macroscopic gas properties, e.g. bulk velocity, can vary in physical space and time, so too can the form of the velocity distribution function. We, therefore, define the single particle distribution function in phase space as

$$dN = f_{sp}(\boldsymbol{c}, \boldsymbol{r}, t)\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r} \tag{2.5}$$

with $\mathrm{d}N$ now representing the number of molecules in the phase space element $\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r}$. In cartesian coordinates, this means that $\mathrm{d}N$ is the number of molecules with spatial coordinates lying in the range x to $x + \mathrm{d}x$, y to $y + \mathrm{d}y$, z to $z + \mathrm{d}z$ where $\mathrm{d}x, \mathrm{d}y, \mathrm{d}z$ are small increments in x, y, z, and with velocity components lying in the range u to $u + \mathrm{d}u$, v to $v + \mathrm{d}v$,w to $w + \mathrm{d}w$. Note that f_{sp} defines the total number of molecules in a phase space element, rather than the fraction of molecules, and integrating over all phase space gives the total number of molecules in the system, N, rather than unity. Applying the single particle distribution function in velocity space to a physical space element $d\boldsymbol{r}$, produces the relationship

$$dN = Nf_s(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} = f_{sp}(\boldsymbol{c}, \boldsymbol{r}, t)\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r} \tag{2.6}$$

which implies that

$$nf_s(\boldsymbol{c}) \equiv f_{sp}(\boldsymbol{c}, \boldsymbol{r}, t). \tag{2.7}$$

since the number density in the phase space element is $N/d\boldsymbol{r}$.

2.1.3 The N-Particle Distribution Function

The most descriptively specific form of the distribution function is that which gives the probability of finding a system of N molecules existing at a point in a 6N dimensional phase space i.e. 3N physical space dimensions and 3N velocity space dimensions. The probability, P, of finding the system in an elemental volume in this 6N dimensional space is represented as

$$P = f_N(\boldsymbol{c_1}, \boldsymbol{r_1}, \boldsymbol{c_2}, \boldsymbol{r_2}, ..., \boldsymbol{c_N}, \boldsymbol{r_N}, t)\,\mathrm{d}\boldsymbol{c_1}\,\mathrm{d}\boldsymbol{c_2}...\mathrm{d}\boldsymbol{c_N}\,\mathrm{d}\boldsymbol{r_1}\,\mathrm{d}\boldsymbol{r_2}...\mathrm{d}\boldsymbol{r_N} \tag{2.8}$$

where f_N is the N-particle distribution function.
A reduced particle distribution function, f_R, for R of the N molecules is defined by

$$f_R(\boldsymbol{c_1}, \boldsymbol{r_1}, \boldsymbol{c_2}, \boldsymbol{r_2}, ..., \boldsymbol{c_R}, \boldsymbol{r_R}, t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_N\,\mathrm{d}\boldsymbol{c_{R+1}}...\mathrm{d}\boldsymbol{c_N}\,\mathrm{d}\boldsymbol{r_{R+1}}...\mathrm{d}\boldsymbol{r_N} \tag{2.9}$$

and we regain the single particle distribution function in the particular case when $R = 1$. This is because the probability of finding molecule number 1 in phase space element $\mathrm{d}\boldsymbol{c_1}\,\mathrm{d}\boldsymbol{r_1}$ at time t is $f_1(\boldsymbol{c_1}, \boldsymbol{r_1}, t)$ irrespective of the positions of the other $N - 1$ molecules. Since the molecules are indistinguishable, the number of molecules in the phase space element at time t is Nf_1, leading to

$$Nf_1 \equiv f_{sp}. \tag{2.10}$$

More details on the subject of velocity distribution functions is given in Chapter 3 of [5]. From now on in this thesis, simply f will be used when referring to the velocity distribution function and this will refer to the single particle distribution function in phase space.

2.2 Derivation of the Boltzmann Equation

2.2.1 The Liouville Equation

The mathematical statement for the conservation of the N–particle distribution function in 6N dimensional phase space is the best description of the evolution of a system of N particles. This mathematical statement is called the Liouville equation (see [34],[17],[36]). Unfortunately, the Liouville equation is not particularly useful in itself, since the description of a real gas flow in terms of the N–particle distribution function, f_N, is completely out of the question due to the massive number of independent variables. However, just as a hierarchy of reduced distribution functions is possible, see equation (2.9), so a hierarchy of equations, known as the BBGKY equations, may be obtained by repeated integration of the Liouville equation. The final equation in the hierarchy defines the single particle distribution function, f_1, and, by making the assumption of molecular chaos and using equation (2.10), we can derive an equation equivalent to the Boltzmann equation.

The derivation of the Boltzmann equation starting with the Liouville equation is, mathematically, the most rigorous approach in terms of defining the limits of validity of the equation. However, the physical interpretation of each term in the Boltzmann equation is more readily appreciated if it is derived from first principles. A simplified derivation from first principles will be shown in Section 2.2.3 based on the single particle distribution function in phase space for a simple, monatomic, single species gas. However, first we must spend some time looking at binary molecular collision theory, which will provide the basis for deriving the term in the Boltzmann equation which deals with the effects of molecular collisions on the distribution function.

2.2.2 Binary Molecular Collision Theory

In a dilute gas, i.e. one in which the mean free molecular path is significantly larger than a typical molecular cross section, molecular collisions are overwhelmingly likely to be binary collisions, involving just two molecules. It is possible to make rapid progress if we make the assumption that these collisions are elastic, i.e. there is no interchange of translational and internal energy. By writing expressions for the conservation of momentum and energy, we can derive relationships for the pre–collision and post–collision velocities of two colliding molecules. The binary collision process is shown in Figure 2.3.

Conservation of momentum implies that

$$m_1\boldsymbol{c_1} + m_2\boldsymbol{c_2} = m_1\boldsymbol{c_1^*} + m_2\boldsymbol{c_2}* = (m_1 + m_2)\boldsymbol{c_m} \tag{2.11}$$

where $\boldsymbol{c_m}$ is the velocity of the centre of mass of the molecules. Conservation of energy can be expressed by the equation

$$m_1\boldsymbol{c_1^2} + m_2\boldsymbol{c_2^2} = m_1\boldsymbol{c_1^{*2}} + m_2\boldsymbol{c_2^{*2}}. \tag{2.12}$$

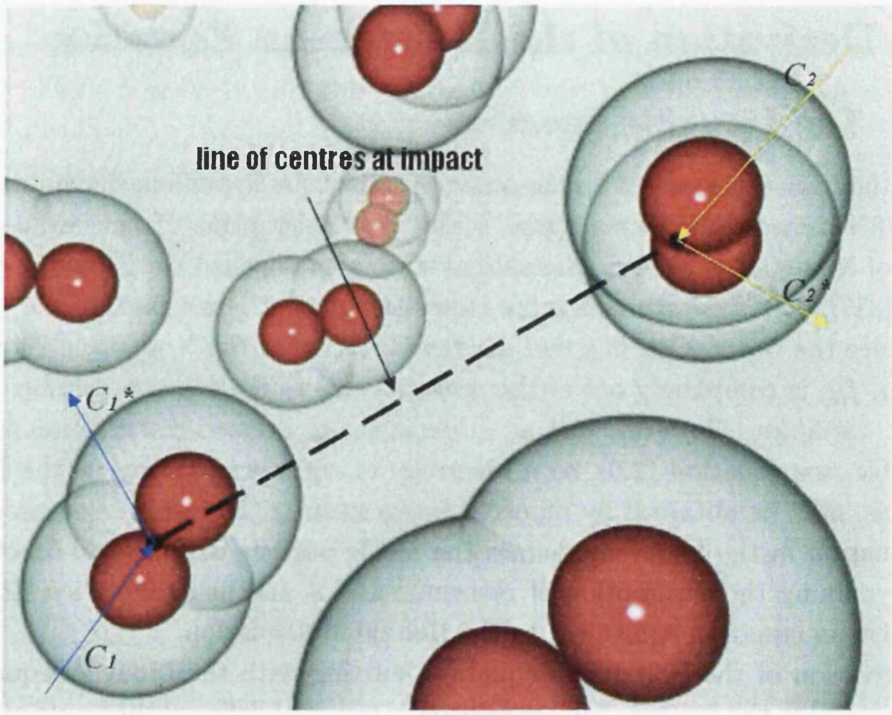


Figure 2.3: A Binary Molecular Collision

Equations (2.11) and (2.12) lead to the results

$$\begin{aligned} \boldsymbol{c}_1^* &= \boldsymbol{c}_m + \frac{m_2}{m_1 + m_2}\boldsymbol{c}_r^* \\ \boldsymbol{c}_2^* &= \boldsymbol{c}_m - \frac{m_1}{m_1 + m_2}\boldsymbol{c}_r^* \end{aligned} \tag{2.13}$$

where $\boldsymbol{c}_r^*$ is the post collision relative velocity, $\boldsymbol{c}_1^* - \boldsymbol{c}_2^*$, and

$$\begin{aligned} m_1\boldsymbol{c}_1^2 + m_2\boldsymbol{c}_2^2 &= (m_1 + m_2)\boldsymbol{c}_m^2 + m_r\boldsymbol{c}_r^2 \\ m_1\boldsymbol{c}_1^{*2} + m_2\boldsymbol{c}_2^{*2} &= (m_1 + m_2)\boldsymbol{c}_m^2 + m_r\boldsymbol{c}_r^{*2} \end{aligned} \tag{2.14}$$

where $m_r = m_1 m_2/(m_1 + m_2)$, the reduced mass. Comparison of equations (2.12) and (2.14) reveals that

$$|\boldsymbol{c}_r^*| = |\boldsymbol{c}_r|. \tag{2.15}$$

The significance of this result is that the determination of the post–collision velocities reduces to the calculation of the change in direction of the relative velocity vector.

The equations of motion of the two molecules in Figure 2.4 are

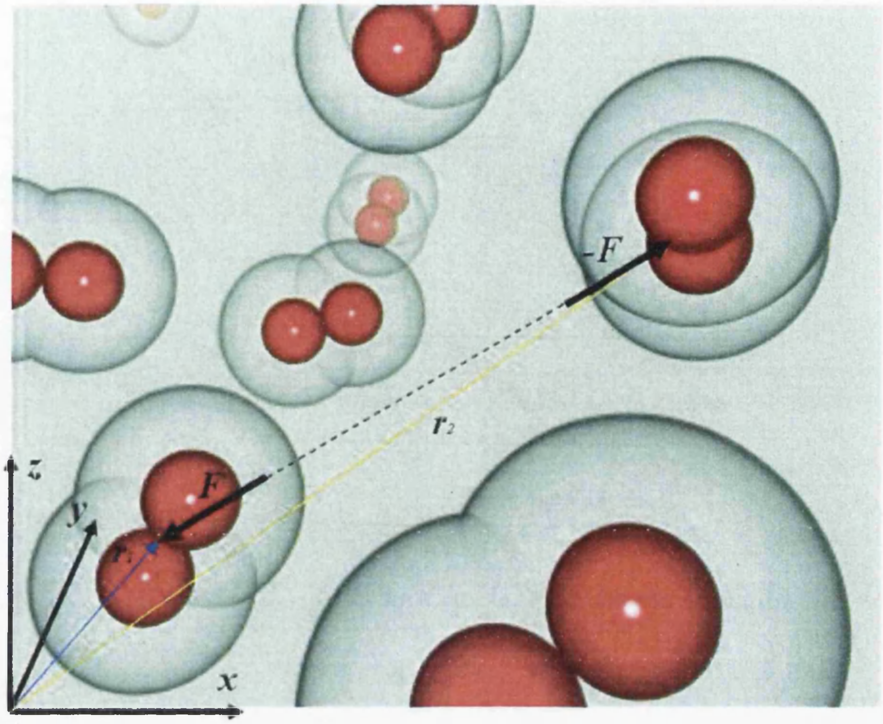


Figure 2.4: Intermolecular Force

$$
\begin{aligned}
m_1\ddot{\boldsymbol{r}}_1 &= \boldsymbol{F} \\
m_2\ddot{\boldsymbol{r}}_2 &= -\boldsymbol{F}
\end{aligned}
\tag{2.16}
$$

These lead to the equation

$$m_1 m_2(\ddot{\boldsymbol{r}}_1 - \ddot{\boldsymbol{r}}_2) = (m_1 + m_2)\boldsymbol{F}$$

and, therefore,

$$m_r\dot{\boldsymbol{c}}_r = \boldsymbol{F} \tag{2.17}$$

This implies that the change in direction, χ of the relative velocity vector, $\boldsymbol{c}_r$, is simply dependent on the intermolecular force model chosen. If we consider the binary collision in the centre of mass reference frame, as shown in Figure 2.5, we notice that just two impact parameters are required to completely specify the collision between two spherically symmetric molecules. The first of these impact parameters is the distance of closest approach of the undisturbed trajectories of the molecules in the centre of mass reference frame b. The plane in which the trajectories lie in the centre of mass reference frame is called the collision plane, and the second impact parameter is chosen as the angle ϵ between the collision plane and some reference plane. Note that the line of intersection of the collision and reference frames must be parallel to $\boldsymbol{c}_r$. If we consider the plane normal to $\boldsymbol{c}_r$ and containing O, we can

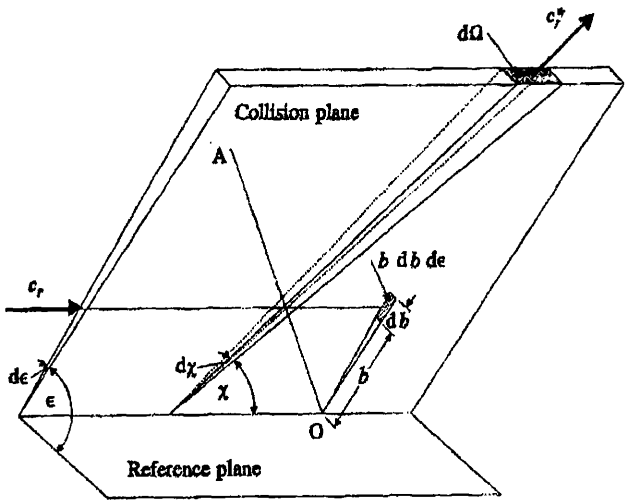


Figure 2.5: Impact Parameters (taken from Bird's *Molecular Gas Dynamics*)

define a differential cross section $\sigma\,\mathrm{d}\Omega$ for the collision, using the impact parameters b and ϵ, as

$$\sigma\,\mathrm{d}\Omega = b\,\mathrm{d}b\,\mathrm{d}\epsilon \tag{2.18}$$

where $\mathrm{d}\Omega$ is the solid unit angle about the vector $\boldsymbol{c}_r^*$. From Figure 2.5 it is apparent that $\mathrm{d}\Omega = sin\chi\,\mathrm{d}\chi\,\mathrm{d}\epsilon$, so that

$$\sigma = (b/\mathrm{sin}\chi)|\mathrm{d}b/\mathrm{d}\chi|. \tag{2.19}$$

Finally, the total collision cross section σ_T is defined by

$$\sigma_T = \int_0^{4\pi} \sigma d\Omega = 2\pi \int_0^{\pi} \sigma sin\chi \mathrm{d}\chi. \tag{2.20}$$

The determination of this integral is key to being able to describe the effect of molecular collisions in a gas in kinetic theory. It can be shown (see [5]) that, for many of the more realistic molecular models, this integral diverges and it is then necessary to introduce effective, or nominal, cross sections. However, for simpler models, such as the hard sphere model, this integral is straightforward to compute.

2.2.3 Derivation from First Principles

The most meaningful approach to a first principles derivation of the Boltzmann equation is to consider conservation of molecules in phase space. This is the same as considering class conservation in physical space. If the location and shape of an element in phase space does not vary in time, the rate of change of the number of class $\boldsymbol{c}$ molecules, N, in that element is given by

$$\frac{\partial N}{\partial t} = \frac{\partial}{\partial t}(nf)\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r} \tag{2.21}$$

where f is the single particle distribution function in phase space. The processes that contribute to this change are

(i) convection of molecules across the face of *dr* by the molecular velocity *c*

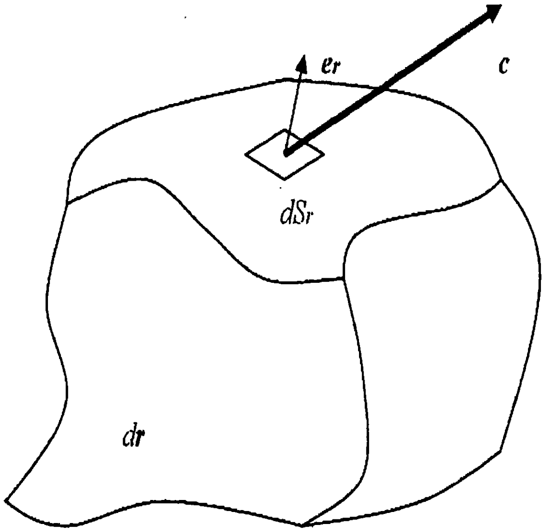


Figure 2.6: Convection of molecules across the face of a P-space element

The number of class $\boldsymbol{c}$ molecules in $d\boldsymbol{r}$ is $nfd\boldsymbol{c}d\boldsymbol{r}$. Therefore, the net inflow of molecules of this class is given by

$$-\int_{S_r} nf\boldsymbol{c}.\boldsymbol{e}_r\,\mathrm{d}S_r\mathrm{d}\boldsymbol{c}.$$

and applying the Divergence Theorem (see Appendix B), we deduce that this becomes

$$-\int_{\mathrm{d}\boldsymbol{r}} \nabla.(nf\boldsymbol{c})\,\mathrm{d}(\mathrm{d}\boldsymbol{r})\,\mathrm{d}\boldsymbol{c} = \nabla.(nf\boldsymbol{c})\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{c} = -\boldsymbol{c}.\frac{\partial(nf)}{\partial\boldsymbol{r}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r}$$

(ii) the 'convection' of molecules across the face of *dc* as the result of the external force per unit mass

In velocity space, convection under the influence of an external force per unit mass, $\boldsymbol{F}$, is completely analagous to convection in physical space due to a velocity vector, $\boldsymbol{c}$. We, therefore, have an expression for the flux of molecules across the surface of the v–space element that is completely analagous to equation (2.22) with $\boldsymbol{c}$ replaced by $\boldsymbol{F}$ and $\partial/\partial\boldsymbol{r}$ replaced by $\partial/\partial\boldsymbol{c}$, i.e.

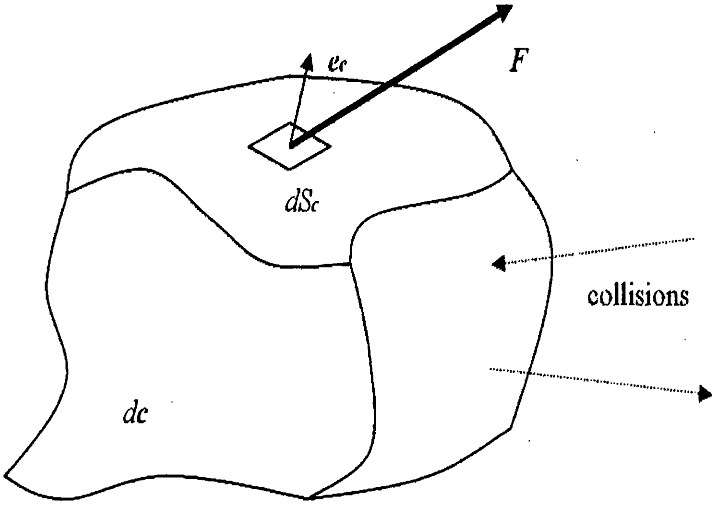


Figure 2.7: Convection / scattering of molecules across the face of a v-space element

$$-\boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}}\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r} \tag{2.22}$$

(iii) the scattering of molecules into and out of d$\boldsymbol{c}$ as a result of intermolecular collisions

If the gas is assumed to be dilute, the collisions are instantaneous events at fixed locations in physical space. Therefore, collisions only cause molecules to change their positions in velocity space. They remain at the same point in physical space and time. Also, all collisions are assumed to be binary.

From the work done in the previous section, it is possible to show that the number of collisions per unit time of a test class $\boldsymbol{c}$ molecule with class $\boldsymbol{c}_1$ molecules is given by

$$n\, f_1\, c_r\, \sigma\, \mathrm{d}\Omega\, \mathrm{d}\boldsymbol{c}_1. \tag{2.23}$$

Since the number of class $\boldsymbol{c}$ molecules in the phase space element is $nf\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{r}$, the number of $\boldsymbol{c}, \boldsymbol{c}_1 \rightarrow \boldsymbol{c}^*, \boldsymbol{c}_1^*$ collisions per unit time is given by

$$n^2\, f\, f_1\, c_r\, \sigma\, \mathrm{d}\Omega \mathrm{d}\boldsymbol{c}_1\, \mathrm{d}\boldsymbol{c}\, \mathrm{d}\boldsymbol{r} \tag{2.24}$$

where f denotes the value of f at $\boldsymbol{c}$ and f_1 denotes the value of f at $\boldsymbol{c}_1$. The existence of inverse collisions means that the number of class $\boldsymbol{c}^*, \boldsymbol{c}_1^* \rightarrow \boldsymbol{c}, \boldsymbol{c}_1$ collisions per unit time is given by

$$n^2\, f^*\, f_1^*\, c_r^*\, (\sigma\, \mathrm{d}\Omega)^*\, \mathrm{d}\boldsymbol{c}_1^*\, \mathrm{d}\boldsymbol{c}^*\, \mathrm{d}\boldsymbol{r} \tag{2.25}$$

in the phase space element $\mathrm{d}\boldsymbol{c}^*\,\mathrm{d}\boldsymbol{r}$ where f_1^* denotes the value of f at $\boldsymbol{c}_1^*$ and f^* denotes the value of f at $\boldsymbol{c}^*$. However, equation (2.15) shows that

$$|\boldsymbol{c}_r^*| = |\boldsymbol{c}_r|$$

and symmetry between the direct and inverse collisions means that there is a unit Jacobian for the transformation between the pre–collision and post–collision values of the product of the differential cross section and velocity space elements. This means that

$$|(\sigma\, d\Omega)\mathrm{d}\boldsymbol{c}_1\,\mathrm{d}\boldsymbol{c}| = |(\sigma \mathrm{d}\Omega)^*\,\mathrm{d}\boldsymbol{c}_1^*\,\mathrm{d}\boldsymbol{c}^*|. \tag{2.26}$$

Expression (2.25) then becomes

$$n^2\, f^*\, f_1^*\, c_r\, \sigma\, \mathrm{d}\,\Omega\, \mathrm{d}\boldsymbol{c}_1\, \mathrm{d}\boldsymbol{c}\, \mathrm{d}\boldsymbol{r}. \tag{2.27}$$

The final expression for process (iii) is, therefore,

$$\int_{-\infty}^{\infty}\int_0^{4\pi} n^2(f^*f_1^* - f f_1) c_r \sigma \mathrm{d}\Omega\, \mathrm{d}\boldsymbol{c}_1\, \mathrm{d}\boldsymbol{c}\, \mathrm{d}\boldsymbol{r}. \tag{2.28}$$

Assembling expressions (2.21),(2.22),(2.22) and (2.28) gives

$$\frac{\partial}{\partial t}(nf) + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}} = \int_{-\infty}^{\infty}\int_0^{4\pi} n^2(f^*f_1^* - f f_1) c_r \sigma\, \mathrm{d}\Omega\, \mathrm{d}\boldsymbol{c}_1 \tag{2.29}$$

which is the Boltzmann equation for a monatomic, single species gas under the influence of an external force field, $\boldsymbol{F}$.

2.3 The BGK Approximation

An expression for the distribution function of a gas in thermodynamic equilibrium was established late in the 19th by Maxwell [67], and later confirmed by Boltzmann himself. This has now become known as the Maxwell–Boltzmann equilibrium distribution function or simply the Maxwellian distribution function. The form of this distribution function is,

$$f_0(\boldsymbol{c}) = (\frac{\beta^3}{\pi^{3/2}})\exp(-\beta^2(\boldsymbol{c}-\boldsymbol{c_0})^2) \tag{2.30}$$

in 3 dimensions, where $\boldsymbol{c_0}$ is the bulk velocity of the flow and $\beta = (2RT)^{-1/2} = \sqrt{m/(2kT)}$. R is the gas constant, T is the gas temperature measured in Kelvin, m is the molecular mass and k is the Boltzmann constant (1.380 650524 x10^{-23} Joules/Kelvin). A helpful derivation of the Maxwellian distribution function is provided by Vincenti and Kruger in [48]. It is important to note that f_0 is a symmetric function, i.e. $f(c_x - c_{x0}, c_y - c_{y0}, c_z - c_{z0}) = f(-(c_x - c_{x0}), c_y - c_{y0}, c_z - c_{z0}) = f(c_x - c_{x0}, -(c_y - c_{y0}), c_z - c_{z0}) etc.$ This is why it is possible to define a single static pressure for gases in equilibrium, since $p_x = p_y = p_z$ without ambiguity (see section 2.4).

Simplified versions of the full Boltzmann equation exist in which the right–hand side term is modelled rather than expressed explicitly, as in equation (2.29). The best known model equation due to Bhatnager, Gross and Krook [69] is called the BGK equation. The assumption is made that the effect of molecular collisions, in a non-equilibrium gas, is to force it back to equilibrium. This is indicated graphically in Figure 2.8.

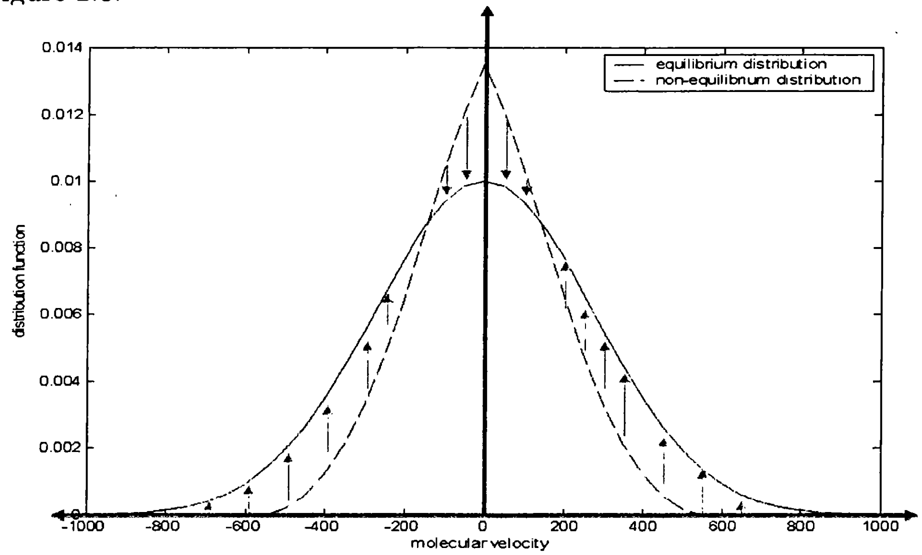


Figure 2.8: Effect of the BGK Collision Term on the Distribution Function

The integrals on the right–hand side of equation (2.29) mathematically describe this process and can, therefore, be replaced by the term $\nu(\boldsymbol{c})((nf_0) - (nf))$, where

n is the molecular number density, $\nu(\boldsymbol{r},t)$ is a term proportional to the molecular collision frequency and f_0 is the local Maxwellian equilibrium distribution function. The BGK equation may therefore be written as

$$\frac{\partial}{\partial t}(nf) + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} + \boldsymbol{F}.\frac{\partial(nf)}{\partial \boldsymbol{c}} = \nu(\boldsymbol{r},t)((nf_0) - (nf)). \tag{2.31}$$

The inclusion of the equilibrium distribution function means that the BGK equation is still a non–linear integro–differential equation, because f_0 is a function of the stream velocity, $\boldsymbol{c_0}$, and the temperature, T, which are obtained by taking integrals over f. However, computationally, the BGK term is significantly less demanding than the full right–hand side term.

2.4 Moments of the Distribution Function

The form of the distribution function at any point in physical space is not particularly useful in itself, but knowledge of the distribution function allows us to calculate more familiar gas properties, such as density, ρ, temperature, T, pressure, p, and bulk velocity, $\boldsymbol{v}$, by taking moments of the distribution function.

At each point in physical space, the mean value of any molecular quantity, Q, is defined as

$$\overline{Q} = \int_{-\infty}^{\infty} Q f(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} \tag{2.32}$$

By setting Q to the appropriate molecular quantity, we can obtain the macroscopic properties as follows;

-density ρ: $Q = nm$ where m is the molecular mass

-bulk velocity $\boldsymbol{v_i}$: $Q = \boldsymbol{c_i}$

-static pressure $\boldsymbol{p_i}$: $Q = nm\overline{\boldsymbol{c}'_i\boldsymbol{c}'_j}$ where $\boldsymbol{c}'_i = \boldsymbol{c_i} - \boldsymbol{v_i}$(thermal / peculiar velocity)

This definition of static pressure is clearly a vector quantity. The vectorial nature of pressure is a true characteristic of a non–equilibrium gas. In the more familiar world of contimuum gas dynamics, an equilibrium gas is always assumed resulting in a pressure vector that has each directional component equal. This is the subtle fact that allows us to specify a single scalar quantity for static pressure. We are more accustomed to dealing with pressure as a scalar and so the static pressures specified in this work will be taken as the mean of the Cartesian components.
To determine the temperature, we simply use the definition of kinetic temperature,

$$T_k = \frac{p}{R\rho}. \tag{2.33}$$

It is also possible to define non–standard macroscopic gas properties that may be of interest in certain applications. The algorithm for performing these moment integrals will be detailed in the following chapter.

Chapter 3

The Solution Algorithm

In this chapter we will look at the solution approach that has been adopted to solve the collisionless, BGK and full Boltzmann equations. The scheme is based on a two step discontinuous Taylor–Galerkin methodology. The continuous Taylor–Galerkin procedure is a method that has been widely used previously ([25],[53],[73],[96]).

3.1 Discretisation of Phase Space

As we discovered in the previous chapter, the dependent variable of the Boltzmann equation, the distribution function, is defined over both physical and velocity space, known collectively as phase space. We must therefore adopt a suitable discretisation procedure for both of these domains.

3.1.1 Physical Space

In this thesis, we restrict ourselves to problems in two physical space dimensions. The two dimensional p–space domain, Ω_r, is discretised into an unstructured assembly of discontinuous, linear, triangular elements. The unstructured nature of the discretisation allows meshing around complex geometries to take place and the discontinuous nature of the elements has a number of benefits, especially for hypersonic applications. These benefits include excellent shock capturing properties, this will be demonstrated in section 4.1.6, allowing the algorithm to be parallelised relatively easily, explicit local conservativity and the potential for variable order elements, i.e. neighbouring elements need not be of the same order. The major disadvantage of the discontinuous elements is a significant increase in the memory requirement. However, the ease of parallelisation offsets this disadvantage to some degree.

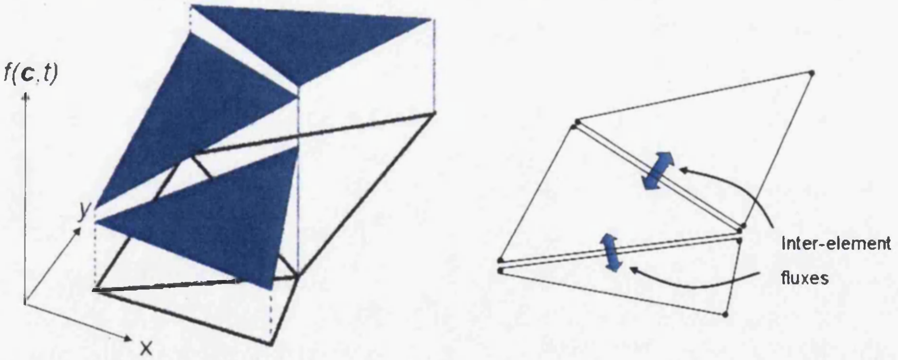


Figure 3.1: Discontinuous representation of the solution

It is clear from Figures 3.1 and 3.2 that the elements do not share nodes. They must therefore communicate by means of inter–element fluxes. Details of the construction of these fluxes will be provided in subsequent sections.

The nodes associated with an individual element will be referred to as element nodes or discontinuous nodes. It is important to make a distinction between these nodes

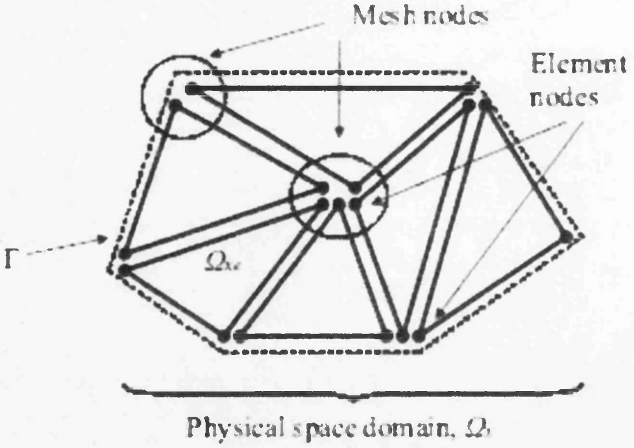


Figure 3.2: Physical Space Discretisation

and the mesh nodes used for the output of results, which are comprised of a number of element nodes which meet at a point as shown in Figure 3.2. The solution at these mesh nodes is constructed from a weighted average based on associated element size.

3.1.2 Velocity Space

The two dimensional v–space domain, Ω_c, is, in principle, infinite in extent. However, we must place a finite limit on the radial extent of the domian, which in real, physical terms means placing a limit on the maximum possible speed of a molecule. We are, thereby, making the assumption that any molecules travelling faster than this critical speed have negligible impact on the bulk flow properties. Gauging where this limit should be placed is critical for ensuring a balance between accurate results and computational efficiency. A standard rule of thumb for the limit is that the maximum speed in v–space should be at least several times the mean thermal molecular velocity [5]. The thermal velocity is defined as the component of molecular velocity relative to the fluid velocity.

An order of magnitude estimation for the value of the mean thermal molecular velocity can be calculated using some simple kinetic theory results. In section 2.4, we noted that we can regain the macroscopic gas flow properties by taking moments of the distribution function. This moment calculation for the non–equilibrium static pressure in a given direction was shown to be

$$\boldsymbol{p_i} = \int_{-\infty}^{\infty} nm\overline{\boldsymbol{c}'_i\boldsymbol{c}'_j} f(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}. \tag{3.1}$$

Taking an average over three spatial dimensions,

$$p = 1/3\,\rho\bar{c'^2}. \tag{3.2}$$

so that the rms thermal molecular velocity may be written as

$$\sqrt{\bar{c'^2}} = \sqrt{3RT}. \tag{3.3}$$

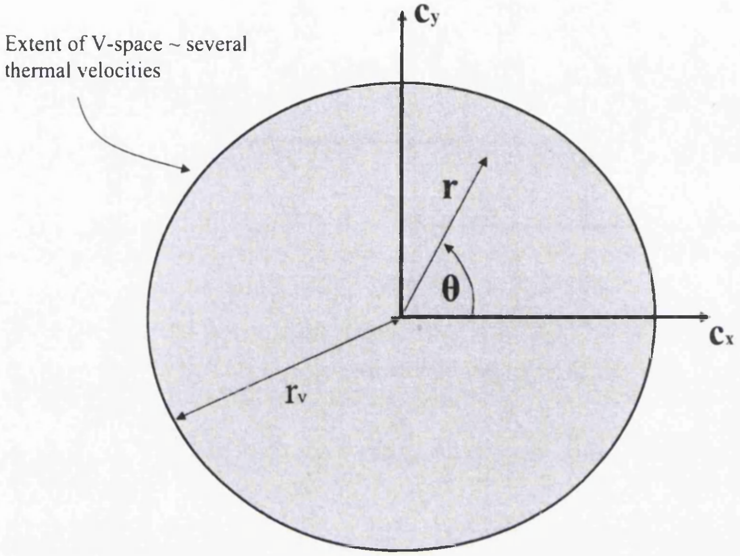


Figure 3.3: Velocity Space Domain

Two approaches for the discretisation of v–space will be detailed in the following sections and results using both methods shown in the following chapter.

V–Space Quads

The most straightforward discretisation approach is to use standard structured, continuous, quadrilateral, linear elements to provide a mesh for the circular domain, Ω_c as shown in Figure 3.4. The major advantage of this approach is its simplicity and the ease in which integrals over the full domain or a sector of the domain may be performed. The importance of this will become apparent in sections 3.5.3 and 3.3. The significant disadvantage of this approach in terms of discretisation, is the relatively large number of nodes required to achieve a given level of accuracy in computing the integrals.

Spectral V–space

As the v–space domain contains no internal geometries, it can be discretised as a single high order (spectral) element. This is advantageous for efficient integration over the domain. However, in order to apply a high order discretisation, the domain must be mapped from Cartesian or polar coordinates in real space as shown in Figure 3.3 into a standard quadrilateral element in the (η, ζ) plane as shown in Figure 3.5. The mapping is achieved using the transformation

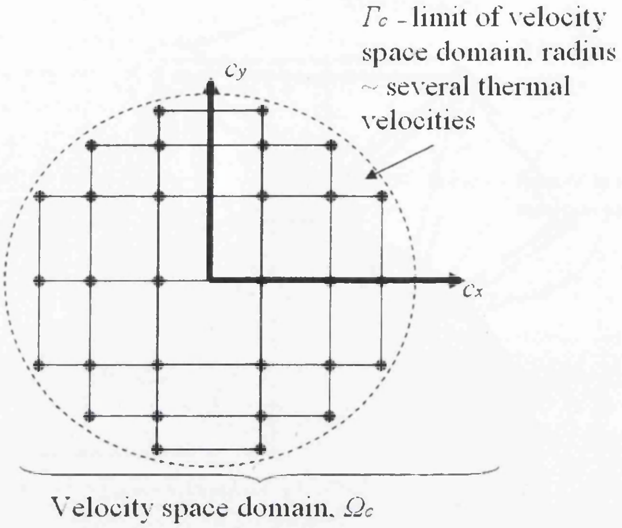

Γc - limit of velocity space domain, radius ~ several thermal velocities
cy
cx
Velocity space domain, Ωc

Figure 3.4: Velocity Space Domain Discretisation using Linear Quads

$$\begin{aligned}\eta &= \frac{2r}{r_v} - 1 \\ \zeta &= \frac{\theta}{\pi}.\end{aligned} \tag{3.4}$$

where (r, θ) are polar coordinates in real v–space and r_v is the radius of the v–space domain, i.e. the maximum molecular speed. A high order quadrature method is then applied to the element. In the η direction, a Lobatto quadrature is applied whereas in the ζ direction a constant spacing / constant weighting discretisation is applied. This results in a rotationally symmetric distribution of sampling points with no preferred radial direction, when the points are mapped back into real space.

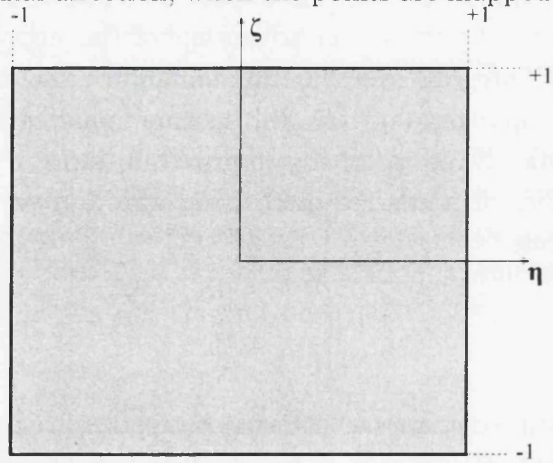


Figure 3.5: A Standard Quadrilateral Element in the (η, ζ) Plane

The coordinates of the quadrature points and the associated weightings in the (η, ζ)

plane are shown in Figure 3.6 for a (20 x 20) discretisation. If these points are then mapped back into real space, the (u, v) plane, the corresponding coordinates and weights are as shown in Figure 3.7.

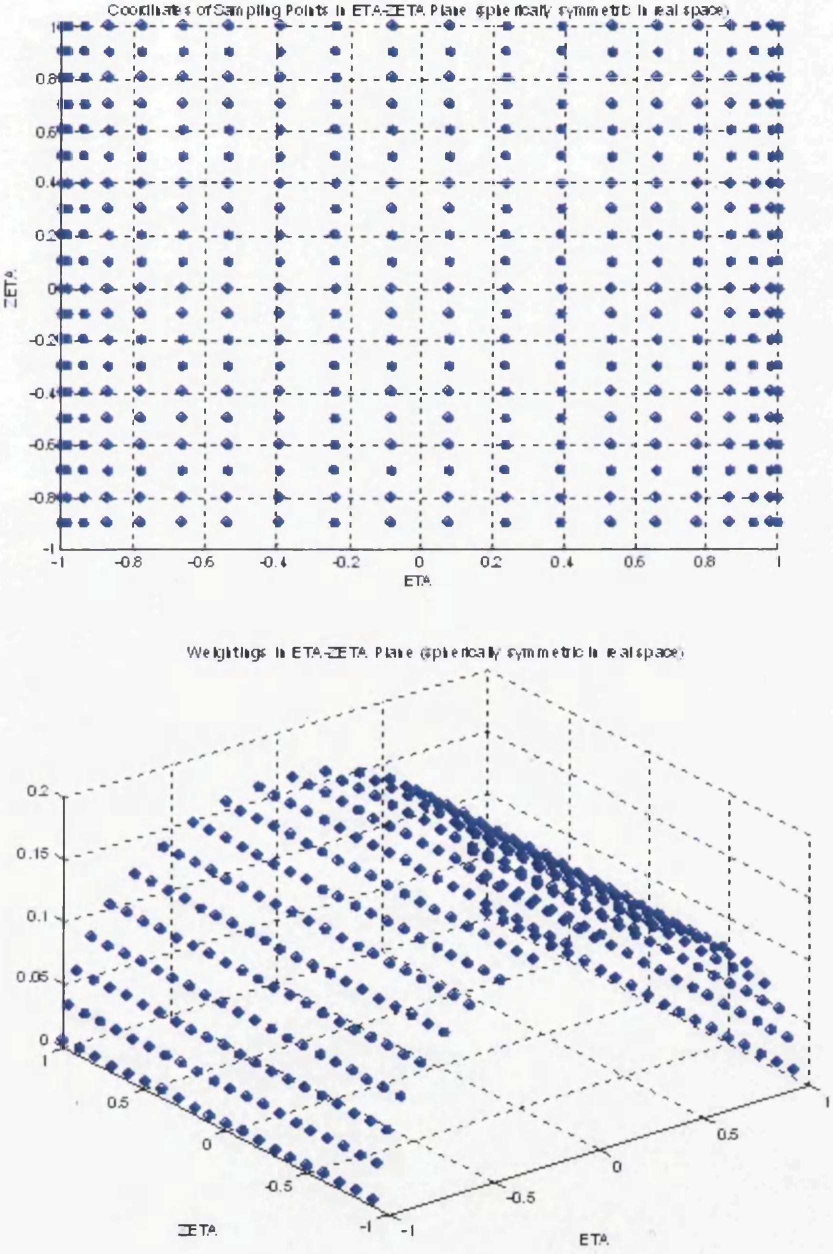


Figure 3.6: Quadrature Coordinates and Weights in the (η, ζ) Plane

One of the dangers in approximating a function using spectral methods is the Runge phenomenon [100] which can occur when approximating functions using high order polynomials. However, it is the accuracy of the integrals of the distribution function over v–space that is of concern here. This is because, ultimately, it is the

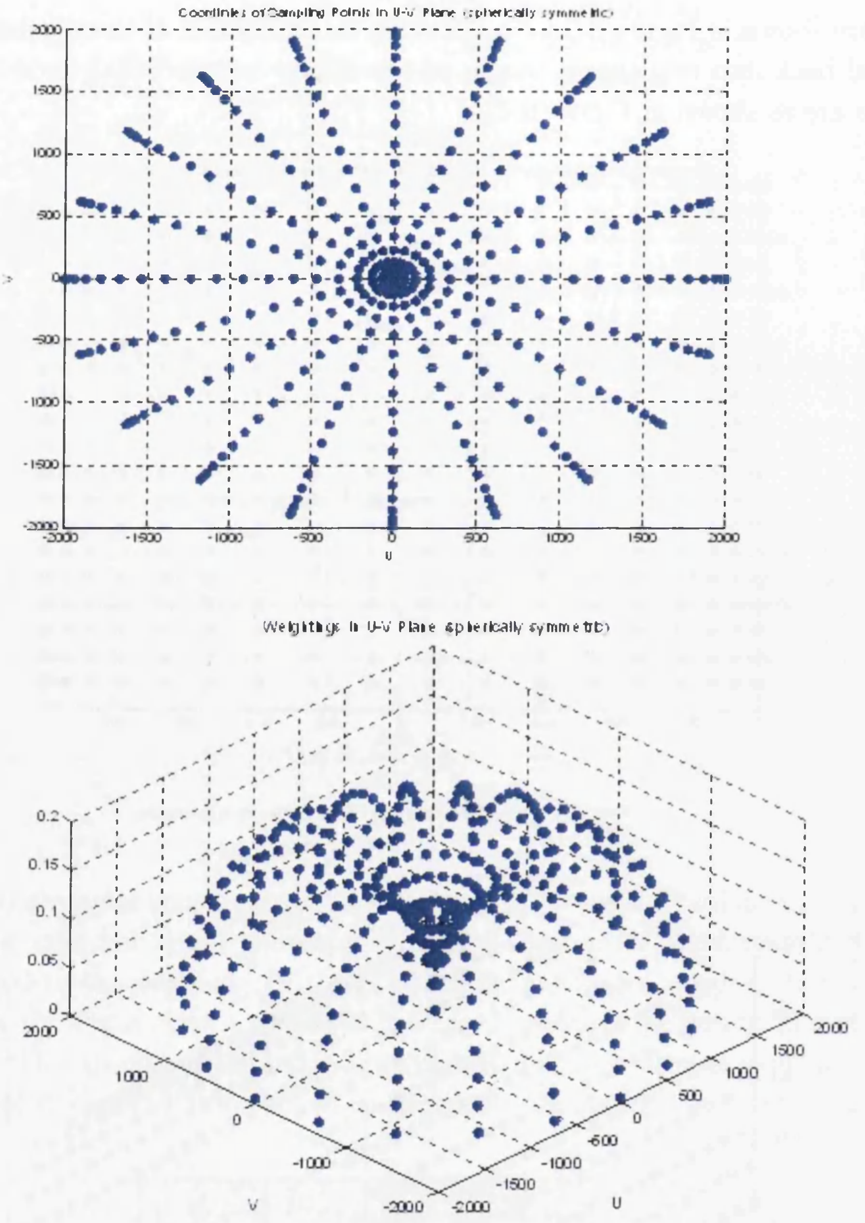


Figure 3.7: Quadrature Coordinates and Weights in the (u, v) Plane

macroscopic variables that are of interest which are computed by taking moments of the distribution function (see section 2.4). Testing of this method for typical forms of the distribution function showed that this spectral approach was perfectly capable of accurately performing the necessary integrals. A more rigorous mathematical analysis of quadrature methods for circular domains is contained in [62].

3.2 A Two Step Discontinuous Taylor–Galerkin Procedure

3.2.1 The Taylor–Galerkin Method for the Scalar Convection Equation

The two step Taylor–Galerkin method is the finite element equivalent of the Lax–Wendroff method in finite difference schemes. A two step version of this method has been adopted to tackle the Boltzmann equation. It will become apparent that underpinning the Boltzmann solution approach is a relatively simple scalar convection algorithm. A single step Taylor–Galerkin algorithm for the scalar convection equation is presented in this section for completeness and to demonstrate the basis of the second order accuracy in physical space and time.

The scalar convection equation for a scalar variable, U, can be written as

$$\frac{\partial U}{\partial t} + \boldsymbol{c}\frac{\partial U}{\partial \boldsymbol{r}} = 0 \tag{3.5}$$

where $\boldsymbol{c}$ is the fixed convection velocity. Consider a Taylor series expansion for U in time of the form

$$U^{m+1} = U^m + \Delta t \frac{\partial U}{\partial t}\bigg|^m + \frac{\Delta t^2}{2}\frac{\partial^2 U}{\partial t^2}\bigg|^m + O(\Delta t^3) \tag{3.6}$$

where the superscript, m denotes an evaluation at time $t = t_m$ and the timestep $\Delta t = t_{m+1} - t_m$. We can remove the time derivatives from (3.6) by substituting from the governing equation (3.5) as

$$\frac{\partial U}{\partial t} = -\boldsymbol{c}\frac{\partial}{\partial \boldsymbol{r}}U = -\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{r}} \tag{3.7}$$

where $F = \boldsymbol{c}U$. It is often more convenient to write expressions such as (3.7) in the indexed format,

$$\frac{\partial U}{\partial t} = -c_i\frac{\partial}{\partial r_i}U = -\frac{\partial F_i}{\partial r_i} \tag{3.8}$$

where $F = c_i U$ and the Einstein summation convention is implied. Similarly, the second time derivative is

$$\frac{\partial^2 U}{\partial t^2} = \frac{\partial}{\partial r_i}(c_i\frac{\partial F_j}{\partial r_j}) \tag{3.9}$$

Substituting into equation (3.6), it follows that

$$U^{m+1} - U^m = \Delta U = -\Delta t\frac{\partial}{\partial r_i}\left[F_i^m - \frac{\Delta t}{2}c_i\frac{\partial F_j}{\partial r_j}\bigg|^m\right] \tag{3.10}$$

ignoring the $O(\Delta t^3)$ terms.

3.2.2 Notation

We must be able to specify our discretised unknown, nf in terms of time, space and velocity. Therefore, we will employ the notation $(nf)^m_{\boldsymbol{r},\boldsymbol{c}}$ to mean the value of nf at timestep m, spatial coordinate $\boldsymbol{r}$ and velocity coordinate $\boldsymbol{c}$. This subscript/superscript convention will also be adopted for other variables.

3.2.3 Solution Algorithm for the Collisionless Boltzmann Equation

The collisionless Boltzmann equation, in the absence of any external force fields, is a good place to start in developing an algorithm for solution of the full Boltzmann equation since the third term on the left–hand side of the equation, the force field term, and the right–hand side term, the collision term, are both set to zero. This results in a considerable simplification. The collisionless Boltzmann equation, in the absence of external force fields, has the form

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} = 0. \tag{3.11}$$

It is apparent that the form of this equation is not disimilar to the scalar convection equation that was studied in section 3.2.1, except that now the dependent variable, nf, is a function not just of space and time, but also of velocity, $\boldsymbol{c}$. However, if we fix our position in v–space, i.e. hold $\boldsymbol{c}$ constant, then the form of the collisionless Boltzmann equation is identical to that of the scalar convection equation. This allows us to develop a collisionless Boltzmann algorithm based on the scalar convection algorithm described in section 3.2.1, by looping over the nodes or sampling points in v–space and applying an adapted scalar convection algorithm at each v–space node independently. The convection velocity used is given by the coordinate of the v–space node in question. A Galerkin weighted residual method is used together with the approximation

$$(nf)_i^{m+\frac{1}{2}} = (nf)_i^m - \frac{\Delta t}{2} c_i \frac{\partial(nf)_j}{\partial r_j}\Big|^m \tag{3.12}$$

to obtain the increment $\Delta(nf)$ in a two-step manner.

First Step

The first step is equivalent to computing the term inside the bracket on the right–hand side of equation (3.10). A piecewise constant increment $\Delta(nf)_{re,\boldsymbol{c}}$ is computed on each physical space element according to

$$\Delta(nf)_{re,\boldsymbol{c}} = -\frac{\Delta t}{2} F^m_{ik} \frac{\partial N_k}{\partial r_i}\Big]_{re,\boldsymbol{c}} \tag{3.13}$$

where the summation k extends over the three nodes of element re, Δt is the global timestep determined by the Courant stability condition

$$\Delta t = \frac{0.5h_{min}}{|\boldsymbol{v}|_{max}} \tag{3.14}$$

with h_{min} representing some minimum characteristic element size in the physical space mesh and $|\boldsymbol{v}|_{max}$ denoting the maximum velocity in the velocity space mesh. Here, N_k is the standard, piecewise, linear finite element shape function associated with node k in physical space and

$$F^m_{ik,\boldsymbol{c}} = F_i\left((nf)^m_{k,\boldsymbol{c}}\right) = \boldsymbol{c}(nf)^m_{k,\boldsymbol{c}} \tag{3.15}$$

The element fluxes at the half timestep are then approximated by the piecewise linear discontinuous representation

$$F_i^{m+\frac{1}{2}}]_{re,\boldsymbol{c}} = F_i((nf)^m_{k,\boldsymbol{c}} + \Delta(nf)_{re,\boldsymbol{c}})N_k \tag{3.16}$$

Second Step

The second step is the discontinuous Galerkin weighted residual approximation to equation (3.6), given the computation of the half–timestep fluxes in step 1. A piecewise linear approximation for $\Delta(nf)$ on each physical space element is assumed which is discontinuous at the element edges. The element nodal values of the solution increment over the complete timestep are determined according to

$$M_L]_{re}\Delta(nf)_{k,\boldsymbol{c}} = \Delta t\int_{\Gamma_{re}} F^{m+\frac{1}{2}}_{n,\boldsymbol{c}} N_k \mathrm{d}\Gamma_{re} - \Delta t\int_{\Omega_{re}} F^{m+\frac{1}{2}}_{ik,\boldsymbol{c}}\frac{\partial N_k}{\partial r_i}\mathrm{d}\Omega_{re} \tag{3.17}$$

where $M_L]_{re}$ is the standard, lumped, 3x3 physical space element mass matrix, $F^{m+\frac{1}{2}}_{n,\boldsymbol{c}}$ denotes the normal component of the upstream flux at the physical space element edges for a velocity of $\boldsymbol{c}$ (this is the inter–element flux mentioned in section 3.1.1 which will be dependent on boundary conditions if the edge is at a domain boundary), Γ_{re} is the physical space element boundary and Ω_{re} is the physical space element domain.

For inter–element edges, the direction of the flux across the edge must be calculated based on the convection velocity determined by the velocity space mesh node under consideration. If the flux is 'into the element', the integral in the first term on the RHS of 3.17 is given a value based on the corresponding upstream element edge flux, i.e. with reference to Figure 3.8, $F^{m+\frac{1}{2}}_{n,\boldsymbol{c}} = \frac{1}{2}\boldsymbol{c}.\boldsymbol{n}((nf)_1^{m+\frac{1}{2}} + (nf)_2^{m+\frac{1}{2}})$. If the flux is 'out of the element', the same term is given a negative value based on the normal edge flux at that edge in the element. This ensures the local conservativeness of the scheme, since the sum of the fluxes into the element, minus the sum of the fluxes out of the element, will give the mass increase within the element.

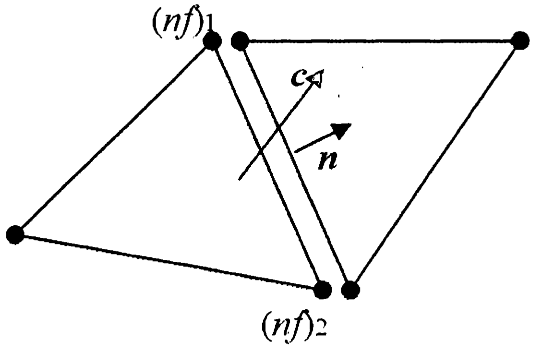


Figure 3.8: Construction of the Inter–element Flux

3.3 Moments of the Distribution Function

The method of taking moments of the distribution function, as described in section 2.4, again depends on the nature of the v–space discretisation. The integral that we are computing, however, is always

$$\overline{Q} = \int_{-\infty}^{\infty} Qf(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} \tag{3.18}$$

3.3.1 Moments using V-Space Quads

In the case where the v–space has been discretised using linear quads, equation (3.18) may be performed as

$$\overline{Q} = \frac{\sum_{e=1}^{N} Q[nf]_e A_e}{\sum_{e=1}^{N}[nf]_e A_e} \tag{3.19}$$

where N is the number of velocity space elements,$[nf]_e$ is the mean value of (nf) for the four element nodes and A_e is the area of the v–space element.

3.3.2 Moments using Spectral V-Space

In the case where the v–space is discretised using the spectral method, we must compute the integral in equation (3.19) by transforming the coordinate system from the real v–space coordinates to the (η, ζ) plane. Moving from Cartesian to polar coordinates gives us

$$\begin{aligned} u = r\cos\theta \quad &\Rightarrow \quad \frac{\partial u}{\partial r} = \cos\theta, \frac{\partial u}{\partial \theta} = -r\sin\theta \\ v = r\sin\theta \quad &\Rightarrow \quad \frac{\partial v}{\partial r} = \sin\theta, \frac{\partial v}{\partial \theta} = r\cos\theta. \end{aligned} \tag{3.20}$$

The Jacobian, J, of the transformation from Cartesian to polar coordinates is therefore

$$|J| = \begin{vmatrix} \frac{\partial u}{\partial r} & \frac{\partial u}{\partial \theta} \\ \frac{\partial v}{\partial r} & \frac{\partial v}{\partial \theta} \end{vmatrix} = r\cos^2\theta + r\sin^2\theta = r \tag{3.21}$$

so that

$$\overline{Q} = \int_{-\infty}^{\infty} Qf(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c} \Rightarrow \int_{-\pi}^{+\pi}\int_{0}^{r_v} Qf(r,\theta)\,r\,\mathrm{d}r\,\mathrm{d}\theta. \tag{3.22}$$

The mapping from the real polar v–space to the (η, ζ) plane gives

$$\begin{aligned} r = \frac{r_v}{2}(\eta+1) \quad &\Rightarrow \quad \frac{\partial r}{\partial \eta} = \frac{r_v}{2}, \frac{\partial r}{\partial \zeta} = 0 \\ \theta = \zeta\pi \quad &\Rightarrow \quad \frac{\partial \theta}{\partial \eta} = 0, \frac{\partial \theta}{\partial \zeta} = \pi \end{aligned} \tag{3.23}$$

so that the Jacobian, J, of the transformation is

$$|J| = \begin{vmatrix} \dfrac{\partial r}{\partial \eta} & \dfrac{\partial r}{\partial \zeta} \\ \dfrac{\partial \theta}{\partial \eta} & \dfrac{\partial \theta}{\partial \zeta} \end{vmatrix} = \begin{vmatrix} \dfrac{r_v}{2} & 0 \\ 0 & \pi \end{vmatrix} = \frac{\pi r_v}{2}. \tag{3.24}$$

We can rewrite equation (3.22) as

$$\overline{Q} = \int_{-1}^{+1} \int_{-1}^{+1} Qf(\eta, \zeta) \frac{\pi r_v^2}{4} (\eta + 1) \, \mathrm{d}\eta \, \mathrm{d}\zeta \tag{3.25}$$

so that the integral in equation (3.18) may be evaluated as

$$\overline{Q} = \frac{\sum_{i=1}^{N} w_i Q(nf)_i |J|}{\sum_{i=1}^{N} w_i (nf)_i |J|} \tag{3.26}$$

where the summations are over all the v–space sampling points in the discretisation, w_i is the weighting associated with the point, as described in section 3.1.2, and $|J| = \pi r_v^2 (\eta_i + 1)/4$.

3.4 Inclusion of the Collision Term

3.4.1 The BGK Collision Term

The BGK equation in the absence of an external force field may be written as

$$\frac{\partial(nf)}{\partial t} + \boldsymbol{c}.\frac{\partial(nf)}{\partial \boldsymbol{r}} = \nu(\boldsymbol{r},t)((nf_0) - (nf)). \tag{3.27}$$

The additional BGK term can be treated as a source term, with $\nu(\boldsymbol{r},t)$ computed as

$$\nu(\boldsymbol{r},t) \propto \int_{-\infty}^{+\infty} \sigma_T |\boldsymbol{c} - \boldsymbol{c_1}| f \, d\boldsymbol{c_1} \tag{3.28}$$

where σ_T is the total collision cross section defined in section 2.2.2. If we make the assumption of hard sphere molecules, we obtain the simplest possible expression for the total collision cross section, $\sigma_T = \pi d^2$, where d is the molecular diameter. The Maxwellian distribution function, f_0, is computed as in equation (2.30).

In the two step procedure described for the collisionless Boltzmann equation (3.12) must now be modified as

$$(nf)_i^{m+\frac{1}{2}} = (nf)_i^m + (\frac{\Delta t}{2})Q^m - \frac{\Delta t}{2} c_i \frac{\partial(nf)_j}{\partial r_j}\Big|^m \tag{3.29}$$

This means that the first step, formerly equation (3.13), is now written as

$$\Delta(nf)_{re,\boldsymbol{c}} = \frac{\Delta t}{2} \Sigma Q^m_{k,\boldsymbol{c}} N_k - \frac{\Delta t}{2} F^m_{ik} \frac{\partial N_k}{\partial r_i}\Big]_{re,\boldsymbol{c}} \tag{3.30}$$

and the second step, formerly equation (3.17), is now written as

$$M_L]_{re}\Delta(nf)_{k,\boldsymbol{c}} = \Delta t M_L]_{re} Q^{m+\frac{1}{2}} + \Delta t \int_{\Gamma_{re}} F_{n,\boldsymbol{c}}^{m+\frac{1}{2}} N_k \mathrm{d}\Gamma_{re} - \Delta t \int_{\Omega_{re}} F_{ik,\boldsymbol{c}}^{m+\frac{1}{2}} \frac{\partial N_k}{\partial r_i} \mathrm{d}\Omega_{re,\boldsymbol{c}} \tag{3.31}$$

Here $Q^m = \nu((nf_0) - (nf)^m)$ and $\Sigma Q^m_{k,\boldsymbol{c}} = 1/3(Q^m_{1,\boldsymbol{c}} + Q^m_{2,\boldsymbol{c}} + Q^m_{3,\boldsymbol{c}})$. In the BGK formulation shown here, the term ν can be regarded as a collision frequency term and governs the rate at which the distribution function is restored to equilibrium. The form of the BGK collision term is such that the distribution function will be restored to equilibrium in a timescale,

$$\tau = O(\frac{1}{\nu}). \tag{3.32}$$

This places a further restriction on the allowable timestep size, in addition to the Courant condition, equation (3.14), that

$$\Delta t < \frac{1}{\nu}. \tag{3.33}$$

We therefore have two limits on the allowable timestep size. The Courant limit, condition (3.14), is fixed by the mesh geometries and is a function of advection instability, whilst the BGK limit, condition (3.33), varies as the solution proceeds through time since it is dependent on the molecular collision frequency which, in turn, is dependent on pressure and temperature. This has the obvious effect of increasing the computational demand as density, and hence, pressure increase, since the timestep size must reduce.

It should also be clear that the BGK approximation provides no guarantee that the integral of the distribution function over v–space is preserved as it is forced to equilibrium. This means that we have lost the local conservativity that was ensured in the collisionless Boltzmann algorithm. This means that solutions based on the BGK approximation are only valid for flows that experience small perturbations from the Maxwellian distribution function. In real, practical terms, this means that the BGK solver will only be able to handle flows with up to Mach 2 strength shocks. This will be shown in the following chapter.

3.4.2 The Full Right–Hand Side Collision Term for Hard Sphere Molecules

The full Boltzmann equation right–hand side is dealt with in a very similar way to the BGK right–hand side in terms of the algorithmic steps. In fact, the two step procedure outlined in section 3.4.1 can be used. However, the source term, Q must now be modified to

$$Q = \int_{-\infty}^{\infty} \int_{0}^{4\pi} n^2 (f^* f_1^* - f f_1) c_r \sigma \mathrm{d}\Omega \, \mathrm{d}\, \boldsymbol{c}_1 \tag{3.34}$$

This equation holds in three dimensions. In the three dimensional case, Bird [5] recognises that for the simplest molecular interaction model, that of hard sphere molecules, $\int_0^{4\pi} \sigma \, \mathrm{d}\Omega$ reduces to πd^2, which is termed the total collision cross section, where d is the molecular diameter. The number density, n, is independent of molecular velocity and can therefore be brought outside the integral, allowing equation 3.34 to be rewritten in two dimensions as

$$Q = \pi n^2 d^2 \int_{-\infty}^{\infty} (f^* f_1^* - f f_1) c_r \mathrm{d}\, \boldsymbol{c}_1 \tag{3.35}$$

Here all the terms take on the same meanings as described in the Boltzmann equation derivation contained in section 2.2.3.

It is clear that computing this full source term will be significantly more expensive in terms of computation time in comparison with the BGK simplification. This is because for every sampling point in velocity space, collisions with molecules travelling with velocities at every other point in velocity space must be considered and the post collision velocities calculated, based on the binary collision theory of section 2.2.2.

3.4.3 The Full Right–Hand Side Collision Term for Variable Soft Sphere Molecules

Unfortunately, the hard sphere molecular interaction model is overly simplistic. Koura and Matsumoto ([65],[66]) suggest a modified molecular collision model called the variable soft sphere (VSS) model. In this model, the effective collision diameter, d varies as

$$d = d_{ref}(\frac{c_{r,ref}}{c_r})^{\nu} \tag{3.36}$$

Here, the subscript $_{ref}$ denotes reference values, c_r is the magnitude of the relative velocity vector of the pre–collision velocity components and ν is a user–specified dimensionless parameter, not to be confused with collision frequency as in the BGK approximation! It is clear that, now, the effective collision cross section is a function of the size of the relative velocity vector. The greater the relative velocitiy of molecules, the smaller the effective cross section of a potential collision. This requires that the term d in equation (3.35) must be kept inside the integral.

Also, in the VSS model, the deflection angle is calculated as

$$\chi = 2\cos^{-1}((b/d)^{1/\alpha}) \tag{3.37}$$

Here, b is the distance of closest approach of the undisturbed pre–collision trajectories and α is a user–specifed parameter ranging between 1 and 2, not to be confused with the absorption coefficient used for the wall boundary condition! In the finite element discretisation, however, we consider the value of the collision term at a fixed node in physical space, implying that the parameter b is effectively zero. A Monte Carlo sampling procedure is therefore employed to randomly select the pre–collision positions of molecules with velocities $\boldsymbol{c}$ and $\boldsymbol{c}_1$ within a circle of radius d centred on the discontinuous p–space node under consideration as shown in Figure 3.9. This determines the direction in which the relative velocity vector will deflect. However, since this simulation is two-dimensional, the distance of closest approach of the undisturbed trajectories, b, is still zero. The parameter, b, is, therefore, randomly sampled between 0 and d to be used in equation (3.37) to determine the magnitude of the deflection angle.

Equation (3.34) is now written as

$$Q = n^2 \int_{-\infty}^{\infty} (f^* f_1^* - f f_1)\, \pi d^2\, c_r \mathrm{d}\, \boldsymbol{c}_1 \tag{3.38}$$

Figure 3.4.3 shows a typical pattern of the pre– and post–collision velocities for a collision molecule with a pre–collision velocity ($u = 270m/s, v = 200m/s$) colliding with an assembly of test molecules positioned at the quadrature sampling points in v–space. In this case, $\alpha = 1.5$. Figure 3.10(a) shows the pre–collision velocities of the test molecules. Figure 3.10(b) shows the post–collision velocities of the collision molecules based on collisions with each of the test molecules. Figure 3.10(c) shows

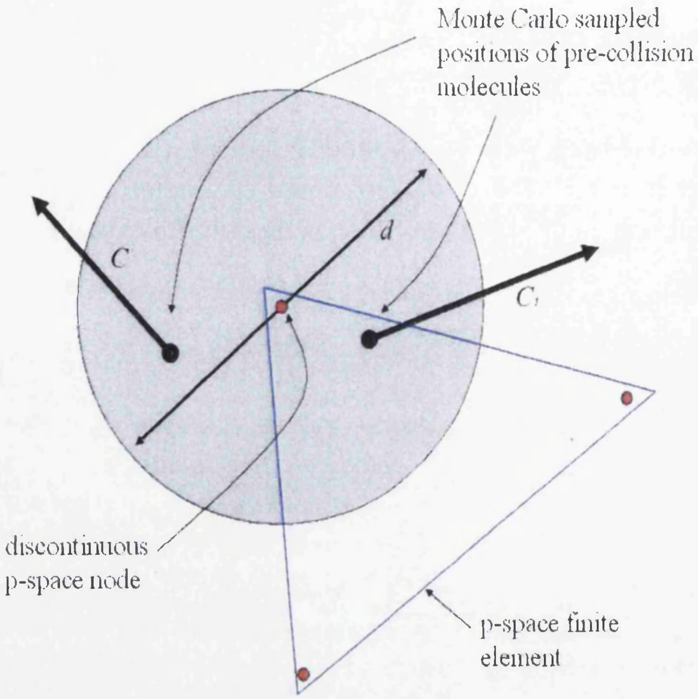

Monte Carlo sampled positions of pre-collision molecules
C
d
C₁
discontinuous p-space node
p-space finite element

Figure 3.9: Monte Carlo sampling to determine the collision deflection angle

the post–collision velocities of each of the test molecules. A ring of post–collision velocities centred on the coordinate ($u = 270m/s, v = 200m/s$) is observed in both post–collision figures. This is due to the concentration of sampling points at the origin in the Lobatto quadrature discretisation.

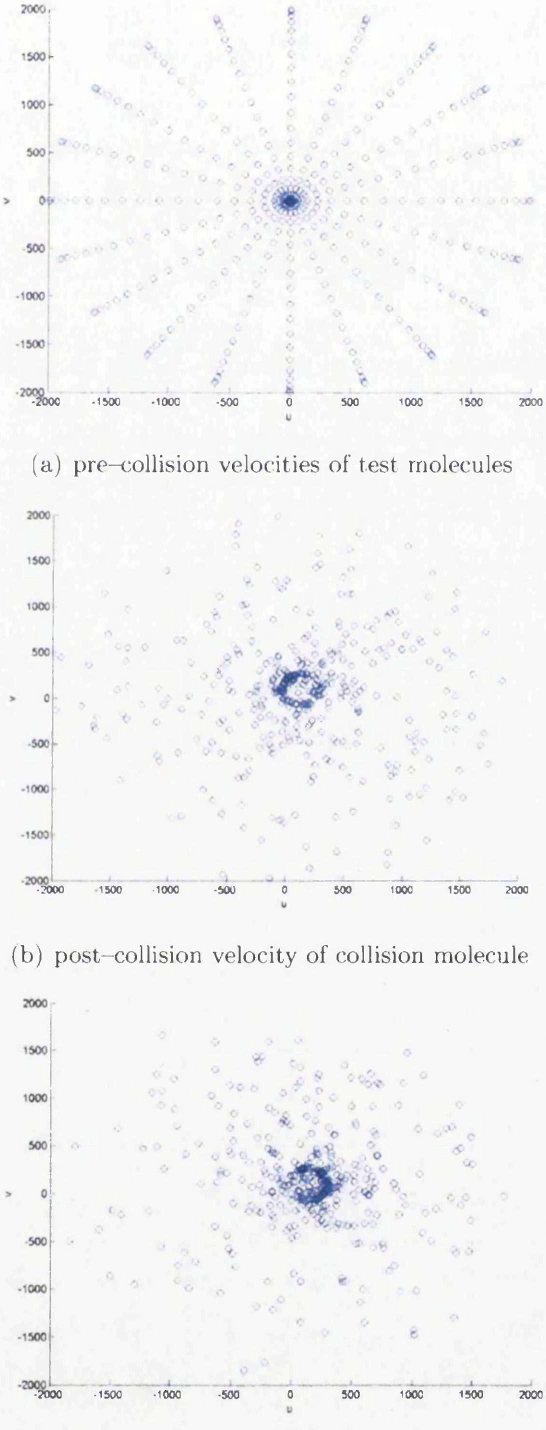
It is essential that momentum and energy are conserved in each collision. This is ensured by the binary collision theory detailed in section 2.2.2. If this is achieved, the integral of (nf) over v–space should not be altered by collisions and, hence, mass is conserved.

As in the hard–sphere molecule approach, the integral in equation (3.35) is evaluated using the spectral method. This was explained in more detail in section 3.3.

3.4.4 Numerical Evaluation of the Full Boltzmann Collision Integral

A convenient method for evaluating moment integrals of the distribution function was detailed in section 3.3. This procedure can be adapted to compute the full Boltzmann source term in equation (3.35) as

$$Q(\boldsymbol{c}_i) = \pi d^2 \sum_{j=1}^{N} w_j((nf)^*_i(nf)^*_j - (nf)_i(nf)_j)|c_r(ij)||J| \tag{3.39}$$



(a) pre–collision velocities of test molecules

(b) post–collision velocity of collision molecule

(c) post–collision velocities of test molecules

Figure 3.10: Example of collision dynamics using the VSS model and Monte Carlo sampling

and in equation (3.38) as

$$Q(\boldsymbol{c_i}) = \pi \sum_{j=1}^{N} w_j((nf)_i^*(nf)_j^* - (nf)_i(nf)_j)\, d^2\, |c_r(ij)||J| \tag{3.40}$$

where $\mathrm{Q}(\boldsymbol{c_i})$ is the source term for v–space node i, the summation is over all other v–space nodes j, w_j is the quadrature weight associated with node j, $(nf)_x$ is the pre–collision value of (nf) associated with node x, $(nf)^*_x$ is the post–collision value of (nf) associated with node x, $|c_r(ij)|$ is the magnitude of the relative velocity between nodes i and j, d is the computed collision cross section diameter, and $|J| = \pi r_v^2(\eta_j + 1)/4$.

3.5 Boundary Conditions

For the standard equations of continuum fluid mechanics, i.e. Navier–Stokes, Euler, we consider the conditions that must be placed upon the macroscopic variables at the boundaries of the domain of interest. For example, we might assume zero mass flux across a wall, or zero pressure gradient across an inflow. Also, the number of conditions placed on each boundary will be determined by the nature of the governing equations; elliptic, parabolic or hyperbolic. However, when solving the Boltzmann equation, our dependent variable is the distribution function and we must do a little bit more groundwork before we can establish the condition that must be placed upon this variable at the boundaries of the physical domain. A study of the literature reveals that there is no standard approach for dealing with these boundaries. A popular approach for solid wall boundaries, especially when applying a lattice Boltzmann method, is the bounce back technique ([113],[49],[50],[4]). Lattice Boltzmann techniques for open boundaries (inflows/outflows) are described in ([113],[4],[54],[116]). There are even more variations in boundary treatment approaches when studying the literature for fully discretised kinetic schemes. For example, Filbet and Russo [87] present a boundary approach applied to a high order time–splitting method for solution of the Boltzmann equation, Pareschi and Trazzi [99] present a boundary approach for their time relaxed Monte Carlo (TRMC) method and Tallec and Mallinger [61] present a method of coupling a Boltzmann domain to a Navier–Stokes domain using, what they term, half–fluxes. Each approach has its merits, but it is unclear which is the most appropriate in any given situation. The approach adopted here borrows ideas from several of the afore–mentioned references.

Application of the boundary conditions in the algorithm is achieved by an appropriate modelling of $F_{n,\boldsymbol{c}}^{m+\frac{1}{2}}$ in the first term on the right-hand side of equation (3.17) in the collisionless Boltzmann algorithm, and the second term on the right-hand side of equation (3.31) in the BGK algorithm. Since the choice of flux at the boundary is dependent on the molecular velocity at a given node in v–space, we are implicitly accounting for the hyperbolic or elliptic nature of the problem. This means that there is no need to adjust the boundary procedure for subsonic, sonic or supersonic inlets or outlets.

Essentially, three types of boundary need to be considered; inflow (inlet), outflow (outlet) and wall (solid).

3.5.1 Inflow

If the assumption is made that the gas flow entering the domain, as in Figure 3.11, is in thermodynamic equilibrium, and we know the macroscopic properties of the gas, i.e. density, bulk velocity, pressure, temperature, then we are in a position to be able to constuct the Maxwellian distribution function for the inflow.

For element edges at an inflow boundary, calculation of the inter–element flux will depend upon whether the molecular velocity is in a direction that is 'into' the

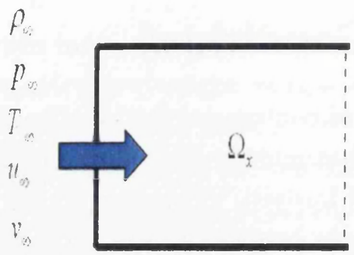


Figure 3.11: An Inflow Boundary

element ($\boldsymbol{c}.\boldsymbol{n} < 0$), as in the top element in Figure 3.12, or the molecular velocity is in a direction that is 'out of' the element ($\boldsymbol{c}.\boldsymbol{n} > 0$) as in the bottom element in Figure 3.12.

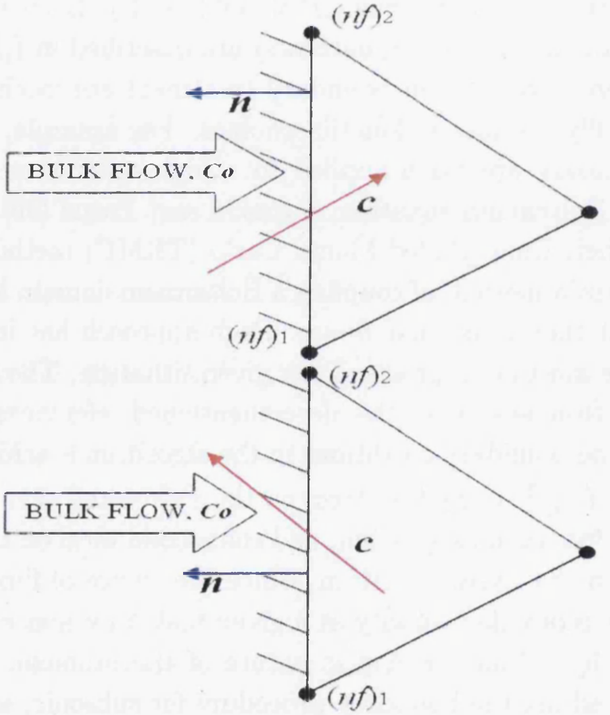


Figure 3.12: Element Edges at an Inflow Boundary

In the first case, the molecular velocity is directed into the element and the inter–element flux is constructed as

$$F^{m+\frac{1}{2}}_{n,\boldsymbol{c}} = \boldsymbol{c}.\boldsymbol{n}(\frac{\beta^2}{\pi})\exp(-\beta^2(\boldsymbol{c}-\boldsymbol{c_o})). \qquad (3.41)$$

$(\beta^2/\pi)\exp(-\beta^2(\boldsymbol{c}-\boldsymbol{c_o}))$ is the Maxwellian distribution function, equation (2.30), modified to its two dimensional equivalent, where $\beta = (2RT)^{-1/2} = (m/2kT)^{1/2}$.

If the molecular velocity is directed out of the element, the inter–element flux is constructed as

$$F_{n,\boldsymbol{c}}^{m+\frac{1}{2}} = \frac{1}{2}\boldsymbol{c}.\boldsymbol{n}((nf)_1^{m+\frac{1}{2}} + (nf)_2^{m+\frac{1}{2}}). \tag{3.42}$$

3.5.2 Outflow

If a domain boundary is an outflow, as in Figure 3.13, the boundary condition application is straightforward.

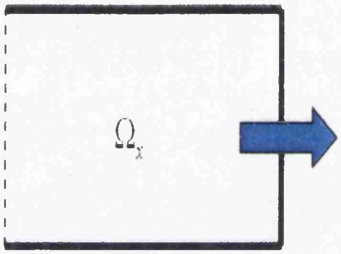


Figure 3.13: An Outflow Boundary

The method of construction of the inter–element flux is independent of the direction of the molecular velocity, i.e. it is independent of $\boldsymbol{c}.\boldsymbol{n}$. The inter–element flux is constructed as

$$F_{n,\boldsymbol{c}}^{m+\frac{1}{2}} = \frac{1}{2}\boldsymbol{c}.\boldsymbol{n}((nf)_1^{m+\frac{1}{2}} + (nf)_2^{m+\frac{1}{2}}) \tag{3.43}$$

and can, therefore, take on a positive or negative value.

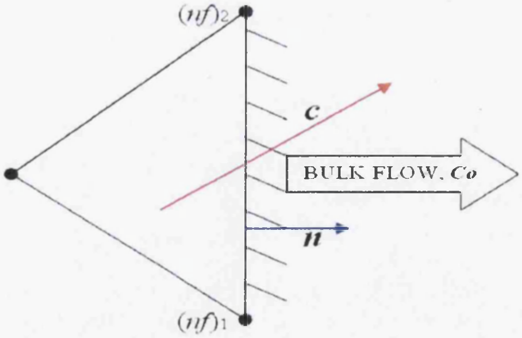


Figure 3.14: Element Edge at an Outflow Boundary

3.5.3 Wall

At a wall, the condition that must be enforced is zero mass flux across the boundary. In a kinetic theory description, this is expressed as

$$\int_{\Gamma_r}\int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\Gamma_r = 0 \tag{3.44}$$

where $F_{n,\boldsymbol{c}} = (\boldsymbol{c}.\boldsymbol{n})f(\boldsymbol{c},\boldsymbol{r},t)$ and Γ_r is the physical space domain boundary. This condition is ensured by an appropriate modelling of molecular collisions with the wall. We make the assumption that a certain fraction, α, of molecules are absorbed by the wall and remitted in equilibrium with the wall, i.e. they are reflected back into the domain with a Maxwellian distribution based on the wall temperature. This is termed diffuse reflection. The remaining fraction, $(1-\alpha)$, are not absorbed by the wall and simply reflect directly back into the domain. This is termed specular reflection. These two models, for the interaction of a flux of molecules with a solid surface, was first suggested by Maxwell [67]. The term α is known as the 'absorption coefficient' and the resulting distribution functions, in terms of the absorption coefficient, for molecules impinging on a wall and reflecting from a wall are shown in Figure 3.15.

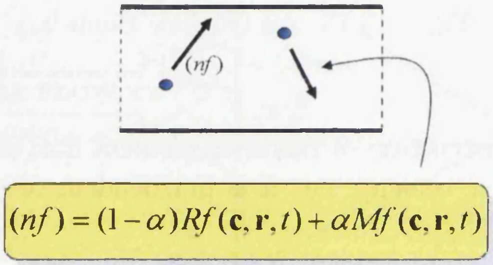


Figure 3.15: A Wall Boundary

The distribution function of the net reflected flux of molecules is, therefore, constructed as

$$f(\boldsymbol{c},\boldsymbol{r},t) = (1-\alpha)Rf(\boldsymbol{c},\boldsymbol{r},t) + \alpha Mf(\boldsymbol{c},\boldsymbol{r},t), \qquad \text{for} \quad \boldsymbol{c}.\boldsymbol{n} \leq 0 \tag{3.45}$$

where α is the absorption coefficient,

$$Rf(\boldsymbol{c},\boldsymbol{r},t) = f(\boldsymbol{c} - 2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}),\boldsymbol{r},t) \tag{3.46}$$

$$Mf(\boldsymbol{c},\boldsymbol{r},t) = \eta(\boldsymbol{r},t)M_w(\boldsymbol{c}) \tag{3.47}$$

and $\boldsymbol{n}$ is the outward facing unit normal at the wall, as shown in Figure 3.16. If T_w is the wall temperature and R is the gas constant, then M_w is determined as

$$M_w = \exp(-\frac{c^2}{2\,R\,T_w}) \tag{3.48}$$

The parameter η is used to enforce the condition in equation (3.44) i.e. it is used to ensure conservation of mass at the the wall, which implies that

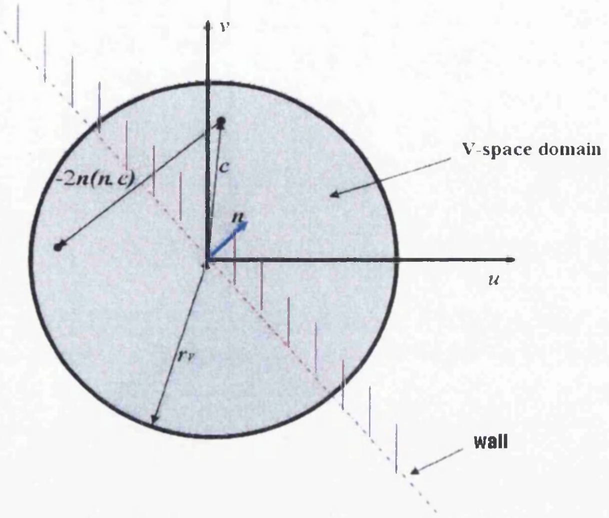


Figure 3.16: Coordinate transformation in V-space under Specular Reflection

$$\eta(\boldsymbol{r},t)=\frac{\displaystyle\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})>0} f(\boldsymbol{c},\boldsymbol{r},t)|\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{c}}{\displaystyle\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})\leq 0} M_w(\boldsymbol{c})|\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{c}} \tag{3.49}$$

A technique similar to this has been applied by Pareschi and Trazzi [99] to a time relaxed Monte Carlo procedure for solution of the Boltzmann equation.

Mathematical Proof of the Wall Boundary Condition

We saw in equation (3.44) that the condition of zero mass flux across a wall is written as

$$\int_{\Gamma_r}\int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}=0. \tag{3.50}$$

This statement is essentially that the integral of the distribution function over the region of v–space outside of the domain bounded by the wall is identical to the integral of the distribution function over the region of v–space inside the domain. These regions are indicated in Figure 3.17. If the reflected wall flux is constructed as described in the previous section, we can write the left hand side of equation (3.50) as

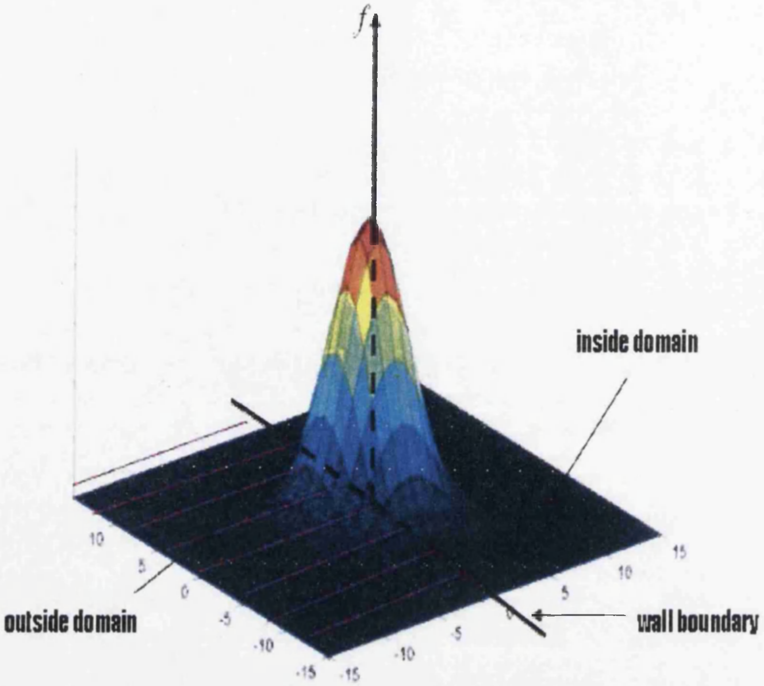


Figure 3.17: Distribution Function at a Wall Boundary

$$
\begin{aligned}
&\int_{\Gamma_r}\int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}\\
&\qquad = \int_{\Gamma_r}\int_{-\infty}^{+\infty} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}\\
&\text{since}\quad F_{n,\boldsymbol{c}} = \boldsymbol{c}.\boldsymbol{n}f_{\boldsymbol{c}}\\
&\qquad = \int_{\Gamma_r}\left(\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \boldsymbol{c}.\boldsymbol{n}(1-\alpha)f(\boldsymbol{c}-2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}),\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \cdots\right.\\
&\qquad\left.\int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \alpha\boldsymbol{c}.\boldsymbol{n}\eta(\boldsymbol{r},t)M_w(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\right)\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}
\end{aligned}
$$

by splitting the integral over v-space into $\boldsymbol{c}.\boldsymbol{n} > 0$ and $\boldsymbol{c}.\boldsymbol{n} \leq 0$ parts, and substituting for f from equation (3.45).

It is shown in Appendix A, that this expression reduces to zero and, hence, the condition described by equation 3.44 is achieved and we have zero mass flux across the wall.

Now that we have a method for modelling the distribution function at a wall whilst ensuring zero mass flux across the wall, we must turn our attention to the construction of the inter–element flux term. For molecular velocities towards the wall ($\boldsymbol{c}.\boldsymbol{n} > 0$), as in the left–hand element in Figure 3.18, the inter–element flux is

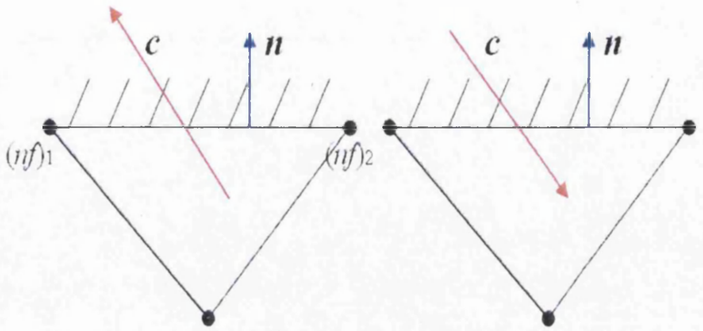


Figure 3.18: Element Edges at a Wall Boundary

constructed as

$$F_{n,\boldsymbol{c}}^{m+\frac{1}{2}} = \frac{1}{2}\boldsymbol{c}.\boldsymbol{n}((nf)_1^{m+\frac{1}{2}} + (nf)_2^{m+\frac{1}{2}}). \tag{3.51}$$

At a wall boundary, the construction of the inter–element flux will depend on the direction of the molecular velocity. For molecular velocities away from the wall ($\boldsymbol{c}.\boldsymbol{n} < 0$), as in the right–hand element in Figure 3.18, the inter–element flux is constructed based on the reflected distribution function defined by equation (3.45), so that

$$F_{n,\boldsymbol{c}}^{m+\frac{1}{2}} = \boldsymbol{c}.\boldsymbol{n}((1-\alpha)Rf^{m+\frac{1}{2}}(\boldsymbol{c},\boldsymbol{r},t) + \alpha M f^{m+\frac{1}{2}}(\boldsymbol{c},\boldsymbol{r},t)). \tag{3.52}$$

Here, α is the absorption coefficient and

$$Rf^{m+\frac{1}{2}}(\boldsymbol{c},\boldsymbol{r},t) = f^{m+\frac{1}{2}}(\boldsymbol{c} - 2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}),\boldsymbol{r},t) \tag{3.53}$$

$$Mf^{m+\frac{1}{2}}(\boldsymbol{c},\boldsymbol{r},t) = \eta^{m+\frac{1}{2}}(\boldsymbol{r},t)M_w(\boldsymbol{c}). \tag{3.54}$$

If T_w is the wall temperature and R is the gas constant, then M_w is determined as in equation 3.48 and

$$\eta^{m+\frac{1}{2}}(\boldsymbol{r},t) = \frac{\displaystyle\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})>0} f^{m+\frac{1}{2}}(\boldsymbol{c},\boldsymbol{r},t)|\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{c}}{\displaystyle\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})\leq 0} M_w(\boldsymbol{c})|\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{c}} \tag{3.55}$$

The calulation of η will depend on whether the v–space domain is discretised using linear quads or the spectral method.

Calculation of η using v-space quads

The numerator in the term on the right–hand side of equation (3.55) is determined as

$$\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})>0} f(\boldsymbol{c},\boldsymbol{r},t)|\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{x})|\,\mathrm{d}\boldsymbol{c} = \frac{\displaystyle\sum_{e=1}^{N} |\boldsymbol{c}.\boldsymbol{n}|\,[nf]_e A_e}{\displaystyle\sum_{e=1}^{N} [nf]_e A_e} \tag{3.56}$$

where the summations extend over all elements with at least two nodes with velocity coordinate (u, v) such that $\boldsymbol{c}.\boldsymbol{n} > 0$ i.e. the shaded elements in Figure 3.19, $[nf]_e$ is the mean value of (nf) for the four element nodes and A_e is the area of the velocity space element.

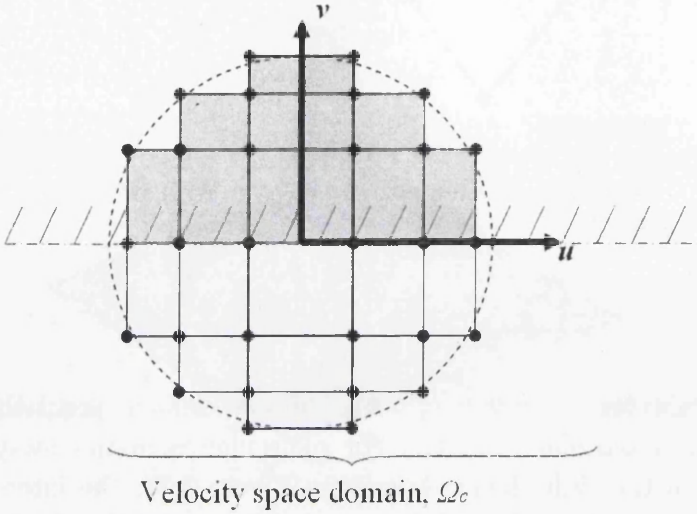


Figure 3.19: Elements to be included in calculating η

The denominator in the term on the right-hand side of equation (3.55) can be calculated analytically as

$$\begin{aligned}
\int_{\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})\leq 0} M_w(\boldsymbol{c})|\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})|\,\mathbf{dc} &= \int_{-\infty}^{+\infty}\int_{0}^{+\infty} |u|\exp(\frac{-u^2}{2RT_w})\exp(\frac{-v^2}{2RT_w})\,\mathrm{d}u\,\mathrm{d}v \\
&\quad \text{since } |\boldsymbol{c}.\boldsymbol{n}(\boldsymbol{r})| = |u| \text{ with reference to Figure 3.19} \\
&= \int_{-\infty}^{+\infty} \exp(\frac{-v^2}{2RT_w})\frac{2RT_w}{2}\,\mathrm{d}v \\
&\quad \text{since } \int_{0}^{+\infty} u\exp(\frac{-u^2}{a})\,\mathrm{d}u = \frac{a}{2} \\
&= RT_w\int_{-\infty}^{+\infty} \exp(\frac{-v^2}{2RT_w})\,\mathrm{d}v \\
&= RT_w\pi^{\frac{1}{2}}(2RT_w)^{\frac{1}{2}} \\
&\quad \text{since } \int_{-\infty}^{+\infty} \exp(\frac{-v^2}{a})\,\mathrm{d}v = \pi^{\frac{1}{2}}v^{\frac{1}{2}} \\
&= \sqrt{2\pi R^3T_w^3} \qquad (3.57)
\end{aligned}$$

The above derivation has been conducted based upon the wall orientation as in Figure 3.19, but due to the spherical symmetry of the Maxwellian distribution function, the final result is applicable to any general case of wall configuration.

Calculation of η using spectral v–space

As the approach for spectral v–space discretisation outlined in section 3.1.2 is rotationally symmetric about the origin in terms of the locations of the sampling points, it is straightforward to compute integrals over half of the v–space. The procedure for computing integrals over the full v–space was detailed in section 3.3.2and the simple modification that is required to integrate over half of the v–space is to only include sampling points for which $\boldsymbol{c}.\boldsymbol{n} > 0$ in the summation process, i.e. sampling points that fall within the shaded region in Figure 3.20.

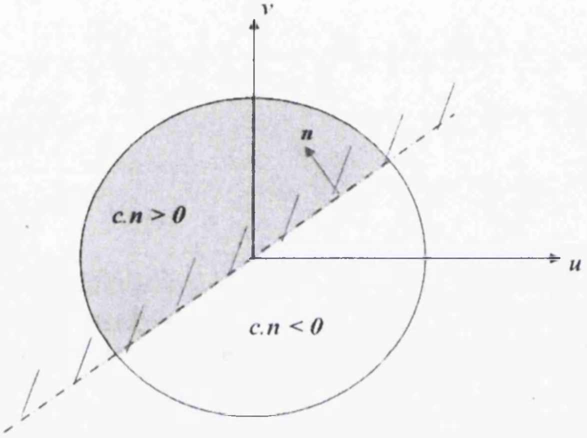


Figure 3.20: Sampling points to be summed in calculating η

This allows the determination of the integral in the numerator on the right–hand side of equation (3.55). The integral in the demoninator can again be determined using the analytical approach detailed in the preceding section.

Wall Boundary Relaxation

For high subsonic and supersonic applications, a wall boundary relaxation procedure is required and applied over the first few hundred timesteps. This procedure effectively 'switches on' the wall boundary gradually rather than instantaneously at the first timestep allowing convergence towards a steady state from a uniform free–flow initial condition.

The implementation of this is straightforward and simply requires that the incoming flux at a wall boundary is modified over the initial r timesteps as

$$F_{n,\boldsymbol{c}}^{m+\frac{1}{2}} = (1 - \frac{m}{r})F_{out} + (\frac{m}{r})F_{wall} \tag{3.58}$$

where F_{out} is the equivalent outlet flux (as detailed in section 3.5.2) and F_{wall} is the wall flux (as described in section 3.5.3).

3.6 Obtaining the Solution at the Mesh Nodes

The macroscopic variables are calculated at the element nodes using the method described in section 3.3. However, for post–processing of the results, data defined at the mesh nodes is used. This is achieved using the simple element weighted averaging procedure.

$$\overline{Q_m} = \frac{\sum_{i=1}^{n} A_i \overline{Q_i}}{\sum_{i=1}^{n} A_i} \tag{3.59}$$

where $\overline{Q_m}$ is the value of the macroscopic variable at the mesh node, the summations are over each element node, i, meeting at the mesh node, A_i is the area of the element associated with element node i and $\overline{Q_i}$ is the value of the macroscopic variable at discontinuous node i.

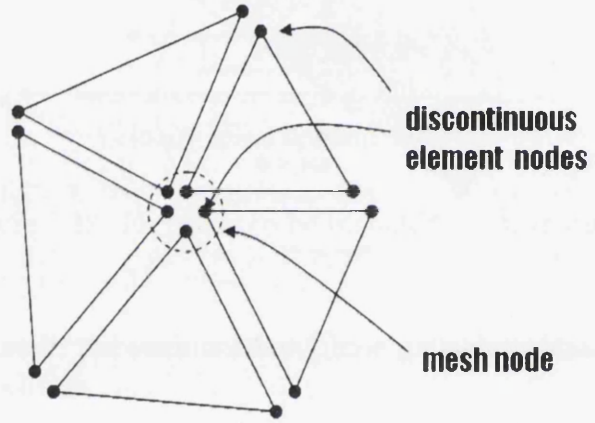


Figure 3.21: A mesh node and element nodes

3.7 Parallelisation

Due to the large memory requirement involved in discretising both p–space and v–space domains, parallelisation of the algorithm was an essential requirement for the code to be applied to realistic problems. Physical space domain decomposition was chosen to achieve this, as shown in Figure 3.22.

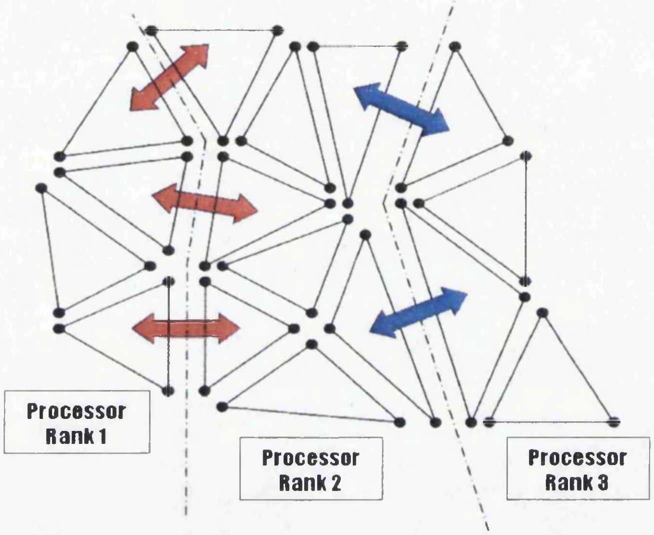
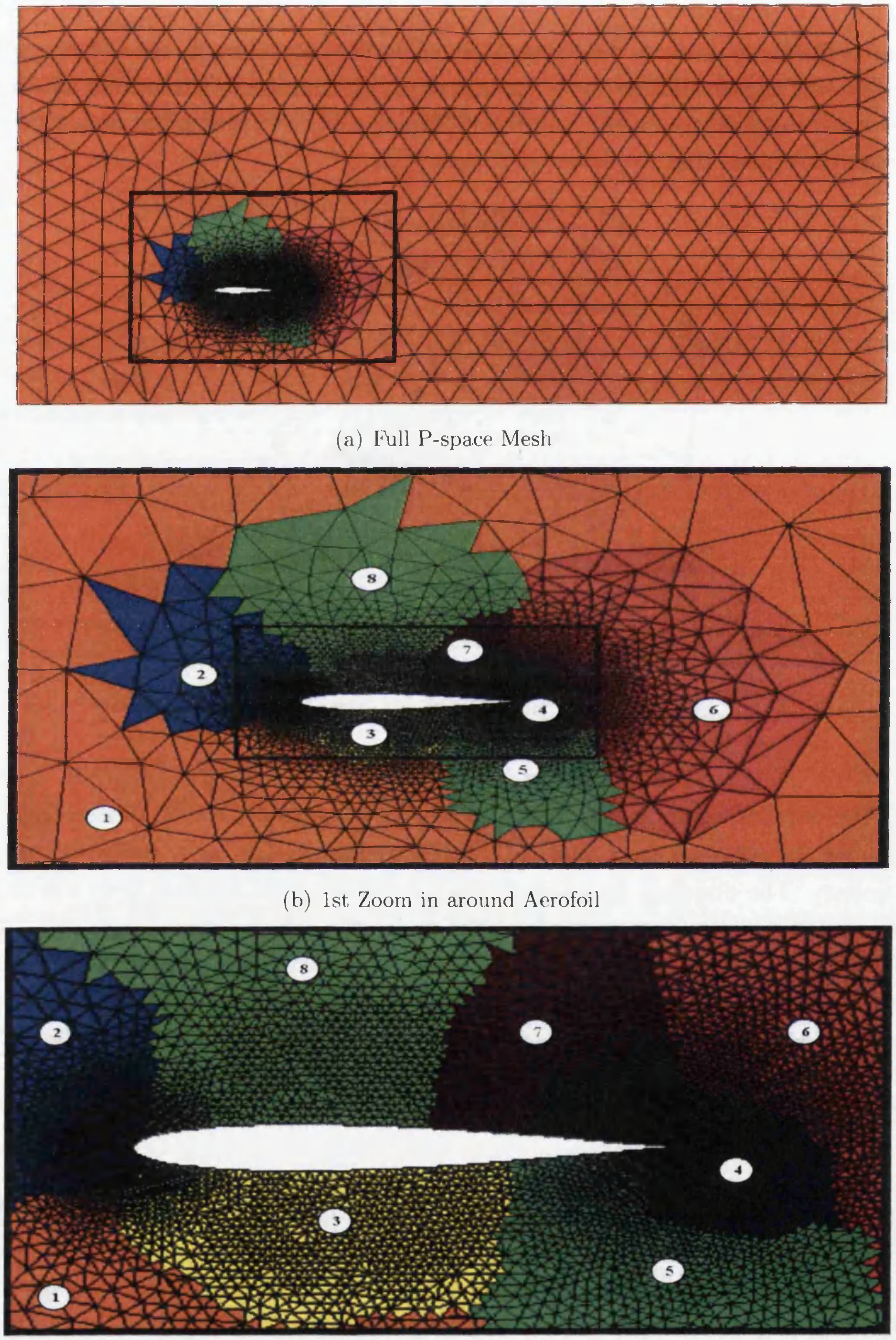


Figure 3.22: Physical Domain Decomposition

The only information that needs to be communicated between processors is the value of the inter–element flux, $F_n^{m+\frac{1}{2}}$, for element edges at a processor domain boundary. The direction in which this information passes depends upon the molecular velocity vector. For example, if the molecular velocity vector is directed from left to right with reference to Figure 3.22, then processor 1 will pass the inter–element flux for its three processor domain element edges to processor 2 and processor 2 will pass the inter–element flux for its two processor domain element edges in contact with processor 3's domain to processor 3. Processor 3 will not communicate any information to processor 2 and processor 2 will not communicate any information to processor 1.

The partitioning is performed as a pre–processing step using the software package METIS to form a partition file, indicating which elements should be distributed to which processors. This program partitions the mesh in such a way as to minimise the number of communicating edges and, hence, maximise the efficiency of the code. An example of how a typical mesh is partitioned is shown in Figure 3.23. This mesh, around an aerofoil cross section, consists of 10,796 elements partitioned between 8 processors. It is clear that the partitioning process has been performed in such a way as to minimise the communication between processors, i.e. it minimises the number of processor boundaries, whilst evenly distributing the number of elements per processor.



(a) Full P-space Mesh

(b) 1st Zoom in around Aerofoil

(c) 2nd Zoom in around Aerofoil

Figure 3.23: Partitioning of the Physical Space Mesh

A master–slave approach was used for the parallelisation. Only the master processor, processor rank 0, has any form of communication with the outside world and its job is to read the input data to start the program, distribute the relevant data to each slave processor, to allow it to begin 'solving', and to assemble the output data file at the end of each timestep, or every given number of timesteps. Message Passing Interface (MPI) is used as the independent software for communications between processors. The relevant flux information was passed between processors simply by calling the predefined MPI routines from the FORTRAN code.

3.8 Coding and Hardware

Initial development took place in serial using the FORTRAN77 language and compilation using Salford Plato3. A standard desktop PC was used for running these jobs, with run times of no longer than a few hours. When the code was parallelised, the language was updated to take advantage of some of the benefits of the FORTRAN90 language, such as allocatable memory. A flow chart showing the organisation of the various routines that comprise the FORTRAN90 Boltzmann solver code is given in Appendix C.

Two machines were used for running the parallel version of the code. The first was the Swansea University Civil and Computational Engineering Centre (C2EC) PC cluster. This is essentially a group of 16 processors connected via Myrinet, a special purpose optical fibre network with high bandwidth (2Gbit/s) and low latency. Each processor has memory associated with it ranging from 2GB to 8GB depending on the processor. This machine was used for the smaller, i.e. lower memory, shorter timescale, runs due to the limit on the number of available processors and the 24 hour limit on the runtime.

The second machine used for running the parallel code was Swansea University's high performance computing (HPC) facility, BLUE C. This is a large-scale supercomputer consisting of 11 nodes of 32 processors. Each node has 120 GB of memory associated with it. Larger runs were undertaken using this facility.

Chapter 4

Examples

4.1 Development of a Scalar Convection Equation Solver

In section 3.2.1, it was mentioned that underpinning the Boltzmann solver is a simple scalar convection algorithm. Various scalar convection algorithms were written to test the properties of the different methods. Initially, simple and standard finite difference techniques were developed in MATLAB and the code for these short programs is contained in Appendix C. All of the MATLAB programs detailed in this section were run from a standard PC with a 3.0 GHz Intel Pentium processor and 1024 MB RAM.

This was followed by a two–step discontinuous Taylor–Galerkin algorithm implemented in FORTRAN77. We will briefly look at some examples from these programs, as they will give some helpful understanding of what is happening within the more complex Boltzmann solver, which will be examined later in this chapter. In developing a scalar convection algorithm, we are aiming to achieve the transportive property i.e. it must be able to propagate a signal in a given direction, at a given speed, without distorting the form of that signal. If this property is achieved in the underlying scalar convection code, then the Boltzmann solver can be developed from a solid footing.

4.1.1 An FTCS Finite Difference Method

The forward time centre space (FTCS) algorithm is a simple finite difference scheme that can be constructed and used as an attempt to solve the scalar convection equation, equation (3.5). A 1D explicit FTCS scheme, with 2nd order accuracy in space, can be written as

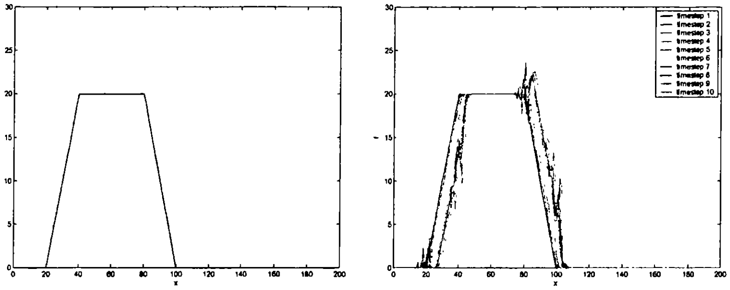
$$f_i^{n+1} = f_i^n - \Delta t\, u(\frac{f_{i+1}^n - f_{i-1}^n}{2h}) \tag{4.1}$$

where Δt is the timestep size, u is the convection velocity, h is the constant distance between sampling points and f_i^n indicates the value of the function at x_i and timestep n. This scheme must be complemented by suitable boundary and initial conditions. Here we look at the convection of a simple pulse and a sine wave from left to right with non–dimensional speed, $u = 5$, and a non–dimensional timestep, $\Delta t = 0.1$.

Unfortunately, as can be shown by a suitable von Neumann analysis [41] and as is indicated by the results in Figures 4.1(b) and 4.2(b), the FTCS scheme is unconditionally unstable, for a purely advective system such as the scalar convection equation.

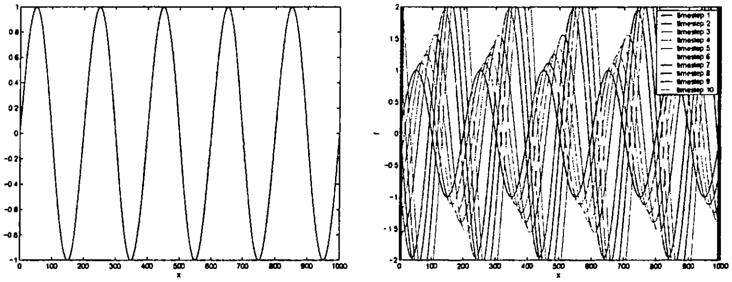
4.1.2 An Upwind Finite Difference Method

An equally simple scheme, which takes into account the physics of the flow, is an upwind method. Here, gradients are determined using only data points in the



(a) Initial conditions (b) Timesteps 1 - 10

Figure 4.1: FTCS Convection of a Simple Pulse



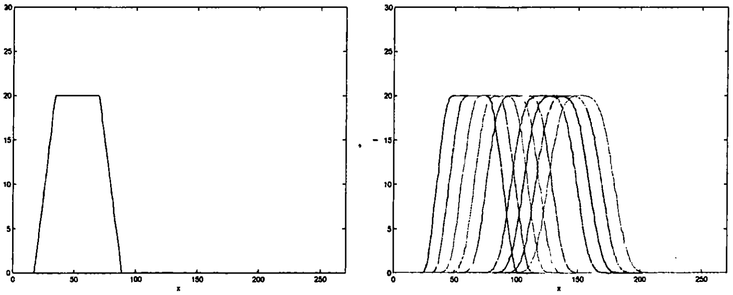
(a) Initial conditions (b) Timesteps 1 - 10

Figure 4.2: FTCS Convection of a Sine Wave

upstream direction to account for the direction in which information is propagating. For flow from left to right, the scheme can be written as

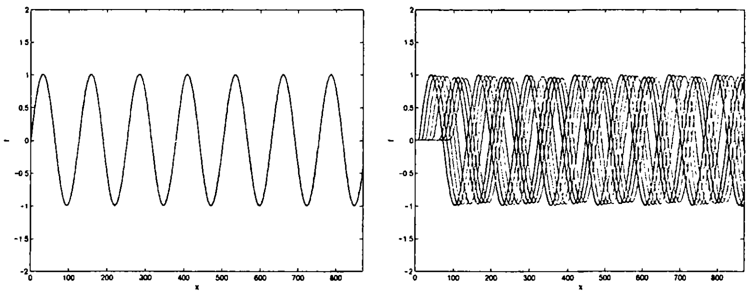
$$f_i^{n+1} = f_i^n - \frac{u\,\Delta t}{h}(f_i^n - f_{i-1}^n). \tag{4.2}$$

where identical notation has been used.



(a) Initial conditions (b) Timesteps 1 - 200 (every 20 timesteps shown)

Figure 4.3: Upwind Convection of a Simple Pulse



(a) Initial conditions

(b) Timesteps 1 - 200 (every 20 timesteps shown)

Figure 4.4: Upwind Convection of a Sine Wave

Again, we consider the pure convection of a simple pulse and sine wave. In this case the distance between sampling points was not fixed. Instead the distance between consecutive sampling points $i-1$ and i was allowed to vary randomly between 0.5 and 1.25. The stability of this scheme is governed by the CFL condition which states that for stability we must ensure

$$C \leq 1.0 \tag{4.3}$$

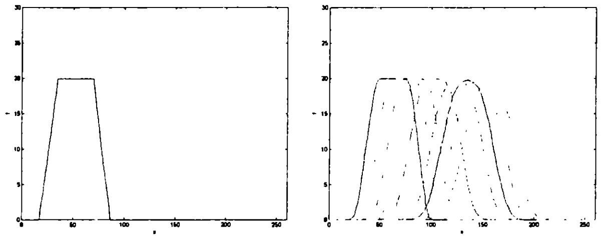
where the Courant number, $C = u\Delta t/h_{min}$. The convection velocity was set as $u = 5$ and the Courant number, $C = 1.0$. Figures 4.1.2 and 4.1.2 show that this scheme is stable. However, in the case of the pulse convection, the signal is seen to be smeared and in the case of the sine wave convection, the signal is decaying with time. This decay is not evident in Figure 4.3(b). However, running for a larger number of timesteps does result in a decay of the signal for the pulse propagation. Another disadvantage of this method is that the scheme is dependent on the direction of the convection velocity. This is not a problem in the 1D case examined here, but can become a more significant drawback in higher dimensions. Also, the upwind method is only first order accurate in space, compared with the second order accuracy achieved in the FTCS scheme. Higher order upwind schemes are available ([115],[105],[89]) but alternative schemes including the Lax and Lax–Wendroff techniques are the most established methods for overcoming some of the problems seen so far. These schemes will be examined next.

4.1.3 A Lax Finite Difference Method

The Lax finite difference scheme for the scalar convection equation is a simple extension to the FTCS scheme with the f_i^n term in equation (4.1) replaced by $1/2(f_{i+1}^n + f_{i-1}^n)$. The Lax scheme is written as

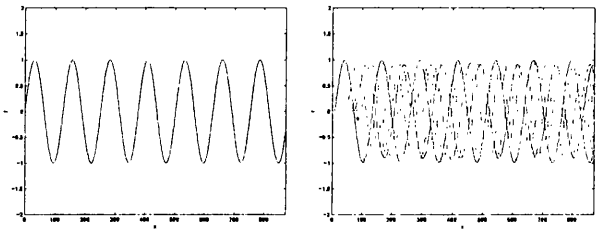
$$f_i^{n+1} = \frac{1}{2}(f_{i+1}^n + f_{i-1}^n) - \Delta t\, u(\frac{f_{i+1}^n - f_{i-1}^n}{h_{i-1} + h_i}) \tag{4.4}$$

with h_i now indicating the distance between sampling points i and $i+1$ but with all other notation identical to that used previously.



(a) Initial conditions (b) Timesteps 1 - 200 (every 20 timesteps shown)

Figure 4.5: Lax Convection of a Simple Pulse



(a) Initial conditions (b) Timesteps 1 - 200 (every 20 timesteps shown)

Figure 4.6: Lax Convection of a Sine Wave

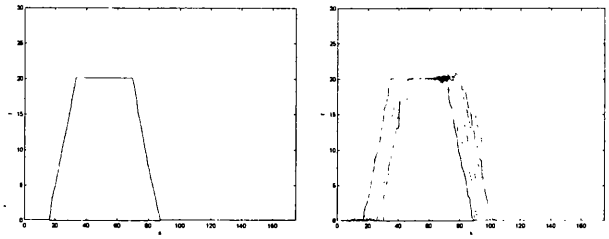
Figures 4.5(b) and 4.6(b) indicate that the LAX scheme again results in a smearing of the convected pulse and a decay of both the pulse and sine wave, but that the scheme is stable. These cases were run under the same conditions as the upwind scheme examples ($u = 5$, $C = 1.0$, $0.5 \leq h \leq 1.25$).

4.1.4 A Lax–Wendroff Finite Difference Method in 1D

The Lax-Wendroff approach achieves stability and second order accuracy in space and time. The scheme can be written, in two steps, as

$$\begin{aligned} f_{i+\frac{1}{2}}^{n+\frac{1}{2}} &= \frac{1}{2}(f_{i+1}^n + f_i^n) - \frac{\Delta t\, u}{2h_i}(f_{i+1}^n - f_i^n) \\ f_i^{n+1} &= f_i^n - \frac{\Delta t\, u}{0.5(h_i + h_{i-1})}(f_{i+\frac{1}{2}}^{n+\frac{1}{2}} - f_{i-\frac{1}{2}}^{n+\frac{1}{2}}) \end{aligned} \tag{4.5}$$

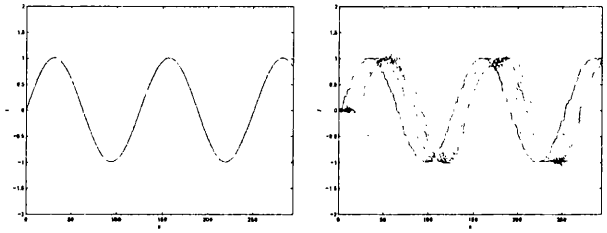
Both of the Lax–Wendroff examples were run with a Courant number of 1.0 (required for stability) and sampling points spaced randomly, as in the previous examples with $0.5 \leq h \leq 1.25$. It is clear that the Lax–Wendroff scheme best



(a) Initial conditions

(b) Timesteps 1 - 100 (every 10 timesteps shown)

Figure 4.7: Lax–Wendroff convection of a simple pulse



(a) Initial conditions

(b) Timesteps 1 - 200 (every 20 timesteps shown)

Figure 4.8: Lax–Wendroff convection of a sine wave

preserves the basic shape of the signal, but the signal is significantly distorted, particularly when the function undergoes large changes in gradient. These distortions are due to the unstructured nature of the discretisation employed.

4.1.5 A Lax–Wendroff Finite Difference Method in 2D

The Lax–Wendroff scheme was extended into a second spatial dimension. The simple function used as an initial condition for a test case is shown in Figure 4.9. In the 2D example shown here, the distance between sampling points, h, is non–uniform in both the x and y directions, but limited by the condition $0.75 \leq h \leq 1.25$, the convection velocity is fixed as $\boldsymbol{u} = 15\boldsymbol{i} + 15\boldsymbol{j}$ (where $\boldsymbol{i}$ and $\boldsymbol{j}$ are unit vectors in the x and y directions respectively) and the x–y plane is such that $0 \leq x \leq 100$ and $0 \leq y \leq 100$. The timestep is computed as

$$\Delta t = \frac{C h_{min}}{|\boldsymbol{u}|} \tag{4.6}$$

where h_{min} is the minimum sampling point spacing in either the x or y directions, $|\boldsymbol{u}|$ is the magnitude of the convection velocity and the Courant number, $C = 1.0$. With the convection velocity fixed as described, the form of the initial function should not change, but be seen to convect diagonally across the x–y plane.

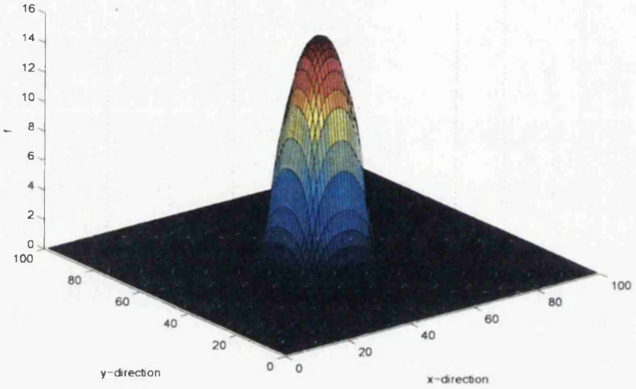


Figure 4.9: Initial Conditions for Lax–Wendroff Convection of a 2D Pulse

The results, in Figure 4.10, show the convection of the initial pulse from three different perspectives. The first column of graphs shows the convection from the perspective of the x–axis, the second column is a view from the perspective of the y–axis and the third column is a view of the whole x–y plane. It is evident, particularly from the first two columns, that the function is decaying with time. This decay characteristic is a property that was not observed in the 1D Lax–Wendroff case.

The finite element equivalent of the Lax–Wendroff method desribed here is the two–step Taylor–Galerkin method. This method is examined next.

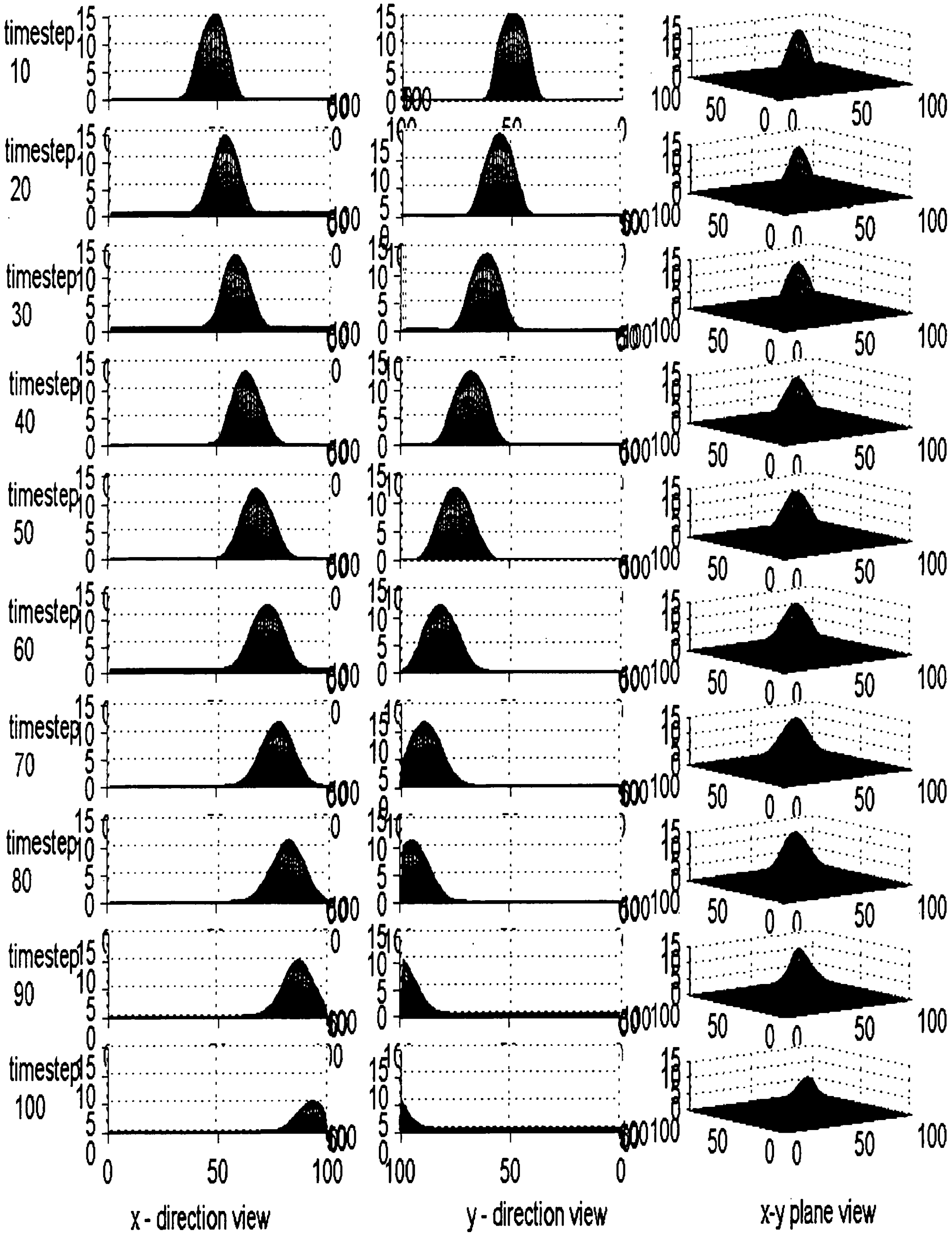


Figure 4.10: Lax–Wendroff Convection in Two Dimensions

4.1.6 Comparison of Continuous and Discontinuous Taylor–Galerkin Finite Element Methods

The problem under consideration here is that of the propagation of a discontinuity from left to right along a two dimensional, rectangular domain using linear, triangular elements. This problem was analysed, as it should demonstrate the benefits of using discontinuous finite elements over continuous finite elements. Initially, the scalar variable f was set to 1.0 across the whole domain and an inflow with $f = 5.0, u = 10.0, v = 0.0$ was set up on the left–hand boundary. The Courant number was set to 0.9 and initial tests were conducted using the structured mesh shown in Figure 4.11.

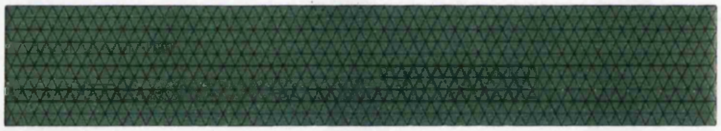


Figure 4.11: Structured Mesh for Discontinuity Propagation Problem

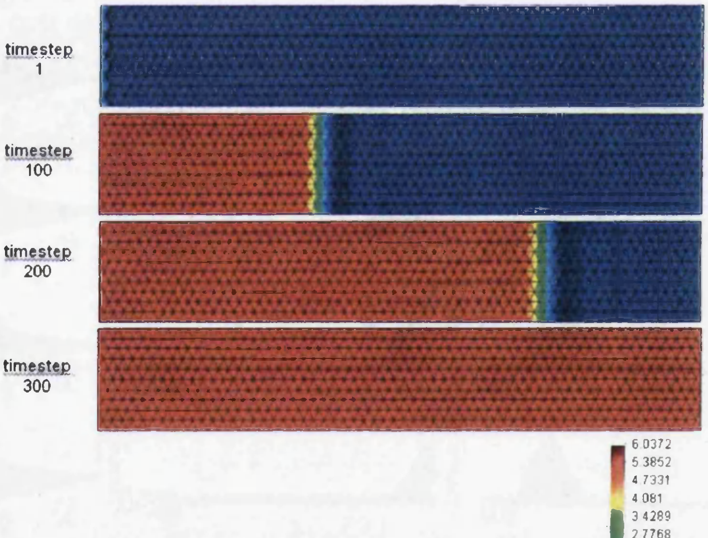


Figure 4.12: Propagation of a Discontinuity using a Continuous Taylor–Galerkin Method

Figure 4.12 shows a plan view of the discontinuity propagating from left to right and Figure 4.13 shows the time evolution of the solution at points spaced along the horizontal centreline of the domain. It is clear that there is a significant distortion to the solution at both the top and bottom of the discontinuity. In the field of Fourier

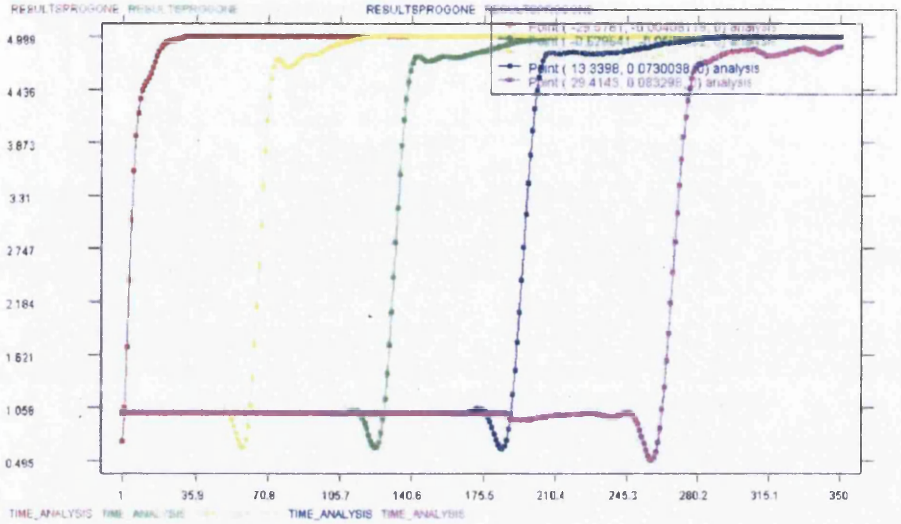


Figure 4.13: Point Analysis of the Discontinuity Convection using a Continuous Taylor–Galerkin Method

analysis, this is referred to as the Gibbs phenomenon. However, the propagation speed of the discontinuity is correct.

Figures 4.14 and 4.15 show equivalent results using discontinuous elements. It was found that a Courant number of 0.5 was required for stability in this case. A lower stability Courant number leads to a shorter timestep and, hence, larger runtime, but the results are clearly an improvement over the continuous finite element method. The distortion at the bottom of the discontinuity has been largely removed but a small overshoot at the top of the shock exists. This could be removed using a filter such as that described by Hirsch in [41] or a discontinuous Galerkin limiter as such as described by Burbeau et al in [12] if the Gibbs phenomenon overshoots and undershoots and the discontinuity are deemed unaccepable. This was not the case in this work, since the final solutions for the Boltzmann equation result from averages over a large number of scalar convection solutions. This means that the oscillations in the individual solutions do not significantly affect the macroscopic properties obtained by taking moments of the distribution function (see section 2.4).

Finally, the same problem, under the same conditions, was analysed using the discontinuous method, but on the unstructured mesh, shown in Figure 4.16, as the problems to be analysed later, using a full Boltzmann solver, would require an unstructured meshing of the domain. Figures 4.17 and 4.18 indicate that switching from a structured mesh to an unstructured mesh led to a small oscillation being set up at the top of the shock. It is unclear why this should be the case, but again, these oscillations could potentially be removed by means of a suitable filter if necessary. An understanding of the properties of various scalar convection algorithms, gained from studying the examples shown in the preceding sections, was important for

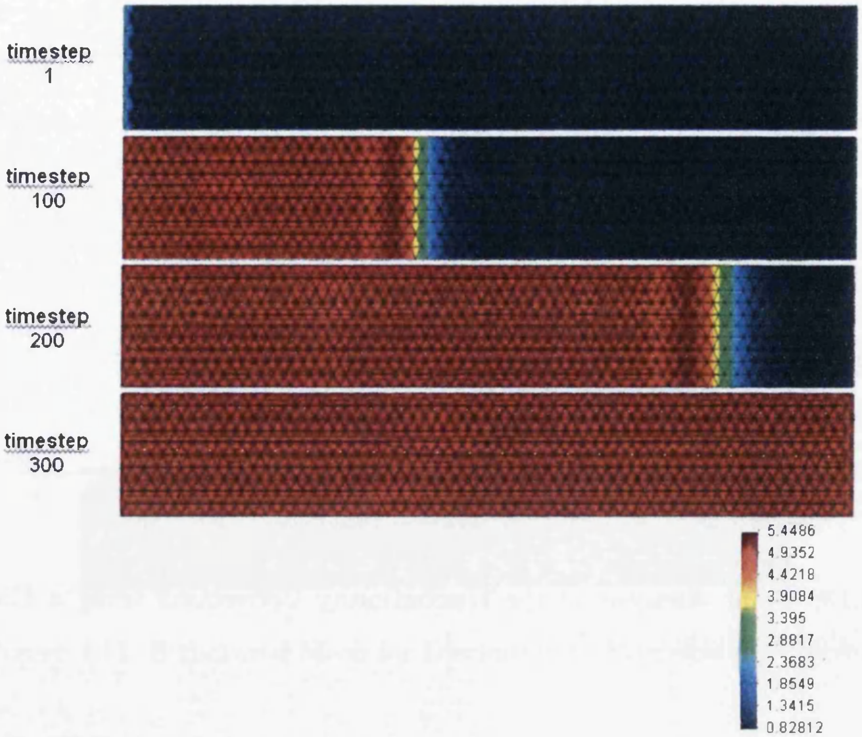


Figure 4.14: Convection of a Discontinuity using a Discontinuous Taylor–Galerkin Method (structured mesh)

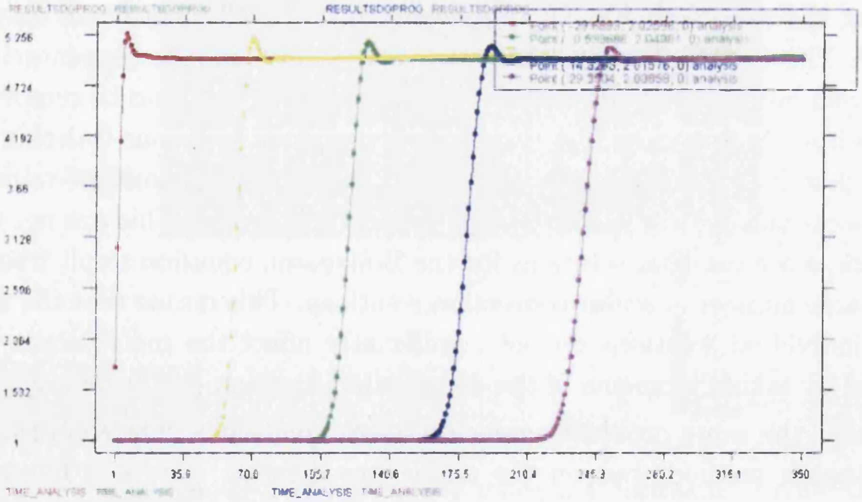


Figure 4.15: Point Analysis of the Discontinuity Convection using a Discontinuous Taylor–Galerkin Method (structured mesh)

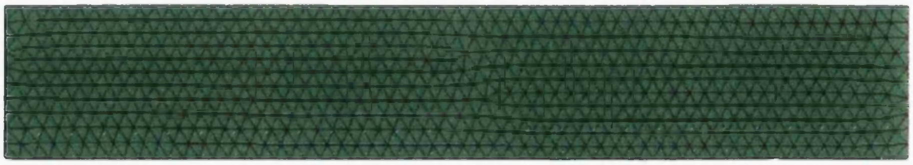


Figure 4.16: Unstructured Mesh for Discontinuity Convection

analysis of the Boltzmann solver results which will be shown in the following sections. The code development for the Boltzmann solver detailed in Chapter 3 began with the discontinuous Taylor–Galerkin scalar convection code used for the discontinuity propagation example already shown.

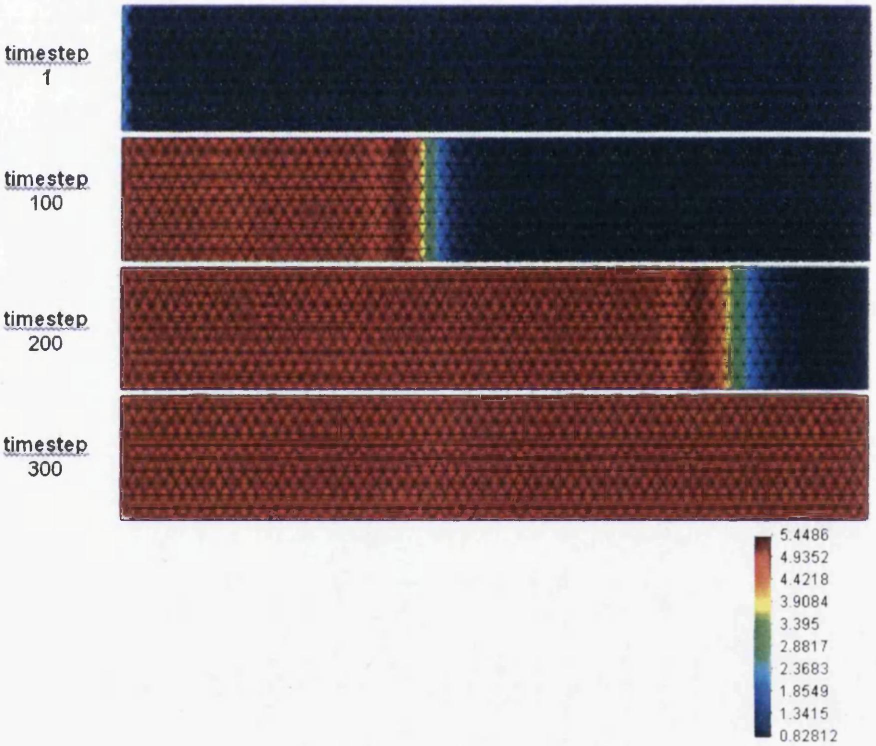


Figure 4.17: Convection of a Discontinuity using a Discontinuous Taylor–Galerkin Method (unstructured mesh)

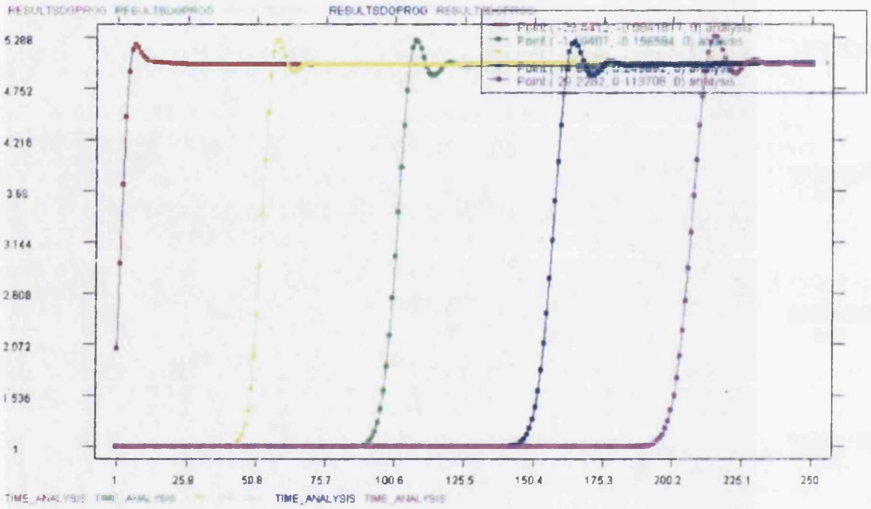


Figure 4.18: Point Analysis of the Discontinuity Convection using a Discontinuous Taybr–Galerkin Method (unstructured mesh)

4.2 Collisionless Boltzmann Examples

The remainder of the examples shown in this chapter were run on high performance parallel machines. This is because there is a significant increase in the computation performance requirement for the Boltzmann solver compared with the scalar convection solver used previously.

As described in Chapter 3, two approaches to the discretisation of the circular velocity space domain were investigated; the first approach was to use an assembly of linear, quadrilateral elements, whilst the second approach used a spectral quadrature approach. In this section, examples will be shown using both methods for comparison.

4.2.1 Gas Expansion Problem

A highly rarefied ($Kn >> 1.0$) gas of finite initial extent ($-L < x < L$) was modelled expanding into an infinite vacuum. This problem has a well defined analytical solution [5] and was, therefore, deemed a good test problem for the algorithm. A physical space mesh with 1538 elements / 4614 discontinuous nodes, shown in Figure 4.19, combined with a velocity space mesh with 900 elements / 961 nodes was used giving a total of 4,434,054 degress of freedom.

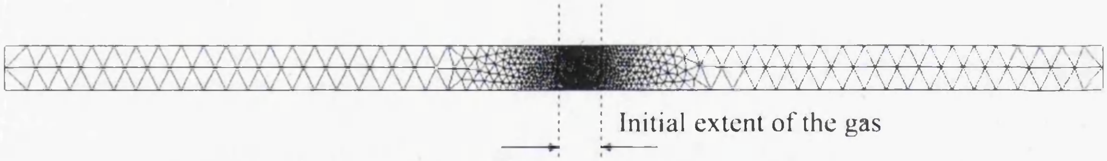


Figure 4.19: Physical Space Mesh for Gas Expansion Problem,Ω_r

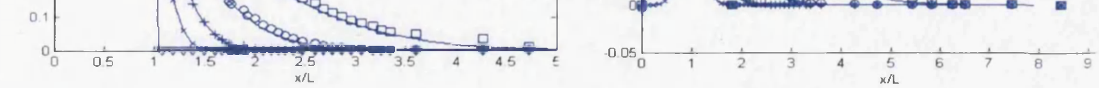


Figure 4.20: Results for The Gas Expansion Problem with Quadrilateral V-Space Elements

In the left–hand plot in Figure 4.20, the time evolution of the non–dimensionalised number density is shown on the y–axis against the non–dimensionalised distance from the vertical centreline of the domain. In the right–hand plot in Figure 4.20 the time evolution of a non–dimensionalised measure of momentum flux is shown on the y–axis against the non–dimensionalised distance from the vertical centreline. The solid lines show the 1D analytical solutions [5], given by equations (4.7) and (4.8), and the various markers are data points from the computational mesh. It is clear that a good agreement between the collisionless Boltzmann solver data and the analytical solution to this problem has been achieved.

$$\frac{n}{n_1} = [\mathrm{erf}(\beta_1(x+L)/t) - \mathrm{erf}(\beta_1(x-L)/t)]/2 \tag{4.7}$$

$$\frac{nu\beta_1}{n_1} = \frac{n\exp(-\beta_1^2(x-L)^2/t^2) - \exp(-\beta_1^2(x+L)^2/t^2}{n_1\pi^{\frac{1}{2}}[\mathrm{erf}(\beta_1(x+L)/t) - \mathrm{erf}(\beta_1(x-L)/t)]} \tag{4.8}$$

where $\mathrm{erf}(x)$ is the error function defined as $\mathrm{erf}(a) = (2/\pi^{\frac{1}{2}})\int_0^a \exp(-x^2)\mathrm{d}x$.

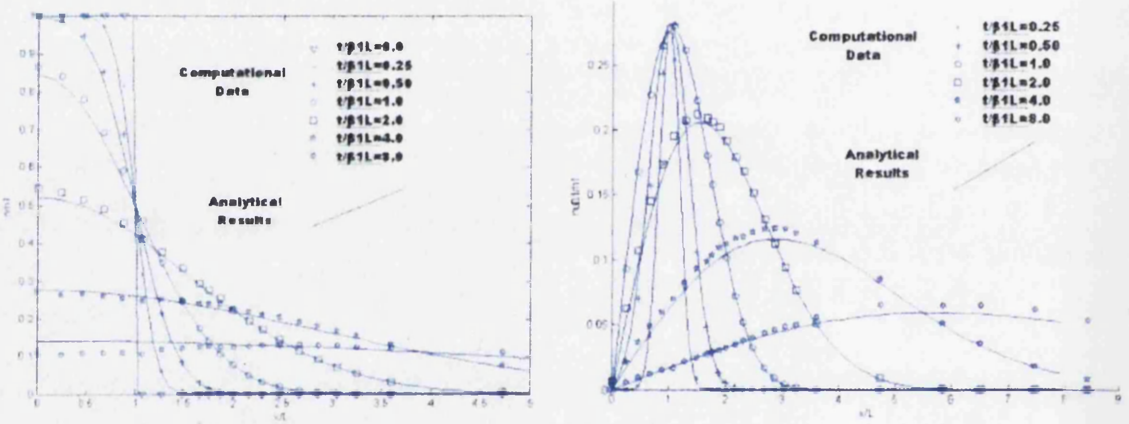


Figure 4.21: Results for The Gas Expansion Problem with Spectral V–Space

The same problem was considered using the spectral quadrature approach for v–space discretisation. The discretisation of v–space consisted of 900 sampling points and radial extent, $r_v = 2000m/s$. This is slightly less than the number of nodes in the previous example, which, combined with the improved integration efficiency of the quadrature method resulted in a runtime speedup of approximately 50%. The simulation was, therefore, run for an increased number of timesteps, up to a non–dimensional time, $t/(\beta_1 L) = 8.0$, where $\beta_1 = \sqrt{1/(2RT_1)}$. The results are shown in Figure 4.21 and, again, we see a good agreement between the computational data and analytical solution.

4.2.2 Flow Over a Vertical Plate

A rarefied, free–molecule ($Kn = 9$) gas was modelled, flowing through a 2D tunnel and over a plate of negligible thickness. The gas chosen was argon with a flow speed of $172m/s$, temperature of $300K$ and with a corresponding Mach number of 0.53. Fully specular reflection was assumed on the lower boundary and vertical plate.

A physical space mesh with 369 elements/1107 discontinuous nodes, shown in Figure 4.22, combined with a velocity space mesh with 900 elements/961 nodes was used giving a total of 1,063,827 degrees of freedom. The runtime to steady state for this case was approximately 1 hour using 4 processors on the Swansea University C2EC PC Cluster.

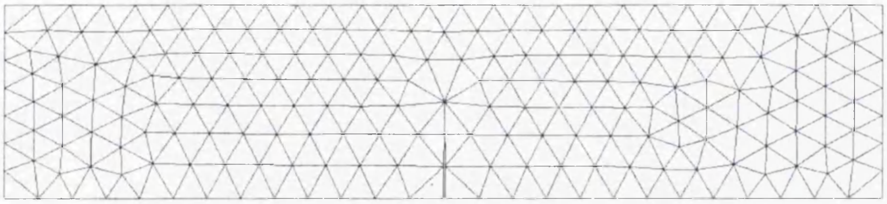


Figure 4.22: P–space mesh for the flow over a vertical plate

The resulting flow is seen, in Figure 4.23, to be symmetric about the plane of the plate. This agrees qualitatively with the DSMC simulation shown by Bird [5].

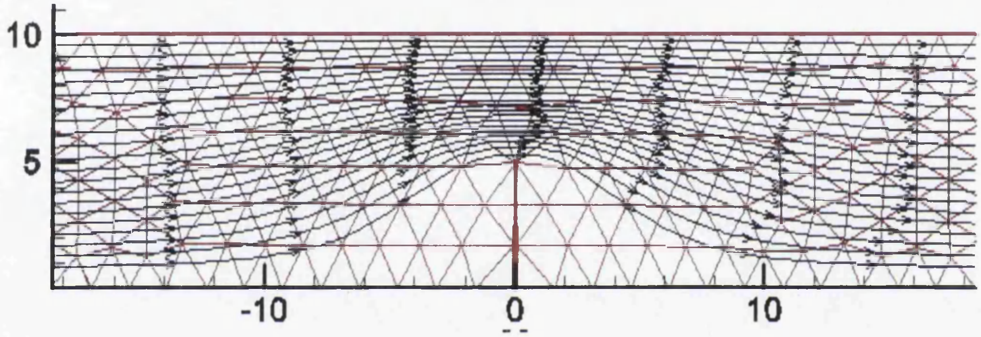


Figure 4.23: Streamtraces of flow over a vertical plate, quad v–space elements

The same problem was examined using the spectral quadrature approach for discretisation of v–space with 400 sampling points and $r_v = 2000m/s$. Figure 4.24 indicates that using the spectral method gives remarkably good agreement with the preceeding results using linear quadrilateral elements, despite the fact that under half the number of data points is used and, therefore, significantly quicker run times are achieved. The runtime to steady state was achieved in under 30 minutes, again using 4 processors on the Swansea University C2EC PC Cluster.

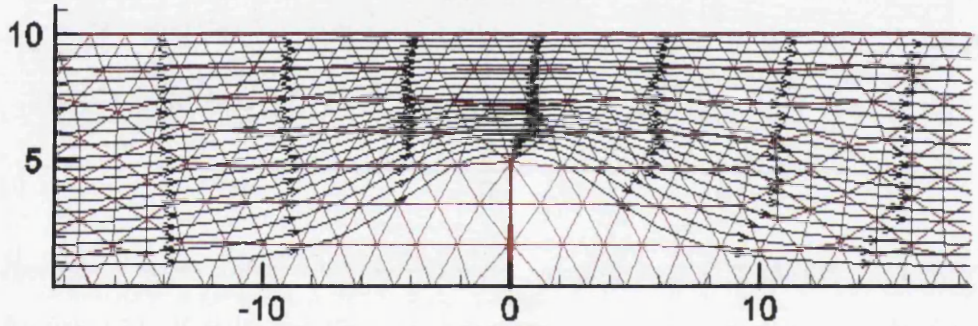


Figure 4.24: Streamtraces of flow over a vertical plate, spectral v–space

4.2.3 Hypersonic Flow over a Double Ellipse

During the early stages of re–entry, a space vehicle will travel at extremely high velocity through a very rarefied atmosphere. The high Knudsen number ($Kn >> 1$) of this flow allows it to be analysed using the collisionless Boltzmann equation. A typical re–entry vehicle, such as the space shuttle, will be required to decelerate from an orbital speed of 28,000 km/h or Mach 25 to allow it to descend into Earth's atmosphere and configure for landing. The space shuttle is manoeuvred, by firing thrusters, so that it approaches the top of Earth's atmosphere at a 40° angle of attack. A massive deceleration then occurs due to the bombardment of molecules and the nose is lowered into a standard flight configuration ([2],[42],[85],[40]).

A standard double ellipse geometry is used to model the nose region of a generic vehicle in two dimensions. Since these high Knudsen number runs assume a negligble effect due to molecular collisions, no shock will form and the solution will extend, in principle, infinitely far both upstream and downstream. The p–space mesh used for runs at a range of angles of attack from 40° to 20° is shown in Figure 4.25. It consists of 7,565 elements / 22,695 discontinuous nodes. Due to the extreme high velocities present in such a flow, the extent of the v–space domain had to be increased. By (numerical) experiment, it was determined that setting $r_v = 13,000m/s$ was sufficient. The number of sampling points in v–space was also increased from 400 to 1,000. This results in a value of 22,659,000 for the total number of degrees of freedom of the system. The positioning of the v–space sampling points is shown in Figure 4.26.

Table 4.1 summarises the flow parameters and computational requirements for the 3 cases studied. The free stream Knudsen number quoted is based on the maximum diameter of the double ellipse geometry cross section for the typical length scale of the flow.

In each of the three cases, the number density and Mach number distributions, constant pressure contours and streamtraces are shown. Each plot has been rotated such that the freestream flow is horizontal. It is first important to note that no shocks form despite the Mach number being greater than 1. This is because the formation of a shock wave is dependent on collisions between molecules which are insignificant at $Kn >> 1$. The flow solution must, therefore, vary continuously from regions where $M > 1$ to regions where $M < 1$.

The number density plots (Figures 4.27(a),4.28(a),4.29(a)) clearly show an order of magnitude increase in the molecular number density on the nose of the vehicle. The exact position of this region and its extent is clearly dependent on the angle of attack. The effect of the vehicle on the number density extends out into the upstream flow a distance of the same order of magnitude as the vehicle diameter.

Coupled with the number density increase on the nose, is a pressure increase, detailed in the plots of constant pressure contours (Figures 4.27(b),4.28(b),4.29(b)). It is this pressure increase that causes the vehicle to decelerate to a velocity at which it can configure for landing. Above the nose there appears to be 'spokes'

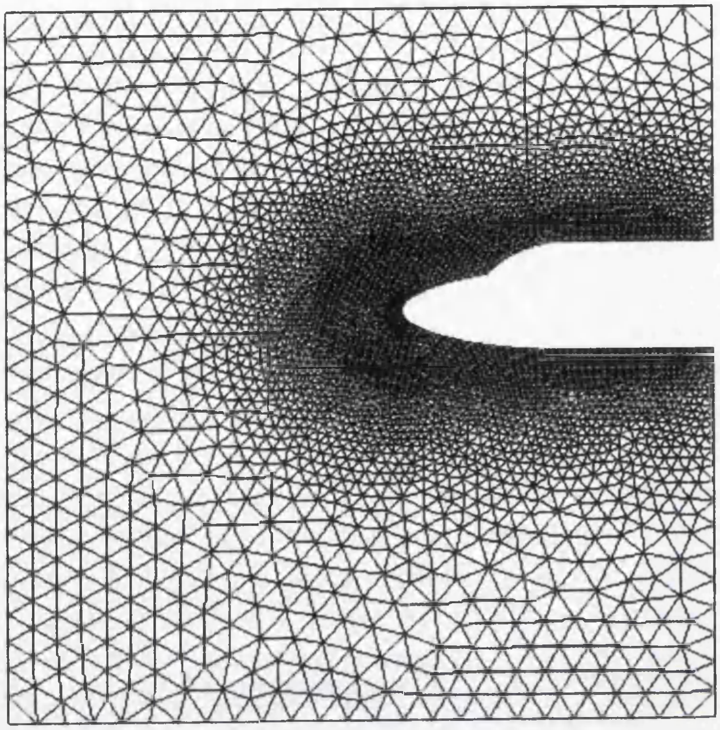


Figure 4.25: Double Ellipse Mesh

	Case 1	Case 2	Case 3
Flow Parameters			
M_∞	25	25	25
AoA	40°	30°	20°
Kn_∞	100	100	100
α	0.9	0.9	0.9
Computing Resources			
number timesteps to converge	15,000	15,000	15,000
runtime to converge	35hrs	35hrs	35hrs
number processors	4	4	4
computing facility	C2EC Cluster	C2EC Cluster	C2EC Cluster

Table 4.1: Details of Mach 25 collisionless flow over a double ellipse

in the pressure contour pattern and a 'jaggedness' in the contours upstream. Investigations discovered that the formation of the spokes and the non–smoothness in the contours is connected with the pattern of sampling points in the v–space discretisation, i.e. they are not real flow phenomena. Using the spectral v–space

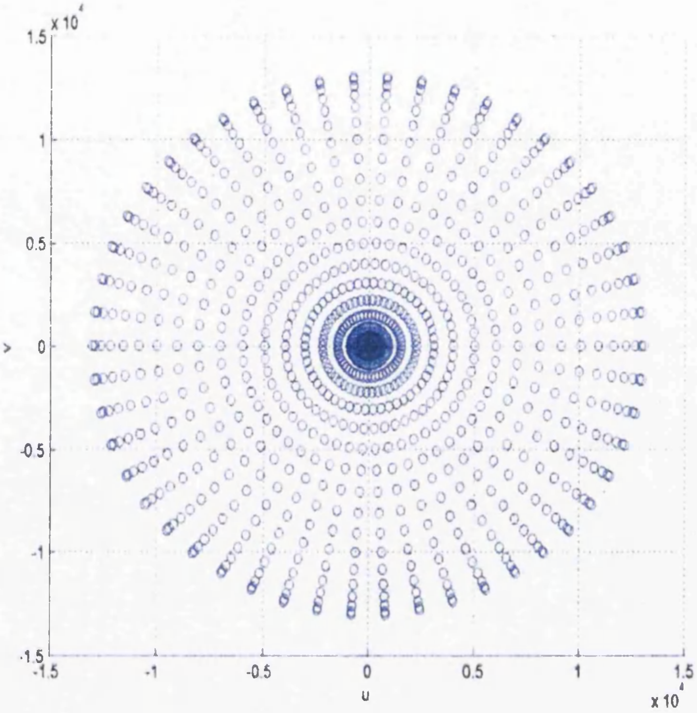


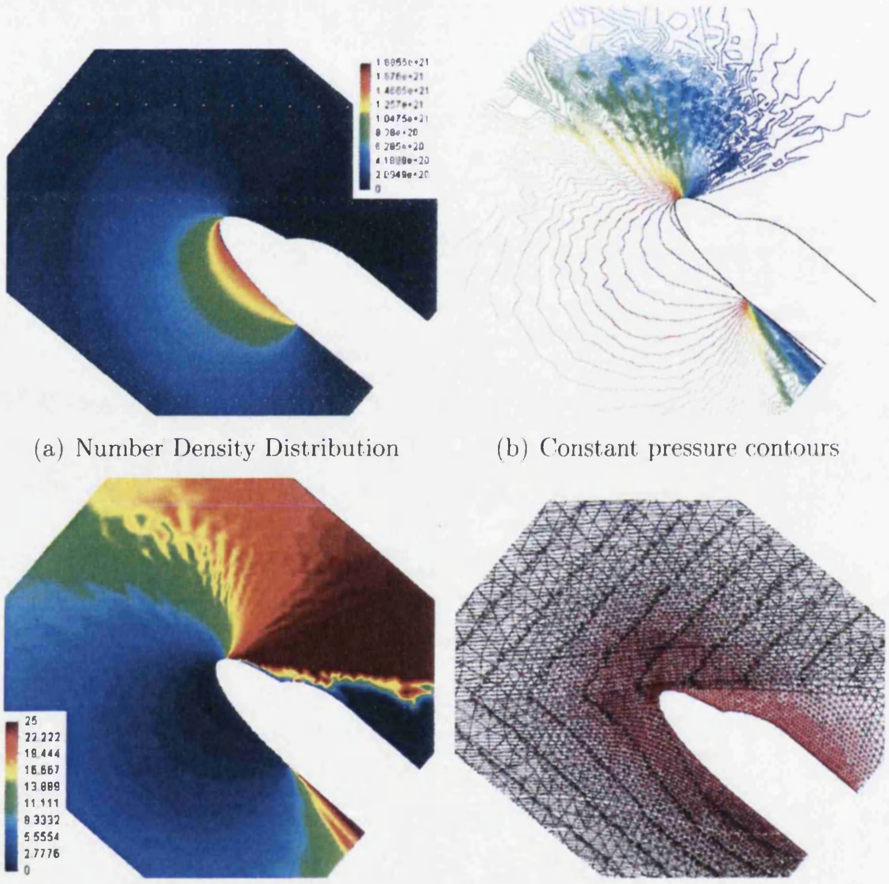
Figure 4.26: V–space sampling points for the collisionless hypersonic examples

discretisation approach used for this work, it is impossible to completely remove this numerical phenomenon, but minimising the tangential distance between sampling points in the far field of the v–space domain appears to help.

The Mach number plots (Figures 4.27(c),4.28(c),4.29(c)) show large regions of massively decelerated fluid upstream of the nose and in the wake region. It is interesting to note, that the flow is not accelerated above the free stream value at any point. The stagnant fluid in the wake is separated from the free stream fluid by a clearly defined shear layer or contact discontinuity. Across this horizontal layer, the pressure is continuous but velocity discontinuous. If collisions were present, the effect of viscosity would not allow such a phenomenon to exist. This contact discontinuity is stable in this case because no collisions means no momentum transfer across the layer. The diffusion that does exist, and shown in the Mach number plots, is purely numerical.

The streamtrace plots (Figures 4.27(d),4.28(d),4.29(d)) indicate that the flow direction remains very close to horizontal at all times. Mass conservation is achieved by means of massive density variations rather than streamline deflection. The position of the stagnation point is seen to move from the lower surface towards the upper surface as the angle of attack is reduced.

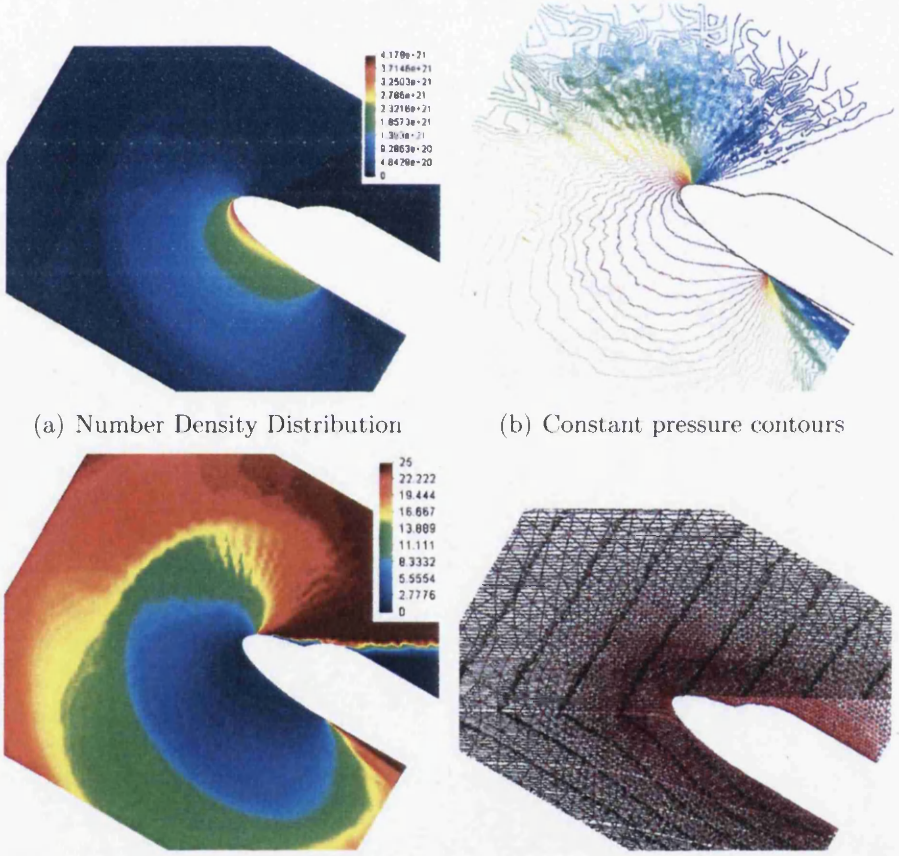
It is also worth noting that in case 3, at 20° angle of attack, a pressure and density increase are experienced on the cockpit bubble. This feature is not present for case 1 and case 2 at the higher angles of attack.



(a) Number Density Distribution (b) Constant pressure contours

(c) Local Mach number distribution (d) Streamtraces

Figure 4.27: Collisionless Hypersonic Case 1 ($M_\infty = 25$, $AoA = 40°$, $Kn \approx 100$)



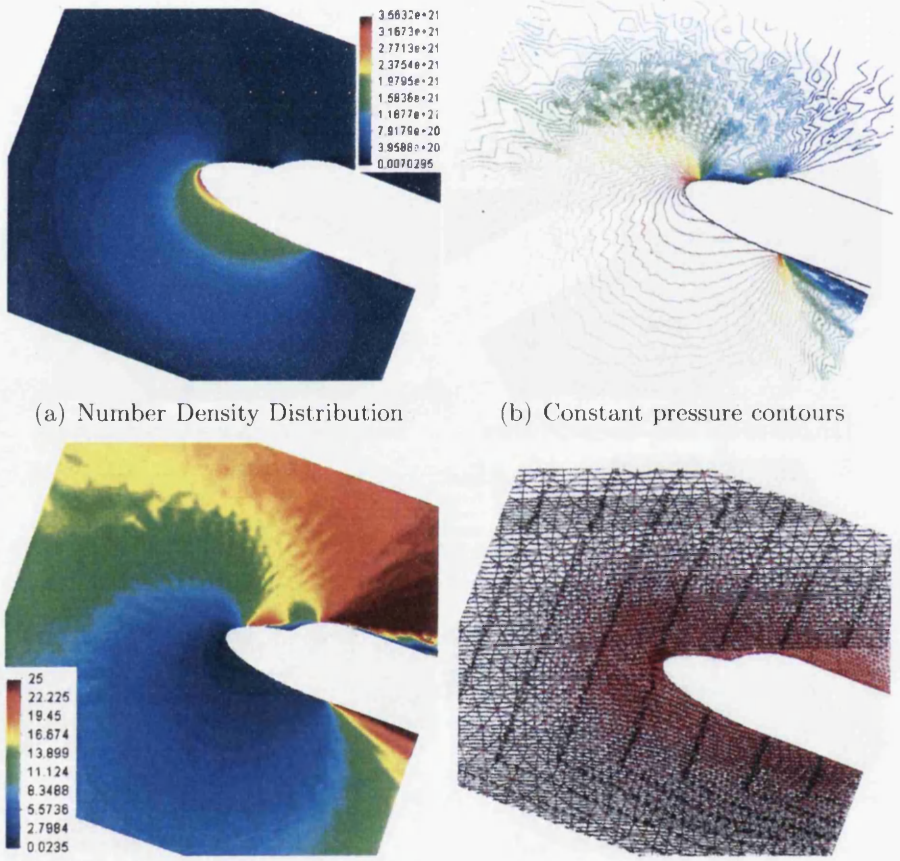
(a) Number Density Distribution

(b) Constant pressure contours

(c) Local Mach number distribution

(d) Streamtraces

Figure 4.28: Collisionless Hypersonic Case 2 ($M_\infty = 25$, $AoA = 30°$, $Kn \approx 100$)



(a) Number Density Distribution

(b) Constant pressure contours

(c) Local Mach number distribution

(d) Streamtraces

Figure 4.29: Collisionless Hypersonic Case 3 ($M_\infty = 25$, $AoA = 20°$, $Kn \approx 100$)

4.3 BGK Boltzmann Examples

4.3.1 Subsonic Flow Over a Vertical Plate

For free molecule flow, governed by the collisionless Boltzmann equation, all molecules travel in straight lines and, for the standard boundary conditions described in section 3.5, it is impossible for vortices to exist. For free molecule flow, we therefore get the streamtrace pattern shown in Figures 4.23 and 4.24. As the Knudsen number is reduced, however, the effect of molecular collisions becomes significant and we must introduce the right hand side collision term into the Boltzmann equation. In this section, the results shown are for the BGK approximation to the full collision term, detailed in section 3.4.1. As in the previous free molecule examples, the monotomic gas, argon, was modelled with the flow conditions at infinity set as $u = 172m/s$, $T = 300K$ with corresponding Mach number $M = 0.53$ at a variety of Knudsen numbers. Again, fully specular reflection was the assumed boundary condition on the lower boundary and on the vertical plate.

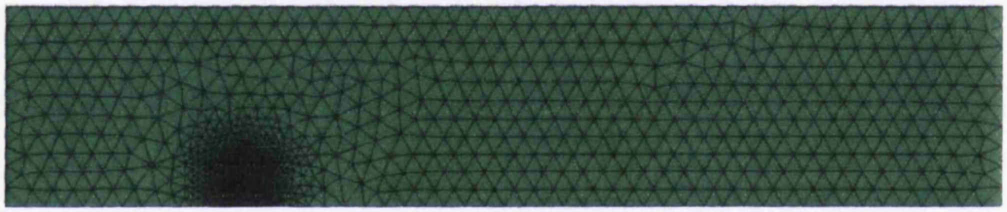


Figure 4.30: P-Space Mesh for BGK Flow Over a Vertical Plate

These examples were only considered using the spectral v–space approach as the results from the preceding sections had indicated that the spectral method results in a better efficiency than the quadrilateral element approach. The p–space mesh used for these examples consisted of 2163 elements/6489 discontinuous nodes and was refined in the region downstream of the plate, where the vortex formation was expected. The mesh is shown in Figure 4.30. The v–space discretisation consisted of 400 sampling points with $r_v = 2000m/s$. The runtimes to steady state for these examples using the BGK code were several hours using 4 processors on the Swansea University C2EC PC Cluster.

We see from Figures 4.31, 4.32 and 4.33 that as the Knudsen number reduces, and we move from the free molecule regime, through the transition regime and to the fully continuous or Navier–Stokes regime, with reference to Figure 1.2, the downstream vortices produced by separation at the top of the plate become more significant features. Initially, as the Knudsen number is reduced by an order of magnitude from the free molecule example in the previous section, we observe that the symmetry in the streamtraces is lost, but no vortex is formed. Reducing Kn by a further order of magnitude results in a small, low velocity vortex and a further decrease in magnitude results in a larger and higher velocity vortex. These results agree well with the same examples modelled by Bird [5] using a DSMC approach.

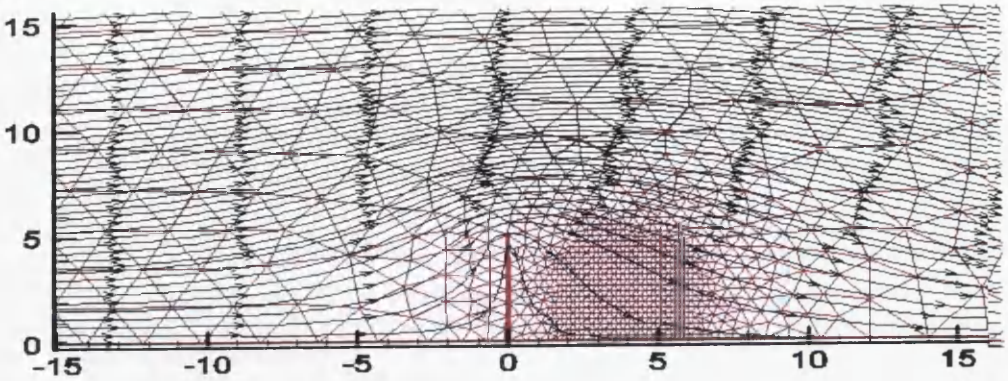


Figure 4.31: Vertical Plate Kn = 1.0

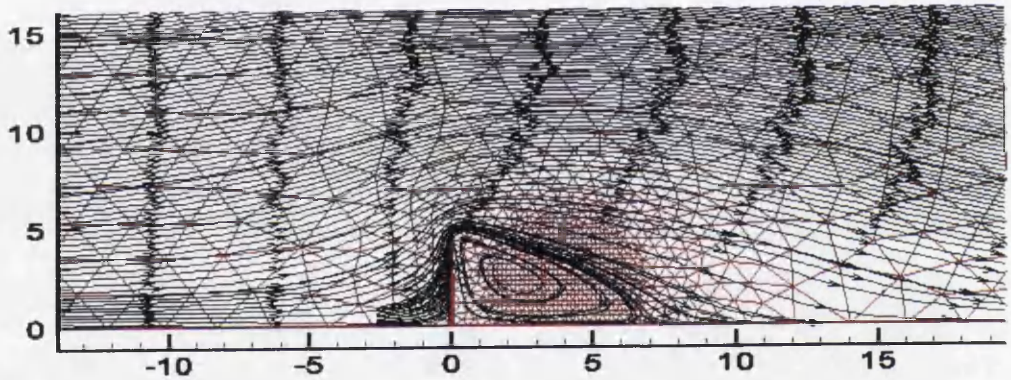


Figure 4.32: Vertical Plate Kn = 0.1

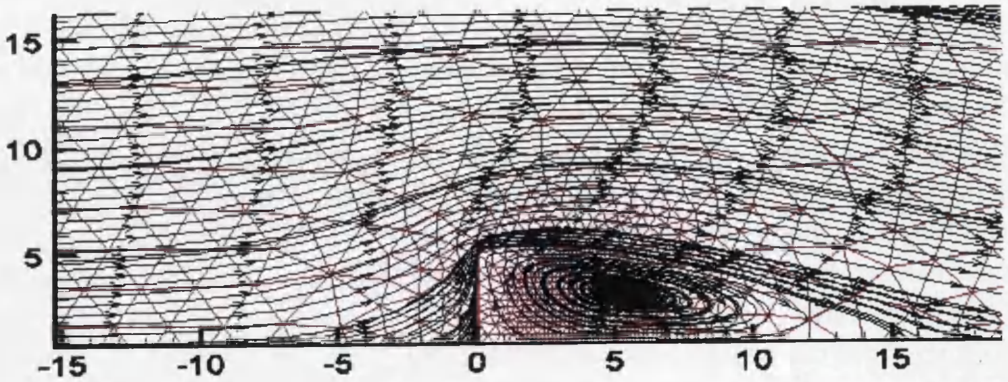


Figure 4.33: Vertical Plate Kn=0.01

4.3.2 Subsonic Flow over an Aerofoil (Investigation into the effect of the absorption coefficient, α)

The examples involving flow over a vertical plate indicated that the BGK solver was capable of predicting phenomena such as flow separation and recirculation zones. The next examples that were considered involved flow over an aerofoil cross section. The p–space mesh used to analyse these flows is shown in Figure 4.34 and consists of 7640 elements / 22920 discontinuous nodes. The velocity space was discretised using 400 sampling points and $r_v = 2000m/s$.

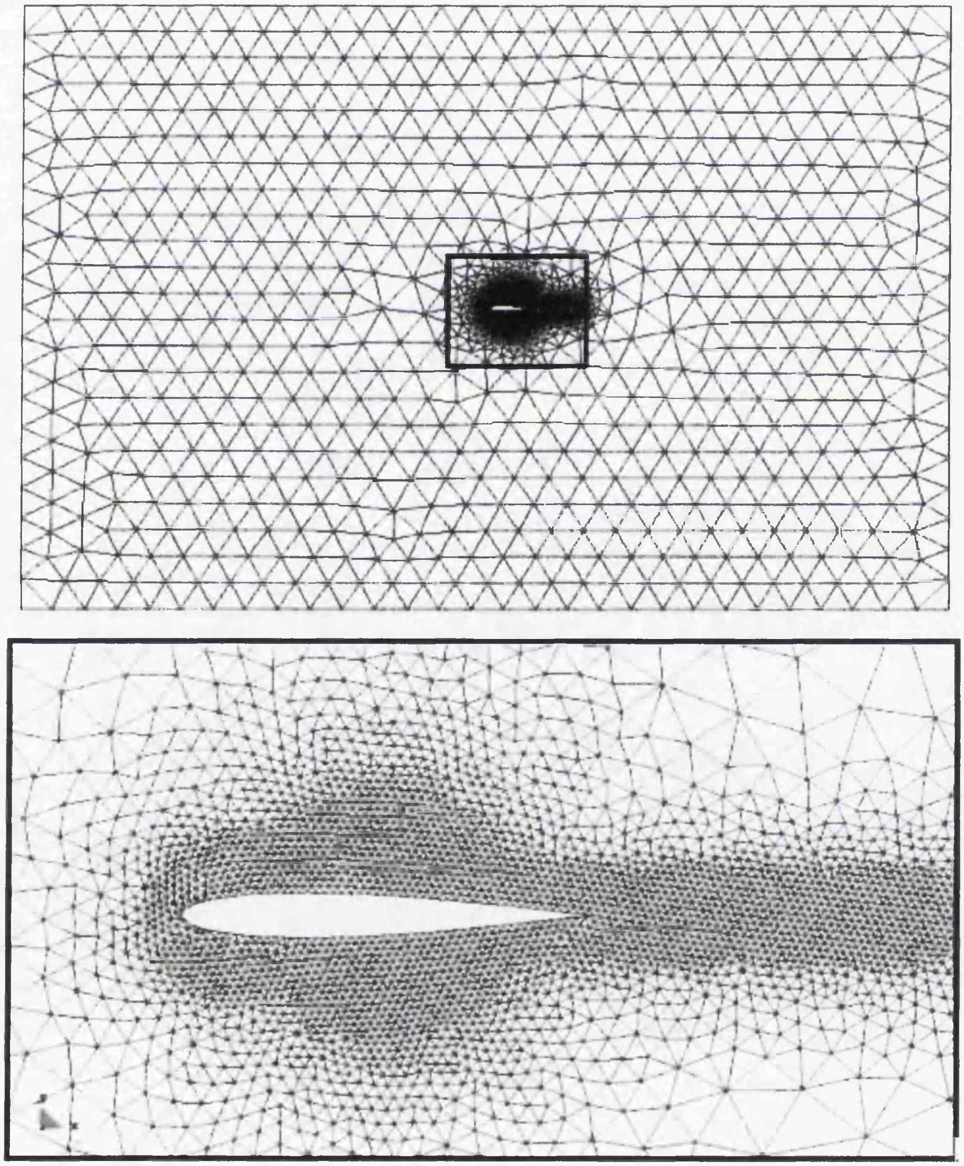


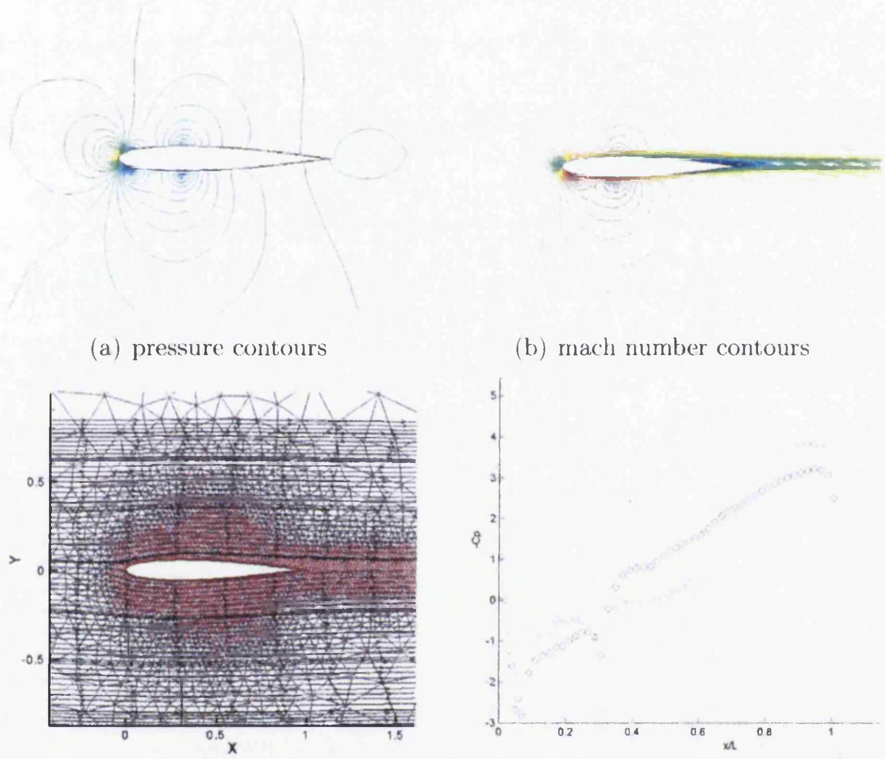
Figure 4.34: P–space mesh for subsonic/transonic NACA0012 cases

The wall boundary condition method was detailed in section 3.5.3 in which it was mentioned that two extreme cases for modelling the interaction of molecules with a solid surface is possible. The first was purely specular reflection (corresponding to the absoption coefficient, $\alpha = 0$) and the other was purely diffuse reflection (corresponding to $\alpha = 1.0$). Flow (of air) over a NACA0012 aerofoil [106] under subsonic conditions ($M_\infty = 0.5, AoA = 0°, Re = 10,000$) was considered under these two extreme cases and the results compared with Navier–Stokes solutions for the same flow conditions. Table 4.2 shows the flow conditions and computational requirements for the $\alpha = 0.0$ case. It is worth noting that the computational requirements using the BGK solver greatly exceed those for a solution to this problem using a standard Navier–Stokes approach. It is also important to note that even though air is being modelled, we are not analysing it in its multi–component form. We are simply using the gas constant for air in constructing the distribution function and an average diameter of a 'typical' air molecule in the collision term. Nevertheless, remarkably accurate predictions are possible and it is certainly feasible to extend the algorithm to consider a multi–component gas.

Flow Parameters	
M_∞	0.5
AoA	0°
Re	10,000
Kn_∞	0.0001
α	0.0
Computing Resources	
number timesteps to converge	45,000
runtime to converge	60hrs
number processors	4
computing facility	C2EC Supercomputing Cluster

Table 4.2: Details of subsonic case 1 ($\alpha = 0.0$)

The streamtrace plot shown in Figure 4.35(c) indicates that the correct general flow pattern is achieved (it also shows that mass conservation is being achieved at the aerofoil boundary). However, it is clear by comparison of Figures 4.35(a),4.35(b) and 4.35(d) with Navier-Stokes solutions (e.g. [108],[35]), that the more specific features of the flow are not captured correctly with a fully specular reflection boundary condition in place. For example, as the flow is at $AoA = 0°$, we would expect a symmetry about the chord of the aerofoil which is clearly not present. Also, Figure 4.35(d) indicates that there is a significant difference between the predicted pressure coefficient distribution on the top (red circles) and bottom (green circles) of the aerofoil which should not be the case. This might suggest that the specular reflection algorithm applied to the spectral quadrature v–space on boundaries that are not vertical or horizontal (as was the case in the vertical plate example), or to be more precise, that are not on a line of symmetry in the v–space discretisation does not perform well.



(a) pressure contours

(b) mach number contours

(c) streamtraces

(d) surface pressure distribution

Figure 4.35: Subsonic Case 1 ($M_\infty = 0.5$, $AoA = 0°$, $Re = 10,000$, $\alpha = 0.0$)

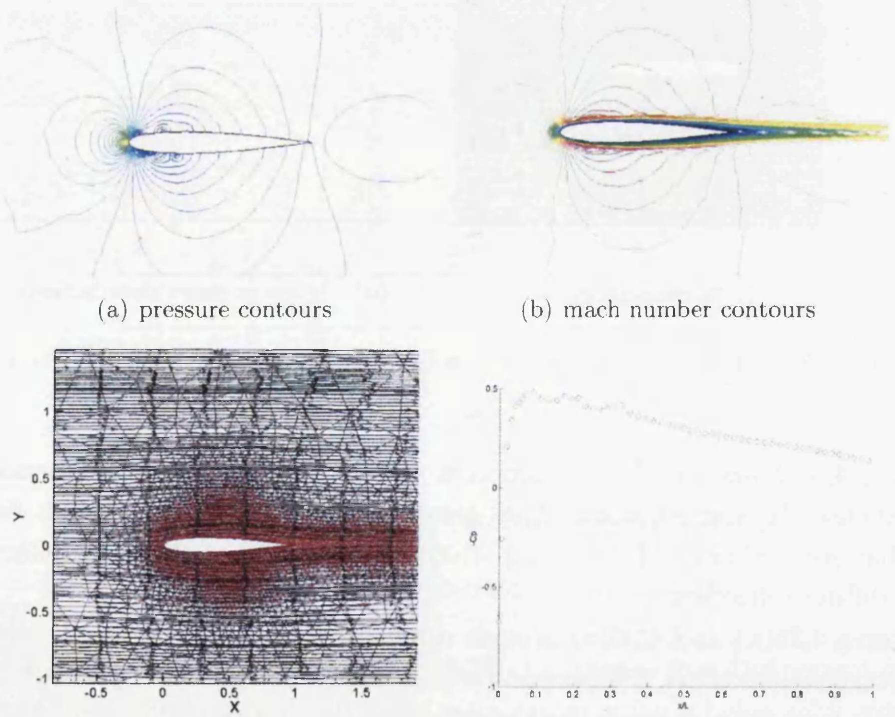
Table 4.3 shows the flow parameters and computational requirements for the $\alpha = 1.0$ case. It is worth noting that every parameter remains constant except for α so that we can get a direct comparison between purely specular reflection and purely diffuse reflection.

Figures 4.36(a) and 4.36(b) provide a much better agreement with solutions in the literature than the $\alpha = 0.0$ case. There are some non–symmetry features (particularly in the pressure contour plot) but these may be attributed to viscous boundary layer phenomena and/or non–symmetry in the unstructured p–space mesh. Again, the sreamtrace plot gives a good general flow pattern solution and indicates that mass is conserved at the aerofoil boundary. The surface pressure coefficient distribution was almost identical on the upper and lower surfaces and this distribution is shown in Figure 4.36(d). This agrees well with pressure coefficient plots generated using Navier–Stokes solvers in the literature.

It is clear that the Kutta–Joukowski condition is met. The pressure coefficient

Flow Parameters	
M_∞	0.5
AoA	0°
Re	10,000
Kn_∞	0.0001
α	1.0
Computing Resources	
number timesteps to converge	45,000
runtime to converge	60hrs
number processors	4
computing facility	C2EC Supercomputing Cluster

Table 4.3: Details of subsonic case 2 ($\alpha = 1.0$)



(a) pressure contours

(b) mach number contours

(c) streamtraces

(d) surface pressure distribution

Figure 4.36: Subsonic Case 2 ($M_\infty = 0.5$, $AoA = 0°$, $Re = 10,000$, $\alpha = 1.0$)

on the upper and lower surfaces merge at the leading edge leading to smooth flow. Also, Cp is close to unity at the stagnation point and approaches zero at the trailing edge. These are all features that are in excelent agreement with viscous aerofoil theory.

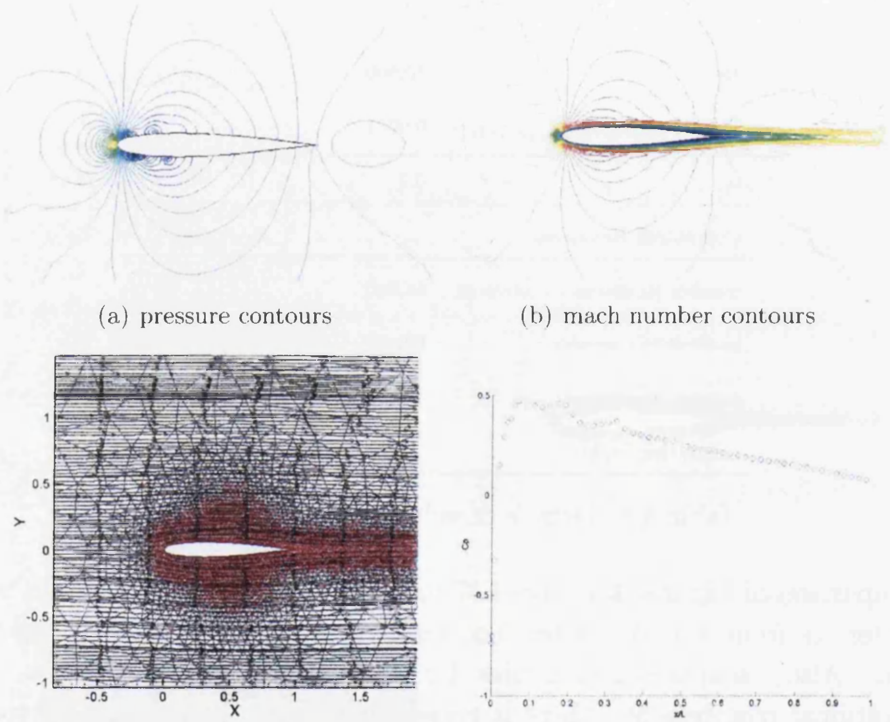
In [99], Trazzi and Pareschi note that in reality a combination of specular and diffuse reflection is the most realistic description of molecular interaction with a solid boundary. They note that a value 'close to 1' for α has been experimentally estimated as an appropriate figure. This subsonic case was therefore repeated with $\alpha = 0.9$ and all other parameters held constant (detailed in Table 4.4).

Flow Parameters	
M_∞	0.5
AoA	0°
Re	10,000
Kn_∞	0.0001
α	0.9
Computing Resources	
number timesteps to converge	45,000
runtime to converge	60hrs
number processors	4
computing facility	C2EC Supercomputing Cluster

Table 4.4: Details of subsonic case 3 ($\alpha = 0.9$)

Comparison of Figures 4.36 and 4.37 indicates that changing the wall boundary parameter, α from 1.0 to 0.9 has not made a significant difference on the flow solution. Also, comparison of Tables 4.2, 4.3 and 4.4 indicates that in terms of computational requirements, there is no advantage in running with any particular value for α. It was, therefore, decided to use $\alpha = 0.9$ for all further examples.

It is worth noting that to achieve the Boltzmann–BGK solutions here, a discretisation involving some 9,168,000 degrees of freedom has been used. If you compare this with the Navier-Stokes solutions obtained by Habashi [35] in which he uses a discretisation of only 6,685 degees of freedom for the same problem, it quickly becomes clear how expensive a Boltzmann solution for such applications is.



(a) pressure contours

(b) mach number contours

(c) streamtraces

(d) surface pressure distribution

Figure 4.37: Subsonic Case 3 ($M_\infty = 0.5$, $AoA = 0°$, $Re = 10,000$, $\alpha = 0.9$)

4.3.3 Transonic Flow over an Aerofoil

Transonic flow over a NACA0012 aerofoil was examined next using the BGK solver. Flow (of air) at a free stream mach number, M_∞ of 0.85 and an angle of attack, AoA of 2° was considered at Reynold's numbers, Re of 1,000 (case 1) and 10,000 (case 2). These two cases correspond to flow with global Knudsen number, Kn_∞ (based on the aerofoil chord length) 0.002 and 0.0002 respectively. This places both cases well within the continuum or NS regime with reference to Figure 1.2.

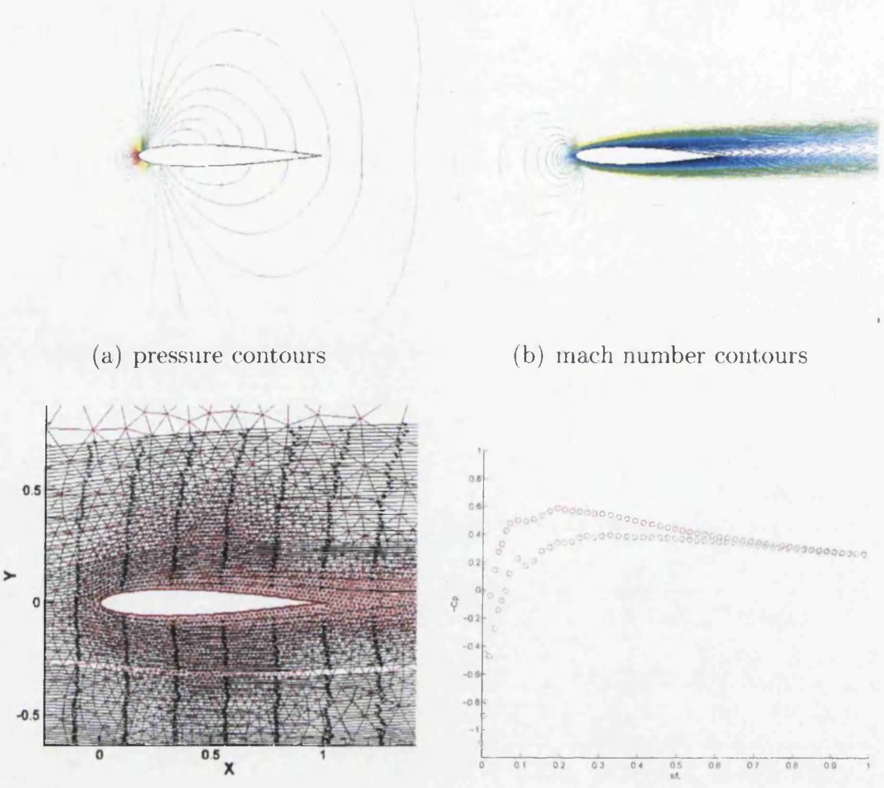
The p–space mesh used for case 1 is shown in Figure 4.34, consisting of 7640 elements / 22920 discontinuous nodes. A region of mesh refinement is positioned around the aerofoil and downstream in an attempt to capture the boundary layer features, viscous wake and any shocks that may be present. The velocity space was discretised using 400 sampling points and $r_v = 2000m/s$.

Table 4.5 gives details of the flow parameters and the computing resources required to run case 1 ($Re = 1000$).

Flow Parameters	
M_∞	0.85
AoA	2°
Re	1000
Kn_∞	0.0002
α	0.9
Computing Resources	
number timesteps to converge	10,000
runtime to converge	14hrs
number processors	4
computing facility	C2EC Supercomputing Cluster

Table 4.5: Details of transonic NACA0012 case 1

Figures 4.38(a) and 4.38(b) show plots of constant pressure contours and constant Mach number contours for the case 1 steady state solution. It is clear that the solution does not involve any shocks and a clearly defined downstream viscous wake is present. Also, the Mach contours indicate the presence of a thick boundary layer. Figures 4.38(c) and 4.38(d) show streamtraces for the flow and the surface pressure coefficient plot respectively (red circles represent the top surface and green circles represent the bottom surface). These results compare well with data in the literature for NS solutions for the same flow conditions. However, the runtime to convergence using the BGK solver is significantly longer than runtimes to convergence using a typical NS solver.



(a) pressure contours

(b) mach number contours

(c) streamtraces

(d) surface pressure distribution

Figure 4.38: Transonic Case 1 ($M_\infty = 0.85$, $AoA = 2°$, $Re = 1000$)

In case 2, the Reynold's number, Re was increased by one order of magnitude with all other parameters held constant. The run was initially undertaken using the same mesh configuration as for case 1 to determine the position of the standing shock on the upper aerofoil surface. Details of this run are shown in Table 4.6.

The run was then repeated on the p–space mesh shown in Figure 4.39. This mesh has 10749 elements / 32247 discontinuous nodes. The mesh has an increased level of refinement for boundary layer capture and a refinement zone positioned to capture the shock formed on the upper surface. The position of this refined region was determined from an initial run on the p–space mesh in Figure 4.34. When combined with a v–space mesh of 400 sampling points (again, $r_v = 2000m/s$), the total number of degrees of freedom in this problem was 12,898,8800.

The details of this run are shown in Table 4.7. It is immediately clear that the introduction of mesh refinement has lead to a significant increase in computational demand. The number of processors, and hence, computational power is doubled and the runtime to convergence is an order of magnitude larger than when using the mesh in Figure 4.34.

The pressure contour and Mach number plots for case 2 (Figures 4.40(a) 4.40(b)) indicate that increasing Re has resulted in the formation of a standing shock on the

Flow Parameters	
M_∞	0.85
AoA	2°
Re	10000
Kn_∞	0.002
α	0.9
Computing Resources	
number timesteps to converge	50,000
runtime to converge	70hrs
number processors	4
computing facility	C2EC Supercomputing Cluster

Table 4.6: Details of transonic NACA0012 case 2 (run 1)

Flow Parameters	
M_∞	0.85
AoA	2°
Re	10000
Kn_∞	0.002
α	0.9
Computing Resources	
number timesteps to converge	120,000
runtime to converge	480hrs
number processors	8
computing facility	C2EC Supercomputing Cluster

Table 4.7: Details of transonic NACA0012 case 2 (run 2)

upper surface of the aerofoil. It is also evident that the boundary layer thickness has decreased. This is in agreement with an analytical analysis of the Prandtl equations for a flat plate boundary layer that indicates that boundary layer thickness, δ is proportional to the square root of the inverse of Re ($\delta \propto \sqrt{\frac{1}{Re}}$). Again, a viscous wake region is present downstream of the aerofoil. The streamtrace plot in Figure 4.40(e) shows a significant region of non–laminar flow on the upper surface downstream of the shock and in the wake. This region is more pronounced than in case 1, another flow feature that is in agreement with theory (see [10]). Although the general trends

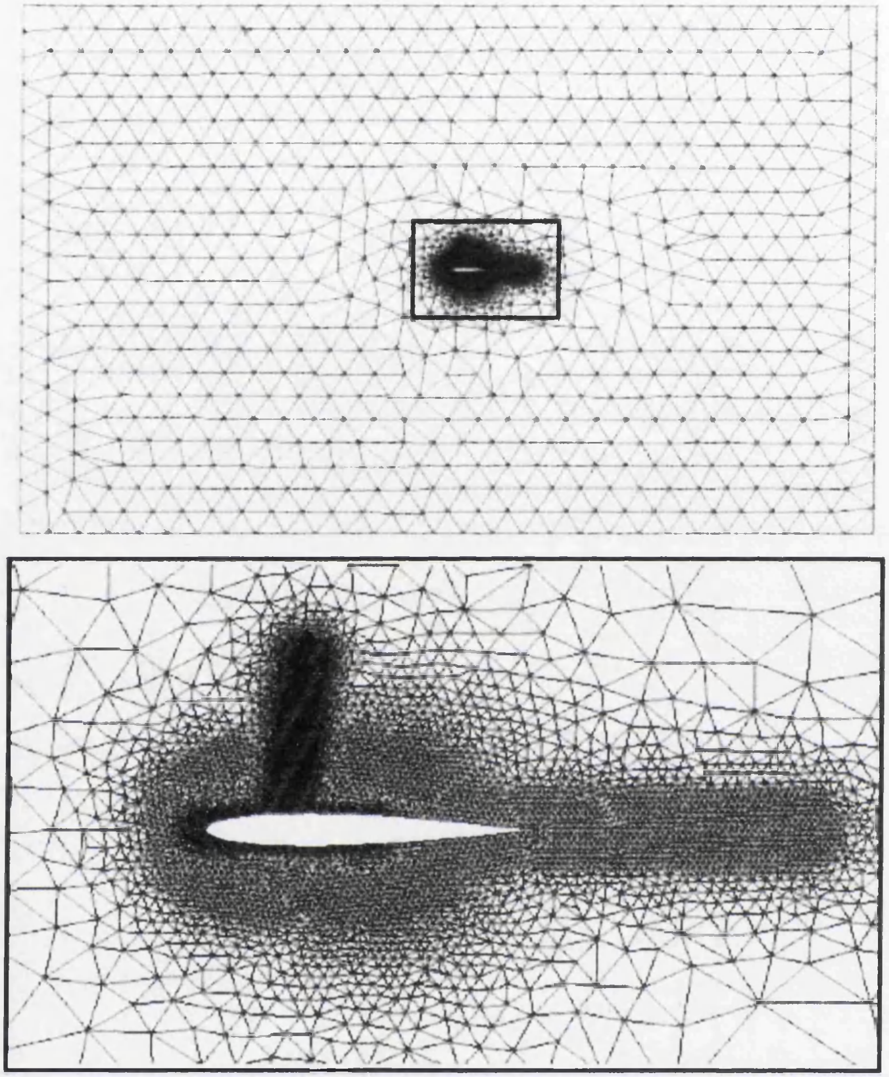
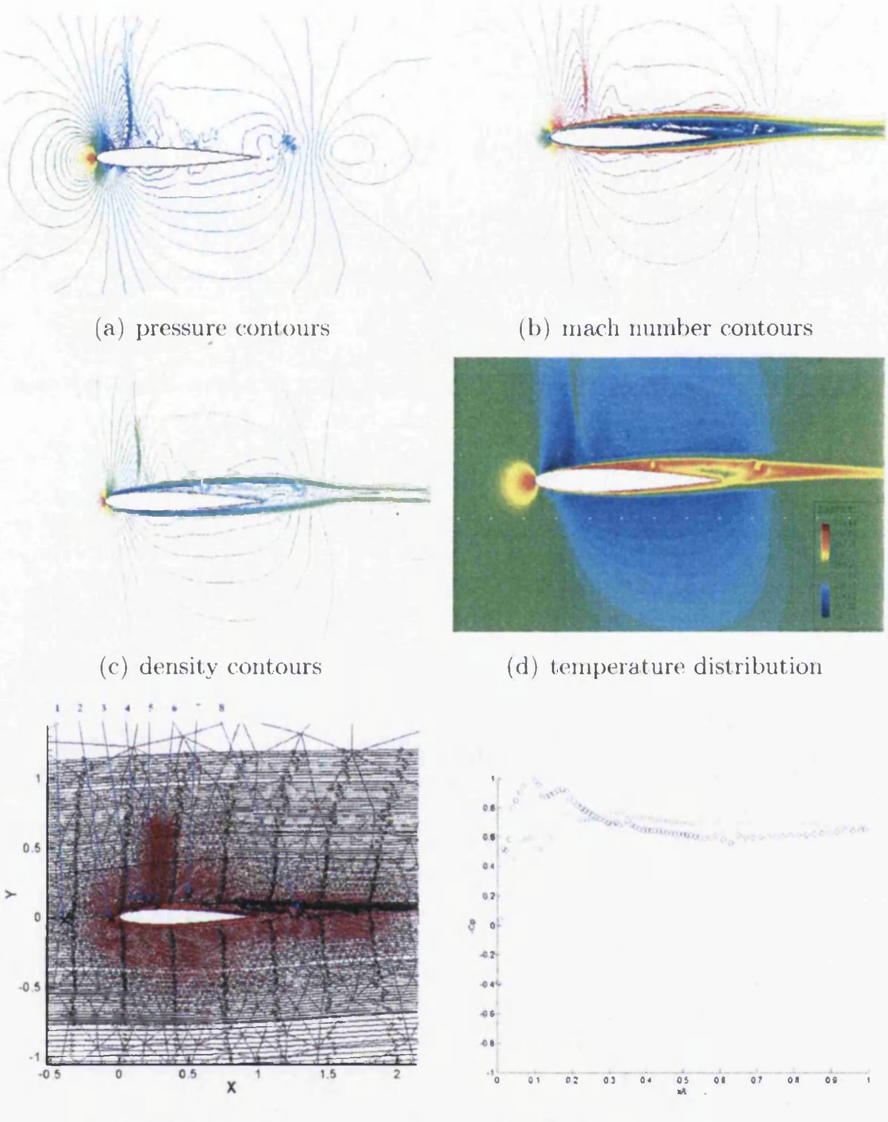


Figure 4.39: Refined p–space mesh for transonic NACA0012 cases

in the surface Cp plot, Figure 4.40(f), are correct, when compared with typical NS solutions for such a flow, the shock is significantly smeared in this BGK–Boltzmann solution. If the pressure contour plot is examined, this smearing appears to be associated with the shock–boundary layer interaction. This phenomenon is not encountered in NS solutions of such a flow. Also, significant oscillations are present in the Cp solution. It is unclear whether these oscillations are real, physical features of the flow, numerical instabilities or a failure of the BGK collision term under such flow conditions.

In this example, the underlying distribution function has been plotted at various points within the flow domain in Figure 4.41. The two plots at each point show the distribution function profile and planform contours of the distribution function using identical scales for each of the axes. The numbering of points in Figure 4.41



(a) pressure contours

(b) mach number contours

(c) density contours

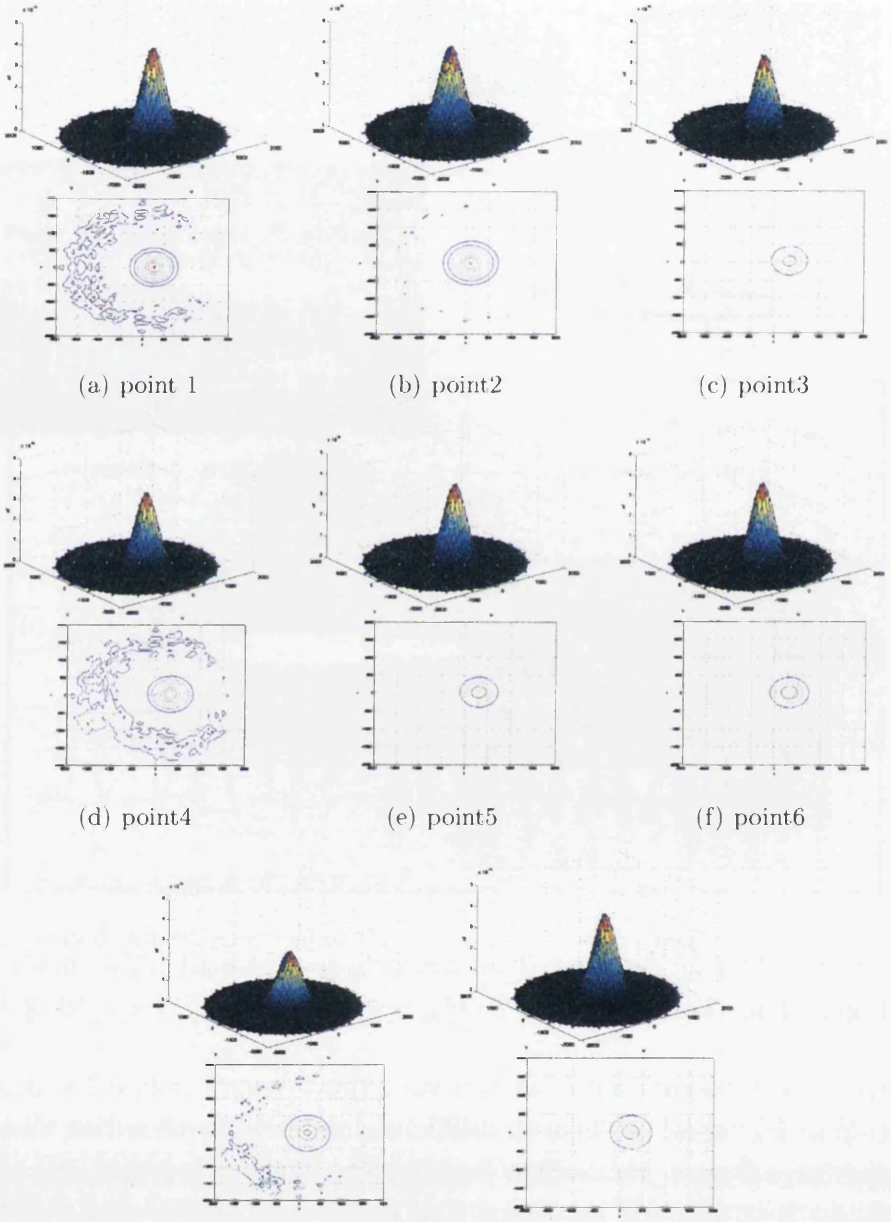
(d) temperature distribution

(e) streamtraces

(f) surface pressure distribution

Figure 4.40: Transonic Case 2 ($M_\infty = 0.85$, $AoA = 2°$, $Re = 10,000$)

relates to position labels in Figure 4.40(e). It is immediately clear that there are no large departures from a Maxwellian form for the distribution function, even close to the shock. It is, perhaps, surprising that such a complex flow pattern can result from such minor variations in the underlying distribution function. One interesting point to note is that all of the points that feature 'distortion' of the distribution function in the far field are outside of regions of the flow where viscous effects are significant.



(a) point 1

(b) point2

(c) point3

(d) point4

(e) point5

(f) point6

(g) point7

(h) point8

Figure 4.41: Velocity distribution functions at selected points in the flow field

4.3.4 Supersonic Flow over an Aerofoil

To complete the validation study of the BGK solver, supersonic flow over a NACA0012 aerofoil was examined. The freestream Mach number, M_∞ was increased to 1.5 at $AoA = 0°$ and $Re = 10,000$ ($Kn_\infty = 0.002$). The p–space mesh used was the same as that in the subsonic and preliminary transonic cases (shown in Figure 4.34) with 7640 elements / 22920 discontinuous nodes. Again, v–space is discretised with 400 sampling points and $r_v = 2000 m/s$. A summary of flow parameters and computational requirements is shown in Table 4.8.

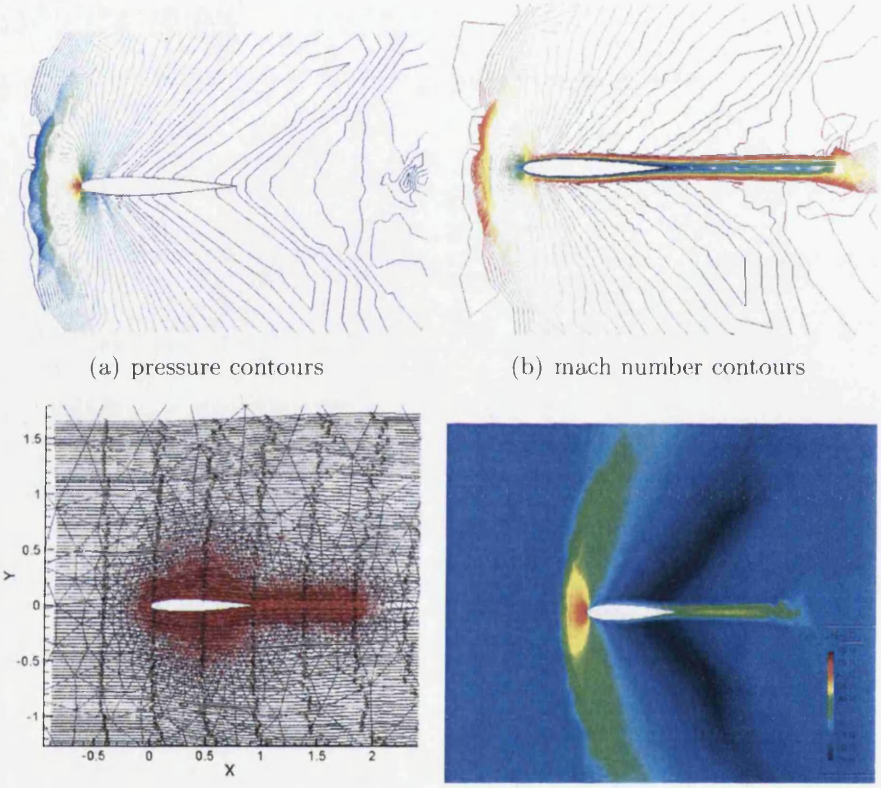
Flow Parameters	
M_∞	1.5
AoA	0°
Re	10000
Kn_∞	0.002
α	0.9
Computing Resources	
number timesteps to converge	30,000
runtime to converge	42 hrs
number processors	4
computing facility	C2EC Supercomputing Cluster

Table 4.8: Details of supersonic NACA0012 case

The pressure contour and Mach number contour plots, Figures 4.42(a) and 4.42(b) respectively, for the fully supersonic case indicate that the solution includes a weak detatched bow shock with subsonic flow downstream. The shock has been smeared over a few elements (see Figure 4.34). A crisper shock capture could be achieved by refining the mesh in the shock position indicated by this solution. The Mach number contour plot also indicates a viscous boundary layer and wake region. This wake region begins to get smeared in the far right of the plot in the region where element size increases (again, with reference to Figure 4.34). This example clearly demonstrates the importance of using an appropriate mesh refinement to achieve an accurate solution. Both the pressure contour and Mach number plots agree well with Navier–Stokes solution for this flow (by, for example, Hafez and Wahba [108]). The streamtrace plot; Figure 4.42(c), shows no significant deflection of the flow at the bow shock which is to be expected for flow across a weak, approximately normal shock wave. The temperature distribution plot, Figure 4.42(d), does indicate a significant temperature increase across the shock. The temperature jump immediately across the shock is given by $\frac{T_{downstream}}{T_\infty} \approx \frac{380}{294} = 1.29$. We can compare the BGK estimation of the temperature jump with the isentropic normal shock relationship for a calorifically perfect gas,

$$\frac{T_{downstream}}{T_\infty} = \frac{[2\gamma M^2 - (\gamma - 1)][(\gamma - 1)M^2 + 2]}{(\gamma + 1)^2 M^2} \tag{4.9}$$

which predicts a temperature jump of $\frac{T_{downstream}}{T_\infty} = 1.32$. This is a very good agreement and again, helps confirm that the BGK solver is a suitable method for such flows. The temperature distribution plot also indicates significant viscous heating in the nose region and wake.



(a) pressure contours

(b) mach number contours

(c) streamtraces

(d) temperature distribution

Figure 4.42: Supersonic Case ($M_\infty = 1.5$, $AoA = 0°$, $Re = 10,000$)

High Mach Number Problems using the BGK Solver

The supersonic case was repeated for higher Mach numbers again using the BGK algorithm. However, above $M_\infty = 2.0$ an instability was observed originating in the vicinity of the bow shock. The problem manifests itself in the post–processing stage as an instability in the macroscopic variables. Analytical BGK solutions (described by Bird in [5]) also fail for Mach numbers greater than 2. The explanation for this is that for strong shocks, a region of significantly non–equilibrium flow exists in the vicinity of the shock. The stronger the shock, the further the distribution function is perturbed from its Maxwellian form. Under such conditions, the BKG approximation to the collision term fails to be sufficient to return the distribution function to equilibrium because of the use of a collision frequency that is independent of f. We should not, therefore, be surprised that the BGK numerical solver fails at high Mach number since the assumption upon which it is based is that perturbations from equilibrium remain small. The full Boltzmann solver was, therefore, developed as an attempt to overcome this problem.

4.4 Full Boltzmann Examples

4.4.1 Validation Study

Using the Hard Sphere Model

Running the full Boltzmann algorithm using a hard sphere model for molecular interaction gave results that were identical to those using the collisionless Boltzmann algorithm. This phenomena, at first, appears paradoxical until we look closely at the pre– and post–collision velocities during binary collisions whilst remembering that when simulating a monatomic gas, all particles are essentially identical. Bird [5] shows that for hard sphere molecules, the collision cross section, σ_T, is independent of the change in direction of the relative velocity vector, χ, and that the scattering of molecules is isotropic in the centre of mass reference frame. This means that all directions are equally likely for the post–collision relative velocity vector. Since all molecules are identical, in terms of mass and size, when integrating over a large number of collisions, it is as if molecules are simply trading places during collisions and, hence, no net effect is observable.

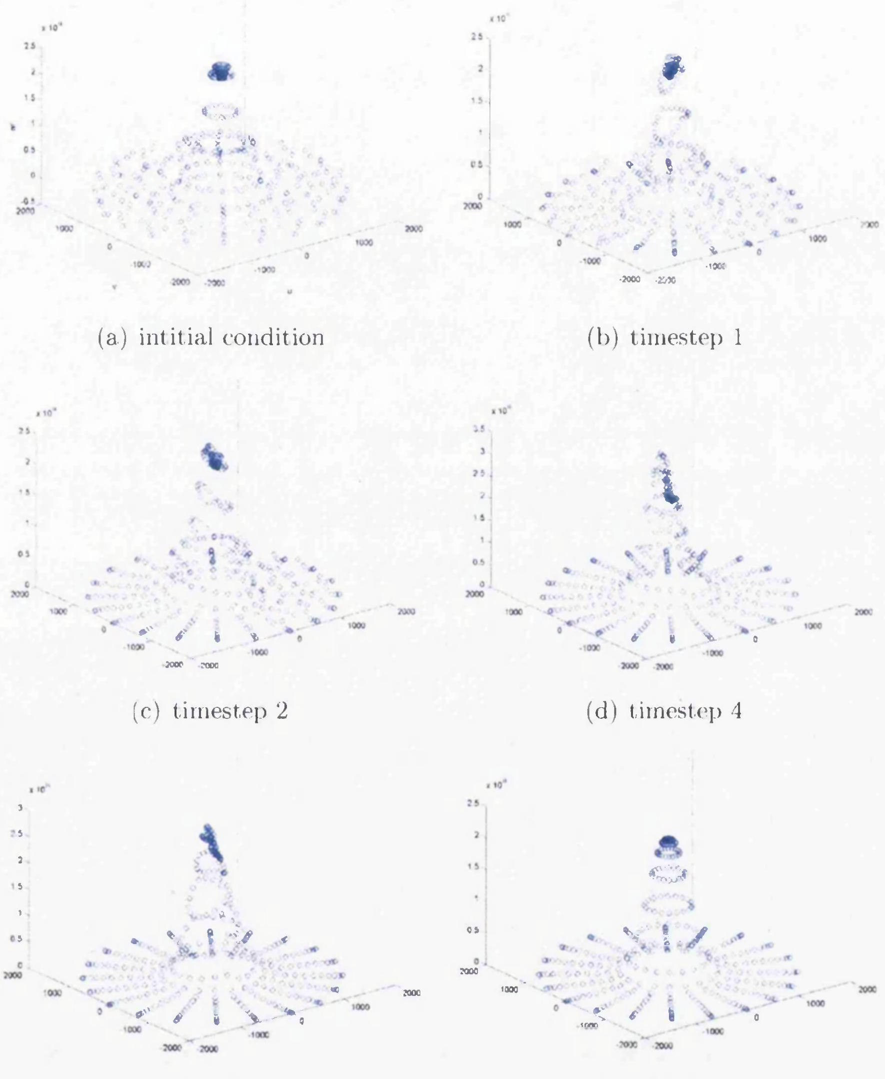
A more sophisticated molecular interaction model was therefore required. This led to the development of the algorithm by employing Bird's variable soft sphere model for molecular interaction. This was achieved using a Monte Carlo sampling technique for the distance of closest approach.

Using the Variable Soft Sphere Model

An important property of the collision term is that it must return a strongly non–equilibrium distribution function to Maxwellian form. This process was the assumed situation in the simplified BGK collision term. However, an important validation test for the full Boltzmann collision term, in which this equilibrium restoration property is not explicitly assumed, is to ensure that the collision term does, in fact, restore a non–equilibrium distribution function initial condition to equilibrium in a spatially homogenous case. The non–equilibrium initial condition was established by superimposing a sinusoidal perturbation, in the radial direction, to the standard Maxwellian distribution function, i.e. $f(\boldsymbol{c}) = f_0(\boldsymbol{c}) + c\sin(10\pi(r/r_v))$, with $T = 294K$, $p = 50Pa$ and c, some non-dimensional constant. This initial condition is shown in Figure 4.43(a), in which, the value of (nf) is shown at each of the sampling points in v–space. The evolution towards equilibrium, driven by the Boltzmann collision term is then depicted in Figures 4.43(b) - 4.43(e).

It is clear that the distribution function returns to a standard Maxwellian equilibrium form. Also, the conservation of momentum and energy in each Monte Carlo sampled collision has ensured that the integral of (nf) over v–space has remained constant, and hence, mass is conserved.

In section 3.4.3, it was shown that the VSS model requires specification of the parameters d_{ref}, $c_{r,ref}$ and ν. d_{ref} was chosen as 250x$10^{-12}m$, a typical estimation of



(a) intitial condition

(b) timestep 1

(c) timestep 2

(d) timestep 4

(e) timestep 8

(f) timestep 16

Figure 4.43: Evolution of distribution function to Maxwellian

the molecular cross–section of an oxygen molecule, $c_{r,ref}$ was chosen as $10,000m/s$, an estimation of the typical maximum relative velocity of the pre–collision velocity components in a binary collision, and ν was chosen to be 0.5, as suggested by Koura and Matsumoto ([65],[66]).

4.4.2 Full Boltzmann Application to Flow over a Double Ellipse

The full Boltzmann solver was applied to the generic re–entry vehicle double ellipse geometry at a selection of subsonic and supersonic Mach numbers and $Kn \sim 1.0$. This Knudsen number places the problem in the transition regime between free molecule flow and fully continuous flow (see Figure 1.2). The same mesh as in the collisionless hypersonic case of section 4.2 was used. The size of the v–space discretisation was limited to 400 sampling points and $r_v = 4000m/s$ due to the large computational demand of this solver. The 7,565 elements / 22,695 discontinuous nodes in p–space combined with 400 sampling points in v–space gives this problem 3,000,000 degrees of freedom. Details of the three cases studied is shown in Table 4.9.

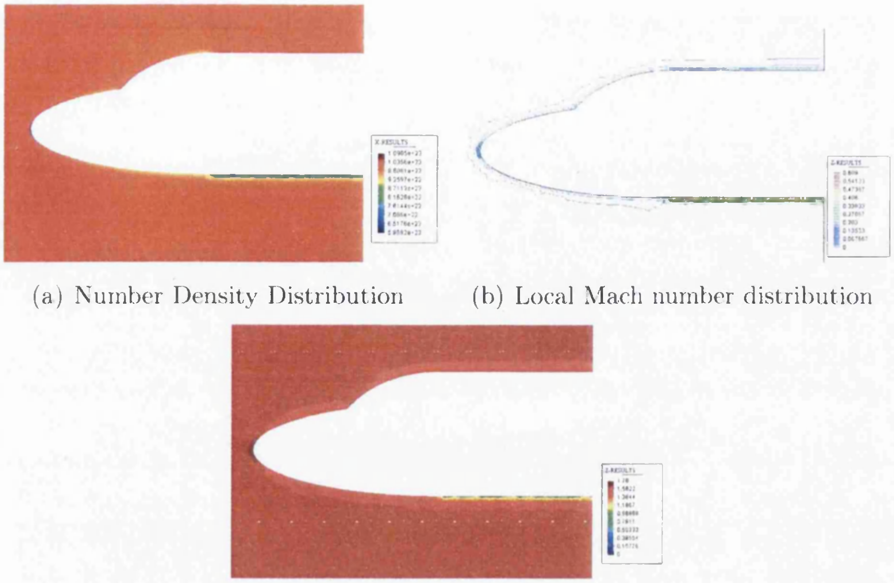
	Case 1	Case 2	Case 3
Flow Parameters			
M_∞	0.2	1.1	2.3
AoA	0°	0°	0°
Kn_∞	1	1	1
α	0.9	0.9	0.9
Computing Resources			
number timesteps to converge	10,000	10,000	10,000
runtime to converge	230 hrs	230 hrs	230 hrs
number processors	32	32	32
computing facility	BLUE C	BLUE C	BLUE C

Table 4.9: Details of Transition Regime Flow over a Double Ellipse

It is immediately clear from Table 4.9, that the full Boltzmann solver computational demand, in terms of runtime, is at least an order of magnitude larger than the BGK–Boltzmann solver. To achieve convergence even on these relatively simple problems required running for almost 10 days on 32 processors with the parallelisation methodology currently employed. This is obviously a considerable drawback and limiting factor in terms of the development of this approach.

Due to the large run times, only a small selection of transition regime flow cases over the generic re–entry vehicle geometry were studied at a global Knudsen number, based on the maximum vehicle width, of approximately 1.0. This means that typical molecular mean free paths are of the same order as the typical vehicle length scale.

A low subsonic case, and two supersonic cases are shown in Figures 4.44(a) to 4.46(c) detailing the number density, Mach number and pressure coefficient distributions immediately around the vehicle. The first observation to make is that significant gradients in the variables considered here are 'trapped' in a small boundary–layer type feature around the vehicle surface. All cases show a stagnation bubble at the nose and cockpit, and a region of accelerated flow on the bottom surface accompanied by a drop in density and pressure. It is also evident that the molecule–wall interaction model is allowing slip at the wall boundary, as is to be expected at this

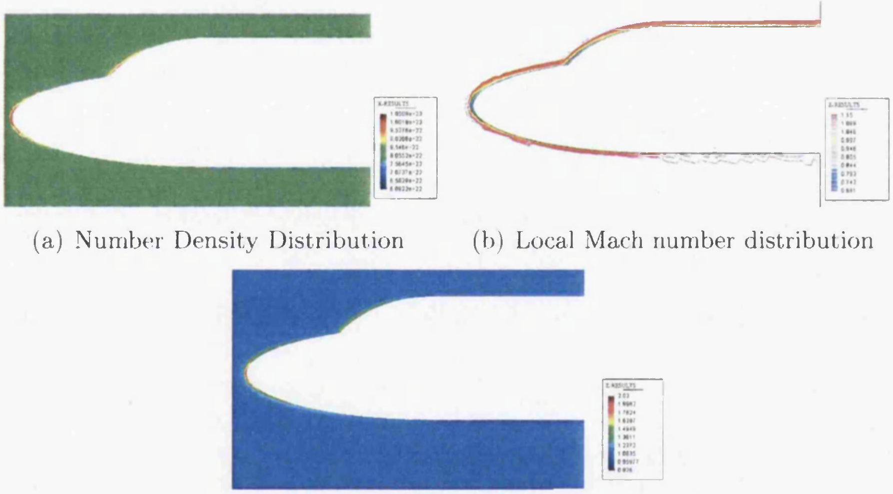


(a) Number Density Distribution

(b) Local Mach number distribution

(c) pressure coefficient

Figure 4.44: Full Boltzmann Case 1 ($M_\infty = 0.2$, $AoA = 0°$, $Kn \approx 1.0$)



(a) Number Density Distribution

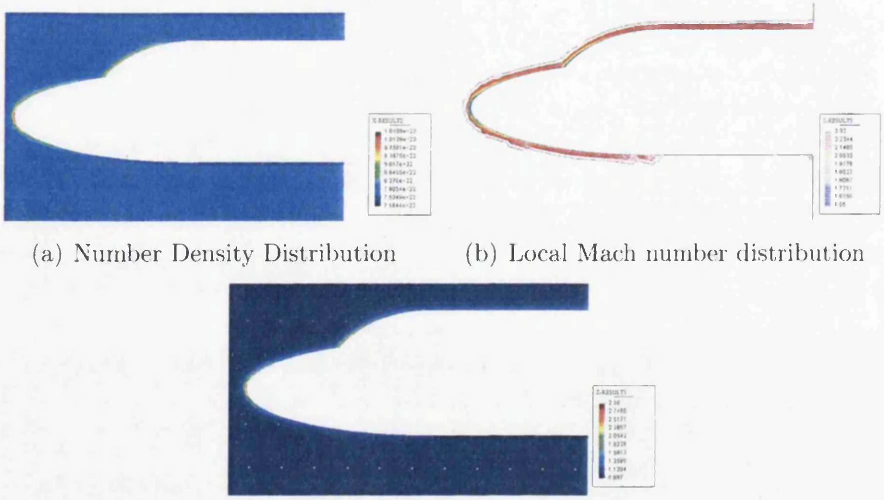
(b) Local Mach number distribution

(c) pressure coefficient

Figure 4.45: Full Boltzmann Case 2 ($M_\infty = 1.1$, $AoA = 0°$, $Kn \approx 1.0$)

number. This slip was also observed in the collisionless cases (section 4.2.3), but not in the continuum cases using the BGK–solver (section 4.3). This does demonstrate one of the advantages of a molecular approach in analysis over a range of Knudsen number since no modification to the wall boundary condition has been necessary to account for the presence of slip at the wall as would be the case when using a typical continuum solver.

The form of the distribution function and collision term at a point immediately



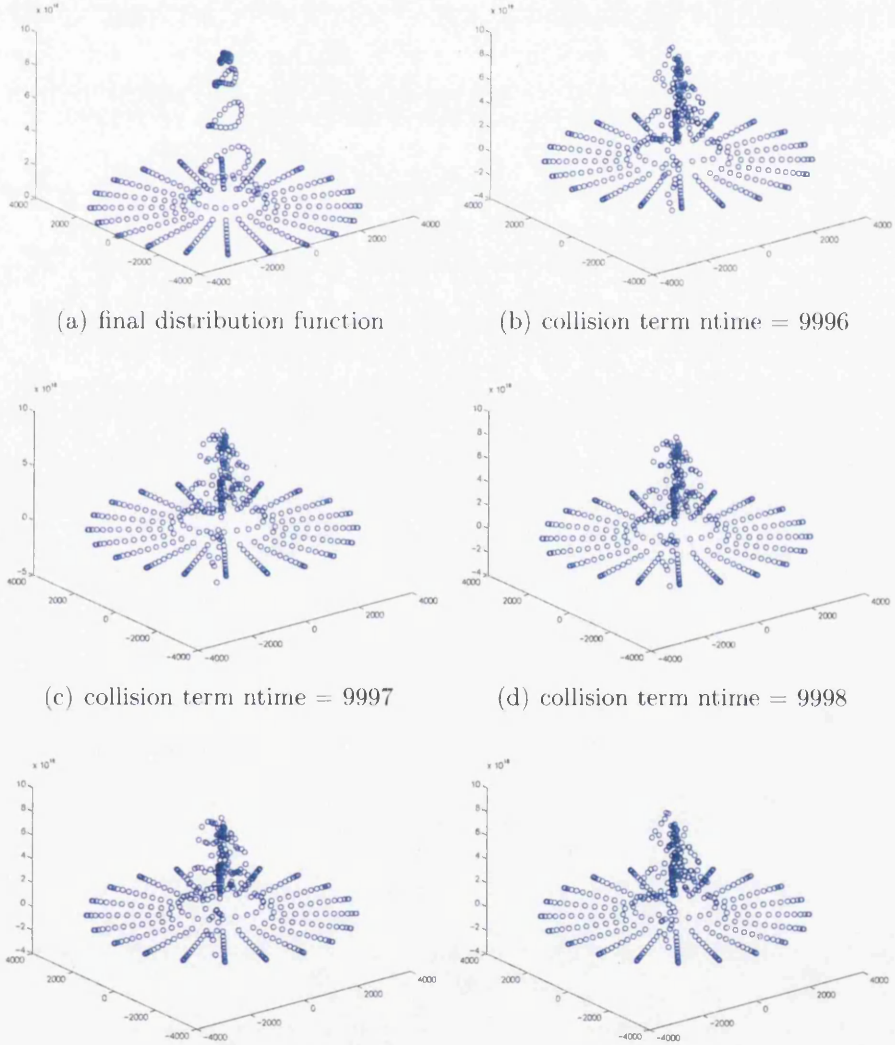
(a) Number Density Distribution

(b) Local Mach number distribution

(c) pressure coefficient

Figure 4.46: Full Boltzmann Case 3 ($M_\infty = 2.3$, $AoA = 0°$, $Kn \approx 1.0$)

ahead of the stagnation point for the $M = 2.3$ case was analysed to gain some understanding of the behaviour of molecules in this region as the solution evolved towards convergence. The form of the final distribution function satisfying steady-state equilibrium and the collision term over the final five timesteps are shown in Figures 4.47(a) to 4.47(f). The interesting point to note here is that even though the distribution function is now in a state of equilibrium, the stochastic nature of the collisions that are maintaining this equilibrium result in a varying collision term (Figures 4.47(b) to 4.47(f)). This does give some further insight into the depth in which this problem is being tackled and understood by using a molecular approach and helps understand why, computationally, it is so demanding.



(a) final distribution function

(b) collision term ntime = 9996

(c) collision term ntime = 9997

(d) collision term ntime = 9998

(e) collision term ntime = 9999

(f) collision term ntime = 10000

Figure 4.47: Velocity distribution function and collision function variation at a point ahead of the nose, $M = 2.3$

Parallelisation Speed Up

The speed–up achieved by parallelisation of the collisionless and BGK–Boltzmann solvers was insignificant and implemented essentially due to memory requirements. The poor speed–up performance in these cases is due to the large volume of data communicated between p–space partitions at each timestep i.e. the entire distribution function defined across v–space.

However, since the full Boltzmann solver performs so many more algebraic operations per timestep than the collisionless and BGK versions, a significant parallelisation speed–up was observed. The graph in Figure 4.48 shows the speed–up for the double ellipse problem running from 4 up to 32 processors. It can be seen that initially a super–linear speed–up is achieved but this level of speed–up deteriorates as the number of processors increases.

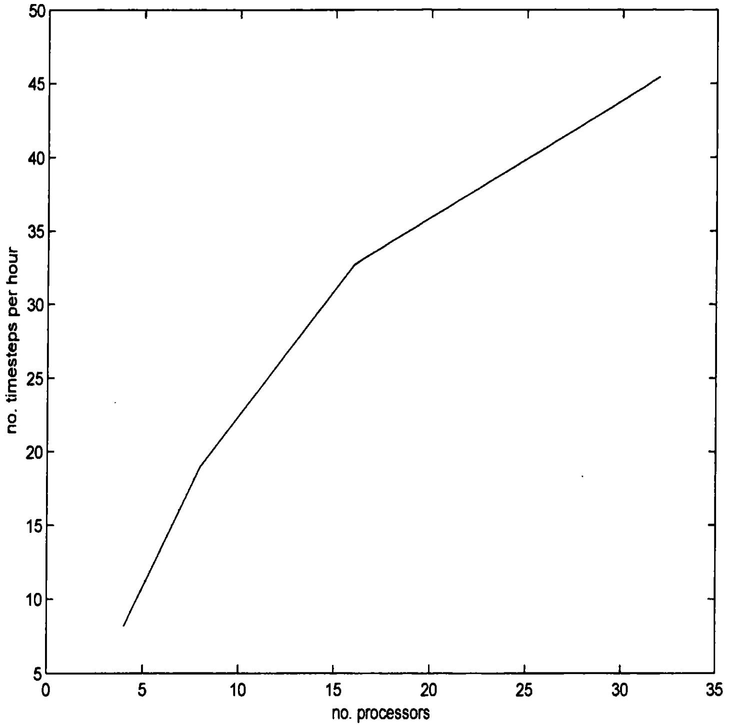


Figure 4.48: Full Boltzmann solver parallelisation speed–up

Parallelisation speed–up performance may be improved by partitioning the v–space domain as well as the p–space domain, but this is beyond the scope of the work presented in this thesis.

4.4.3 Limitations of the Full Boltzmann Solver

Clearly a ten day convergence time on 32 processors to solve a relatively simple 2D problem such as those shown here presents a massive limitation in terms of the scope of analysis using this method in its current form. It has already been mentioned that a parallelisation including decomposition of the v–space domain could provide further significant speedups.

The full Boltzmann solver was also applied to problems with higher Mach numbers and lower Knudsen number (higher density) problems. Unfortunately, these cases resulted in an instability that appeared to originate upstream of the nose at the point of formation of a bow shock. This limited the full Boltzmann analysis to cases with $Kn_\infty \geq 1.0$ and $M < 2.5$. There are a number of possible strategies for investigating the cause of and protection from this instability. Some of these are detailed below:

P–space Refinement The p–space mesh used for these examples was relatively coarse. It might be the case that an adaptive refinement method to capture the shock would be sufficient to stabilise the scheme.

V–space Refinement Other than the simple distribution function and collision term plots shown in the previous section, no significant analysis into the form of the distribution function solution across v–space has taken place. It might be the case that the spectral discretisation of v–space described in section 3.1.2 is not providing an adequate description of the distribution and/or collision term in zones of highly non–equilibrium flow. A v–space refinement might be necessary to overcome this.

Limiting It might be the case that the discontinuous Taylor–Galerkin scheme underlying the whole solver requires further stabilisation in the form of a discontinuous limiter such as described in [12] when strong shocks are present.

Evidently, there is much scope for further research and development but this, unfortunately, is beyond the scope of this project. In the final chapter, the main conclusions of all the work presented here and recommendations for further study will be summarised.

Chapter 5

Concluding remarks

5.1 Conclusions

The work presented in this thesis lays the foundations for the study of the connection between the governing equations of molecular gas dynamics and macroscopic gas flow phenomena in terms of finite element solutions of the Boltzmann equation and its variants. Research in this field is very much still in its early days and simply showing the connection between molecular interaction models and phenomena such as shock waves may be regarded as significant. The significance of analysing such macroscopic flows as shown in this thesis begninning with microscopic behaviour of fundamental particles, rather than the standard continuum equations, cannot be stressed enough.

The first stage of the research was a comparison of scalar convection equation solvers. Particular emphasis was placed upon comparing discontinuous and continuous finite elements and structured/unstructured meshes. The results and discussion from this study are detailed in section 4.1. It is clear that the main benefits of a discontinuous approach are that it minimises the undershoots and overshoots at a discontinuity. Unfortunately, moving from structured to unstructured meshes, which is necessary for computations around complex geometries, offsets some of the benefits gained from the discontinuous approach. Nevertheless, the advantages of the discontinuous approach, combined with the improved parallelisability that it provides, made it an attractive scheme for tackling the Boltzmann equation.

The solution methodology presented has been based on the discontinuous Taylor–Galerkin algorithm and has been developed for three of the fundamental equations of molecular gas dynamics; the collisionless Boltzmann equation (3.11), the Boltzmann–BGK equation (2.31) and the full Boltzmann equation (2.29). The basis for each of the solution approaches is a scalar convection equation solver applied at indivudual nodes or sampling points in a v–space discretisation. Two discretisation approaches for the v–space domain have been suggested and compared. These approaches are laid out in section 3.1.2 and the conclusion that a spectral approach is preferable to using standard linear finite elements is drawn in section 4.2.1.

In section 4.2, the collisionless Boltzmann equation solver was applied to highly rarefied (high Knudsen number) gas flows in which the effect of collisions between molecules can be deemed negligible. The examples examined included a shock tube–type free expansion problem, flow over a vertical plate and hypersonic flow over a generic re–entry vehicle nose geometry. The benchmark tests showed remarkable agreement with both analytical solutions and results from alternative numerical approaches resulting in a trustworthy rarefied gas flow solver that could be applied to the non–standard highly rarefied, hypersonic cases.

In section 4.3, the Boltzmann–BGK equation solver was applied to lower Knudsen number, low Mach number problems. These problems included flow over a vertical plate and subsonic, transonic and supersonic flow over an aerofoil at a range of Knudsen and Reynolds' numbers. The boundary condition formulation was also examined in this section comparing the two types of molecule–wall interaction; specular reflection and diffuse reflection. A wall boundary condition dominated by diffuse reflection was deemed to be the most appropriate when modelling such flows. It was also shown in this section, that the Boltzmann–BGK solver fails when the flow contains shocks stronger than approximately Mach 2.

In section 4.4, two different molecular interaction models were first considered. This analysis showed that a simple hard sphere interaction model fails in the necessary requirement of the Boltzmann equation collision term to restore a non–equilibrium distribution function to equilibrium. The more complex variable soft sphere model was, therefore, adopted using a Monte Carlo sampling methodology for selection of the pre–collision velocity pairings. The full Boltzmann equation solver was then applied to a selection of transition regime flows, that is flows with a Knudsen number close to unity, at a range of Mach numbers over the same generic re–entry vehicle nose geometry as in section 4.2. It became clear that computational demand places a massive limitation on the scope of study when using the full Boltzmann solver to analyse transition regime flows. This limitation made it impossible to overcome stability problems experienced for lower Knudsen number problems and higher Mach number problems. In section 1.3, in which the question of why a combined kinetic theory and CFD approach was analysed, the different Knudsen regimes experienced by a spacecraft on hypersonic re–entry was described. It appears to be the case that it is the transition regime between fully free molecular flow, governed by the collisionless Boltmann equation, and fully continuous flow, governed by the continuum Navier–Stokes equations, that remains to be the real challenge for CFD.

Clearly, one of the significant disadvantages, and indeed, limiting factors, of modelling a gas at the molecular level is the massive increase in computational demand over traditional macroscopic continuum approaches. The necessity of phase–space discretisation leads to a memory increase of at least an order of magnitude and similar increases in convergence times. When using typical computing resources available today, this essentially limits study to fairly modest two–dimensional problems. Nevertheless, the opportunities that a scheme such as that described in this thesis may unveil are many and varied and will continue to diversify as computational resources grow in the future. Any fluid flow problem that exhibits significantly non–equilibrium behaviour should be analysed beginning with the fundamental equations of molecular gas dynamics. Such applications include combustion problems, nano–scale problems, rarefied gas flows and hypersonics. At some point in the future, Moore's law suggests that computational resources will be such that extensions into three–dimensions will be made possible.

5.1.1 Extensions

Since the work presented here is foundational, there are numerous possibilities for extensions to the research. These extensions highlight both the limitations of what has been achieved so far, but also the possibilities that a finite element analysis of the Boltzmann equation holds.

In this thesis, when 'air' is modelled, it is not done so in its true multi–component form. Instead, air is considered as a single–species gas, with identical monatomic molecules satisfying a variable soft sphere interaction model when collisions are considered. When a Maxwellian distribution function is constructed, it is done so based upon the gas constant for air. This is obviously a massive simplification of reality (yet remarkably accurate results have been achieved!) and, therefore, a modelling of air using separate distribution functions for its different component gases is an obvious next stage in development. Also, if we are to model the constituent particles as the polyatomic molecules of reality, the distribution function becomes a function of internal vibrational and rotational energy modes (see [5]) as well as time and phase–space position.

In the present analysis of wall boundaries, detailed in section 3.5.3, the wall temperature is assumed to be fixed at a preset user–defined temperature. Of course, in reality, the wall temperature of a container/aerofoil/re–entry vehicle will vary according to the gas in contact with it by means of heat flux to and from the wall. This process is ignored here and presumably is an important factor in an accurate simulation of problems, particularly those involving very high temperatures such as the hypersonic applications. An adaption of the boundary condition to allow the wall temperature to vary until it falls into equilibrium with the gas in contact with it is, therefore, another obvious extension. This is already a standard boundary condition technique in continuum solvers.

Whilst on the subject of gas at high temperatures, it was mentioned in the introduction, that one of the features of hypersonic flow is high temperature gas effects including chemical reactions and ionisation. Clearly, the simple description of a gas in the analysis here does not allow any such effects to be considered. An extension into a multi–component gas model, with internal energy modes accounted for, would allow such effects to be considered.

All of the examples in this thesis have indicated that one of the serious drawbacks of this family of methods is convergence time. Although the methods presented here will never compete with standard continuum solvers for flows in which the continuum approach is valid, it still would seem sensible to investigate the application of convergence accelerating algorithms, such as multigrid ([44],[96]), to the Boltzmann solvers.

The source term in the BGK–Boltzmann approximation requires the calculation of the molecular collision frequency. The algorithm to perform this calculation is detailed in section 3.4.1. This is a computationally expensive procedure since it involves integrals over v–space. Analytical formulations for the collision frequency of a gas do exist (see [47]) that are strictly only applicable to gases in thermodynamic equilibrium. It might be beneficial, however, to investigate the effect of applying a collision frequency formula, which will be a function of the macroscopic properties of the gas, instead of computing it explicitly i.e. make the assumption that the difference between the collision frequency for an equilibrium and non–equilibrium gas given the same macroscopic conditions is negligible. If this turns out to be true, significant acceleration of the BGK–solver could be achieved.

A further source of convergence acceleration could be achieved if a method for parallelising the v–space were to be investigated as well as the physical domain decomposition detailed in section 3.7. The physical space domain decomposition is not as efficient as one might expect in comparison with a typical Navier–Stokes solver due to the massive volume of data transfer at processor domain boundaries. A v–space parallelisation might help overcome this problem.

Finally, it was mentioned in the introductory chapter that it is slightly misleading to characterise an entire flow with a single Knudsen number, but rather a local Knudsen number varying throughout the flowfield is more appropriate. Also, it is certainly the case that non–equilibrium features resulting from strong shocks will always be localised. This suggests that the most effective application of a non–equilibrium methodology such as those described here is only applied in the truly non–equilibrium regions of a flow field and a more efficient continuum solver everywhere else. This would necessitate the ‘patching’ of the Boltzmann solver to a Navier–Stokes solver. In the discontinuous Galerkin context, this might be achieved simply by having a tranformation from the distribution function inter–element flux to the Navier–Stokes fluxes and vice versa, possibly resulting in a very powerful and multi–purpose algorithm.

5.1.2 Final Remarks

The work in this thesis has demonstrated an alternative approach for simulation of gas flows to the mainstream continuum CFD solvers. This approach begins with a more fundamental model of the physics of gas flows and hence, at least in principle, is more generally applicable. The drawbacks of such an approach in its current form and with the current day computing power are evident, but the possibilities for development and progress in a field of computer modelling in its infancy, that might best be described as ‘computational molecular gas dynamics’, are many and fascinating.

Part I

Appendices

Appendix A

Proof of the Wall Boundary Condition

$$\int_{\Gamma_r}\int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}$$

$$= \int_{\Gamma_r}\int_{-\infty}^{+\infty} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c}$$

by applying the definition of the normal flux, $F_{n,\boldsymbol{c}}$, at a wall

$$= \int_{\Gamma_r}\left(\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \boldsymbol{c}.\boldsymbol{n}(1-\alpha)f(\boldsymbol{c}-2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}),\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \cdots\right.$$
$$\left.\int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \alpha\boldsymbol{c}.\boldsymbol{n}\eta(\boldsymbol{r},t)M_w(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\right)\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}$$

by splitting the integral into components that are towards the wall and away from the wall and substituting from equation (3.45) for the reflected flux

$$= \int_{\Gamma_r}\left(\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} - \int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \cdots\right.$$
$$\left.\alpha\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \alpha\int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \boldsymbol{c}.\boldsymbol{n}\eta(\boldsymbol{r},t)M_w(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\right)\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}$$

noting that $\int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c}-2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{c}),\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} = -\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c}$

$$= \int_{\Gamma_r}\left(\alpha\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} + \alpha\int_{\boldsymbol{c}.\boldsymbol{n}\leq 0} \boldsymbol{c}.\boldsymbol{n}\eta(\boldsymbol{r},t)M_w(\boldsymbol{c})\,\mathrm{d}\boldsymbol{c}\right)\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}$$

by cancelling like terms.

If $\eta(\boldsymbol{r},t)$ is defined as in equation (3.49), we then have

$$\int_{\Gamma_r}\int_{-\infty}^{+\infty} F_{n,\boldsymbol{c}}\,\mathrm{d}\boldsymbol{c}\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}$$
$$= \int_{\Gamma_r}\left(\alpha\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c} - \alpha\int_{\boldsymbol{c}.\boldsymbol{n}>0} \boldsymbol{c}.\boldsymbol{n}f(\boldsymbol{c},\boldsymbol{r},t)\,\mathrm{d}\boldsymbol{c}\right)\,\mathrm{d}\boldsymbol{\Gamma}_{\boldsymbol{r}}$$
$$= 0$$

Appendix B

Divergence (Gauss') Theorem

Note that the divergence theorem is a special case of the more general Stokes' theorem which generalizes the fundamental theorem of calculus.

$$\int\int\int_V (\nabla.\boldsymbol{F})\,\mathrm{d}V = \int\int_S \boldsymbol{F}.\boldsymbol{n}\,\mathrm{d}S$$

where $\boldsymbol{n}$ denotes the outward unit normal at differential surface element $\mathrm{d}S$ and surface S encloses volume V.

Appendix C

MATLAB code

```
function FTCS
      clear
      clc
      % set value of timestep
      dt=0.1;
      %set the number of timesteps
      nstep=10
      %set the value of the flow velocity
      u=5
      %set the lefthand and righthand boundaries of the domain
      a=0;
      b=500;
      %set the number of spatial discretisations
      n=1000;
      %set the width of a spatial step
      h=(b-a)/n;
      %construct the spatial coordinate (x-axis) vector
      for i=1:n
      x(i)=i;
      end
      %construct the initial value of the function f across the domain (in this case a pulse)
      for i=1:20
       f1(i)=0;
      end
      for i=21:40
       f1(i)=(i-20);
      end
      for i=41:80
       f1(i)=20;
      end
      for i=81:100
       f1(i)=(100-i);
      end
      for i=101:n
       f1(i)=0;
      end
      %plot the initial f-function
      plot(x,f1)
      axis([0 200 0 30])
      %create an (nstep x n) matrix of f-functions, each row is an f-function at
      %a given timestep,first row set to initial value of f
      for i=1:n
       f(1,i)=f1(i);
      end
      %loop over the number of timesteps to determine the time-evolution of f
      for t=2:nstep
       %calculate the FTFS approximation at f(1)
       f(t,1)=f((t-1),1)-(dt*u)*((f((t-1),2)-f((t-1),1))/h);
       %calculate the FTCS approximation at f(2)->f(n-1)
       for i=2:(n-1)
       f(t,i)=f((t-1),i)-(dt*u)*((f((t-1),(i+1))-f((t-1),(i-1)))/(2*h));
       end
       %calculate the FTBS approximation at f(n)
       f(t,n)=f((t-1),n)-(dt*u)*((f((t-1),n)-f((t-1),(n-1)))/h);
      end
      %bring up new figure to plot the time evolutions of f
      figure
      %plot each time evolution of f (contained in matrix f)
      plot(x,f)
      axis([0 200 0 30])
      %=====================================================================
      %repeat procedure for a new initial function
      %=====================================================================
      %for new g-function set the boundaries of the domain
```

```
a=0
b=10*pi
%set the width of a spatial step
h=(b-a)/n;
%construct the initial value of the g-function across the domain (in this
%case a sine wave)
for i=1:n
 g1(i)=sin(i*h)
end
%plot intial value of the function
figure
plot(x,g1)
axis([0 n -1 1])
%create an (nstep x n) matrix of g-functions, each row is a g-function at
%a given timestep,first row set to initial value of g
for i=1:n
 g(1,i)=g1(i);
end
%loop over the number of timesteps to determine the time-evolution of g
for t=2:nstep
 %calculate the FTFS approximation at g(1)
 g(t,1)=g((t-1),1)-(dt*u)*((g((t-1),2)-g((t-1),1))/h);
 %calculate the FTCS approximation at g(2)->g(n-1)
 for i=2:(n-1)
 g(t,i)=g((t-1),i)-(dt*u)*((g((t-1),(i+1))-g((t-1),(i-1)))/(2*h));
 end
%calculate the FTBS approximation at g(n)
 g(t,n)=g((t-1),n)-(dt*u)*((g((t-1),n)-g((t-1),(n-1)))/h);
end
%bring up new figure to plot the time evolutions of g
figure
%plot each time evolution of g (contained in matrix g)
plot(x,g)
axis([0 n -2 2])
```

```
function upwind
clear all; clc;
%set the initial value of the problem
% iopt=1 SQUARE PULSE
% iopt=2 smoothed wave
% iopt=3 PULSE2
% iopt=4 SINE WAVE
iopt=3;
%choose whether to animate the output or freeze frame the output
% jopt=1 ANIMATE
% jopt=2 FREEZE FRAME THE PULSE 2 FUNCTION
% jopt=3 FREEZE FRAME THE SINE WAVE FUNCTION
jopt=2;
%set the calue of the Courant number
=1;
%set the value of the flow velocity
u=5;
%set the value of time_end
time_end = 1.7;
%set the number of timesteps if freeze framing
nstep=2000;
%set the number of nodal points
n=300;
%number of elements
ne=n-1;
%construct initial function dependent on iopt selection
switch iopt
 case 1
 %construct a vector of the element lengths of random length between
 %0.5 and 1.25
 x(1)=0.0;
 for ie=1:ne
 h(ie)=(0.5+rand*0.75);
 %construct the spatial coordinate (x-axis) vector
 x(ie+1)=x(ie)+h(ie);
 end
 %set the limits of the domain
 a=x(1);
 b=x(n);
 %construct the initial value of the function f across the domain (in this
 %case a pulse)
 for i=1:n
 if x(i)<=10
 f(i)=0;
 elseif x(i)<=20
 f(i)=1;
 else
 f(i)=0;
 end
 end
 case 2
 %construct a vector of the element lengths of random length between
 %0.5 and 1.25
 x(1)=0.0;
 for ie=1:ne
 h(ie)=(0.5+rand*0.75);
 %construct the spatial coordinate (x-axis) vector
 x(ie+1)=x(ie)+h(ie);
 end
 %set the limits of the domain
 a=x(1);
 b=x(n);
 % construct the initial value of the f-function across the domain
 % (in this case a sine pulse)
 for i=1:n
 if x(i)<=2*pi
 f(i)=0;
 elseif x(i)<=2*1.25*pi
 f(i)=(sin(x(i)/2+(pi)/2)+1);
 elseif x(i)<=2*1.75*pi
 f(i)=0.4*sin((x(i)/2-pi));
 elseif x(i)<=2*2*pi
 f(i)=(sin((x(i)/2-(pi)/2))+1);
 else
 f(i)=0;
end
end
case 3
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
for ie=1:ne
h(ie)=(0.5+rand*0.75);

x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
```

```
b=x(n);
%construct the PULSE 2 initial condition
for i=1:20
f(i)=0;
end
for i=21:40
f(i)=(i-20);
end
for i=41:80
f(i)=20;
end
for i=81:100
f(i)=(100-i);
end
for i=101:n
f(i)=0;
end
case 4
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
ne
for ie=1:ne
h(ie)=(0.5+rand*0.75);
%construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%construct the initial value of the g-function across the domain (in this
%case a sine wave)
for i=1:n
f(i)=sin(0.05*x(i));
end
end
end
%choose whether to animate the output or static freeze frame the output
switch jopt
case 1
plot(x,f)
% ANIMATE THE OUTPUT:
%set the time variable to 0
time=0;
%set the icount variable to 0
icount=0;
%plot the initial distribution of f
set(gcf,'doublebuffer','on')
p=plot(x,f);
axis([a b -0.5 1.5])
%label the x-axis
xlabel('distance, x');
set(p,'EraseMode','xor');
str=['time= ' num2str(time)];
title(str, 'FontSize', 10)
drawnow;
pause
%loop over time steps
while time < time$_end$
%calculate value of timestep based on Courant number
dt=C*h/u;
time=time+dt;
%flux vector splitting to allow for LR advection and RL
%advection
upos=0.5*(u+abs(u));
uneg=0.5*(u-abs(u));
%upwind differencing method
for k=2:n-1
fnew(k)=f(k)-(dt/h)*(upos*(f(k)-f(k-1))+uneg*(f(k)-f(k+1)));
end
%SET BOUNDARY CONDITIONS
%left hand boundary:  unchanged
fnew(1)=f(1);
%right hand boundary:  unchanged
fnew(n)=f(n);
for k=1:n
f(k)=fnew(k);
end
pause(0.01);
set(p,'Ydata',f);
str=['time= ', num2str(time)];
title(str, 'FontSize',10)
drawnow;
end
case 2
dt=C*min(h)/u;
%flux vector splitting to allow for LR advection and RL
%advection
```

```
upos=0.5*(u+abs(u));
uneg=0.5*(u-abs(u));
plot(x,f)
axis([a b 0 30])
%create an (nstep x n) matrix of f-functions, each row is an
%f-function ata given timestep, first row set to initial value of f
for i=1:n
fplot(1,i)=f(i);
end
%loop over the number of timesteps to determine the time-evolution
%of f
for t=2:nstep
%fix the left and right boundary values
fplot(t,1)=fplot(1,1);
fplot(t,n)=fplot(1,n);
%calculate the upwind approximation for f(2)-> f(n-1)
for i=2:(n-1)
fplot(t,i)=fplot(t-1,i)-(dt/h(i))*(upos*(fplot(t-1,i)-fplot(t-1,i-1))+uneg*(fplot(t-1,i)-fplot(t-1,i+1)));
end
end
% new figure to plot the time evolution
figure
plot(x,fplot)
axis([a b 0 30])
%plot the evolution every 20 timesteps
for t10=1:10
t=t10*20
for i=1:n
newfplot(t10,i)=fplot(t,i);
end
end
figure
plot(x,newfplot)
axis([a b 0 30])
case 3
dt=C*min(h)/u;
%flux vector splitting to allow for LR advection and RL
%advection
upos=0.5*(u+abs(u));
uneg=0.5*(u-abs(u));
plot(x,f)
axis([0 b -2 2])
%create an (nstep x n) matrix of f-functions, each row is an
%f-function ata given timestep, first row set to initial value of f
for i=1:n
fplot(1,i)=f(i);
end
%loop over the number of timesteps to determine the time-evolution
%of f
for t=2:nstep
%fix the left and right boundary values
fplot(t,1)=fplot((t-1),1);
fplot(t,n)=fplot((t-1),n)-(dt*u)*((fplot((t-1),n)-fplot((t-1),(n-1)))/h(ne));
%calculate the upwind approximation for f(2)-> f(n-1)
for i=2:(n-1)
fplot(t,i)=fplot(t-1,i)-(dt/h(i))*(upos*(fplot(t-1,i)-fplot(t-1,i-1))+uneg*(fplot(t-1,i)-fplot(t-1,i+1)));
end
end
% new figure to plot the time evolution
figure
plot(x,fplot)
axis([0 b -2 2])
%plot the evolution every 10 timesteps
for t10=1:10
t=t10*20;
for i=1:n
newfplot(t10,i)=fplot(t,i);
end
end
figure
plot(x,newfplot)
axis([0 b -2 2])
end
end  %construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%construct the PULSE 2 initial condition
for i=1:20
f(i)=0;
end
for i=21:40
f(i)=(i-20);
end
for i=41:80
f(i)=20;
```

```
end
for i=81:100
f(i)=(100-i);
end
for i=101:n
f(i)=0;
end
case 4
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
ne
for ie=1:ne
h(ie)=(0.5+rand*0.75);
%construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%construct the initial value of the g-function across the domain (in this
%case a sine wave)
for i=1:n
f(i)=sin(0.05*x(i));
end
end
end
%choose whether to animate the output or static freeze frame the output
switch jopt
case 1
plot(x,f)
% ANIMATE THE OUTPUT:
%set the time variable to 0
time=0;
%set the icount variable to 0
icount=0;
%plot the initial distribution of f
set(gcf,'doublebuffer','on')
p=plot(x,f);
axis([a b -0.5 1.5])
%label the x-axis
xlabel('distance, x');
set(p,'EraseMode','xor');
str=['time= ' num2str(time)];
title(str, 'FontSize', 10)
drawnow;
pause
%loop over time steps
while time < time$_{e}nd$
%calculate value of timestep based on Courant number
dt=C*h/u;
time=time+dt;
%flux vector splitting to allow for LR advection and RL
%advection
upos=0.5*(u+abs(u));
uneg=0.5*(u-abs(u));
%upwind differencing method
for k=2:n-1
fnew(k)=f(k)-(dt/h)*(upos*(f(k)-f(k-1))+uneg*(f(k)-f(k+1)));
end
%SET BOUNDARY CONDITIONS
%left hand boundary:  unchanged
fnew(1)=f(1);
%right hand boundary:  unchanged
fnew(n)=f(n);
for k=1:n
f(k)=fnew(k);
end
pause(0.01);
set(p,'Ydata',f);
str=['time= ', num2str(time)];
title(str, 'FontSize',10)
drawnow;
end
case 2
dt=C*min(h)/u;
%flux vector splitting to allow for LR advection and RL
%advection
upos=0.5*(u+abs(u));
uneg=0.5*(u-abs(u));
plot(x,f)
axis([a b 0 30])
%create an (nstep x n) matrix of f-functions, each row is an
%f-function ata given timestep, first row set to initial value of f
for i=1:n
fplot(1,i)=f(i);
end
%loop over the number of timesteps to determine the time-evolution
```

```
%of f
for t=2:nstep
%fix the left and right boundary values
fplot(t,1)=fplot(1,1);
fplot(t,n)=fplot(1,n);
%calculate the upwind approximation for f(2)-> f(n-1)
for i=2:(n-1)
fplot(t,i)=fplot(t-1,i)-(dt/h(i))*(upos*(fplot(t-1,i)-fplot(t-1,i-1))+uneg*(fplot(t-1,i)-fplot(t-1,i+1)));
end
end
% new figure to plot the time evolution
figure
plot(x,fplot)
axis([a b 0 30])
%plot the evolution every 20 timesteps
for t10=1:10
t=t10*20
for i=1:n
newfplot(t10,i)=fplot(t,i);
end
end
figure
plot(x,newfplot)
axis([a b 0 30])
case 3
dt=C*min(h)/u;
%flux vector splitting to allow for LR advection and RL
%advection
upos=0.5*(u+abs(u));
uneg=0.5*(u-abs(u));
plot(x,f)
axis([0 b -2 2])
%create an (nstep x n) matrix of f-functions, each row is an
%f-function ata given timestep, first row set to initial value of f
for i=1:n
fplot(1,i)=f(i);
end
%loop over the number of timesteps to determine the time-evolution
%of f
for t=2:nstep
%fix the left and right boundary values
fplot(t,1)=fplot((t-1),1);
fplot(t,n)=fplot((t-1),n)-(dt*u)*((fplot((t-1),n)-fplot((t-1),(n-1)))/h(ne));
%calculate the upwind approximation for f(2)-> f(n-1)
for i=2:(n-1)
fplot(t,i)=fplot(t-1,i)-(dt/h(i))*(upos*(fplot(t-1,i)-fplot(t-1,i-1))+uneg*(fplot(t-1,i)-fplot(t-1,i+1)));
end
end
% new figure to plot the time evolution
figure
plot(x,fplot)
axis([0 b -2 2])
%plot the evolution every 10 timesteps
for t10=1:10
t=t10*20;
for i=1:n
newfplot(t10,i)=fplot(t,i);
end
end
figure
plot(x,newfplot)
axis([0 b -2 2])
end
end
```

```
function LAX
clear clc
%set value of timestep
dt=0.1;
%set the number of timesteps
nstep=200
%set the number of discretisation points
n=1000;
%number of elements
ne=n-1;
%set the value of the flow velocity
u=5
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
for ie=1:ne
h(ie)=(0.5+rand*0.75);
%construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%the Courant number
C=u*dt/min(h)
%construct the initial value of the function f across the domain (in this
%case a pulse)
for i=1:20
f1(i)=0;
end
for i=21:40
f1(i)=(i-20);
end
for i=41:80
f1(i)=20;
end
for i=81:100
f1(i)=(100-i);
end
for i=101:n
f1(i)=0;
end
%plot the initial f-function
plot(x,f1)
axis([a b 0 30])
%create an (nstep x n) matrix of f-functions, each row is an f-function at
%a given timestep,first row set to initial value of f
for i=1:n
f(1,i)=f1(i);
end
%loop over the number of timesteps to determine the time-evolution of f
for t=2:nstep
%calculate the FTFS approximation at f(1)
f(t,1)=f((t-1),1)-(dt*u)*((f((t-1),2)-f((t-1),1))/h(1));
%calculate the LAX 'staggered' FTCS approximation at f(2)->f(n-1)
for i=2:(n-1)
f(t,i)=0.5*(f((t-1),(i+1))+f((t-1),(i-1)))-(dt*u)*((f((t-1),(i+1))-f((t-1),(i-1)))/(h(i)+h(i-1)));
end
%calculate the FTBS approximation at f(n)
f(t,n)=f((t-1),n)-(dt*u)*((f((t-1),n)-f((t-1),(n-1)))/h(n-1));
end
%bring up new figure to plot the time evolutions of f
figure
%plot each time evolution of f (contained in matrix f)
plot(x,f)
axis([a b 0 30])
%plot the evolution every 20 timesteps
for t10=1:10
t=t10*20
for i=1:n
newfplot(t10,i)=f(t,i);
end
end
figure
plot(x,newfplot)
axis([a b 0 30])
% new figure to plot the time evolution
% figure
% plot(x,f)
% axis([0 200 0 30])
%plot the evolution every 10 timesteps
% for t10=1:10
% t=t10*10;
% for i=1:n
% newf(t10,i)=f(t,i);
% end
% end
% figure
% plot(x,newf)
% axis([0 200 0 30])
%=====================================================================
%repeat procedure for a new initial function
%=====================================================================
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
for ie=1:ne
h(ie)=(0.5+rand*0.75);
%construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%construct the initial value of the g-function across the domain (in this
%case a sine wave)
for i=1:n
g1(i)=sin(0.05*x(i));
end
%plot intial value of the function
figure
plot(x,g1)
axis([a b -2 2])
%create an (nstep x n) matrix of g-functions, each row is a g-function at
%a given timestep,first row set to initial value of g
for i=1:n
g(1,i)=g1(i);
end
%loop over the number of timesteps to determine the time-evolution of g
for t=2:nstep
%calculate the FTFS approximation at g(1)
%g(t,1)=g((t-1),1)-(dt*u)*((g((t-1),2)-g((t-1),1))/h(1));
g(t,1)=0;
%calculate the LAX 'staggered' FTCS approximation at g(2)->g(n-1)
for i=2:(n-1)
g(t,i)=0.5*(g((t-1),(i+1))+g((t-1),(i-1)))-(dt*u)*((g((t-1),(i+1))-g((t-1),(i-1)))/(2*h(i)));
end
%calculate the FTBS approximation at g(n)
g(t,n)=g((t-1),n)-(dt*u)*((g((t-1),n)-g((t-1),(n-1)))/h(n-1));
end
%bring up new figure to plot the time evolutions of g
figure
%plot each time evolution of g (contained in matrix g)
plot(x,g)
axis([a b -2 2])
%plot the evolution every 20 timesteps
for t10=1:10
t=t10*20
for i=1:n
newgplot(t10,i)=g(t,i);
end
end
figure
plot(x,newgplot)
axis([a b -2 2])
```

```
function LAXWENDROFF
clear all
clc
% iopt=1 PULSE2
% iopt=2 SINE WAVE
iopt=2;
%set the timestep size
dt=0.01;
%set the value of the flow velocity
u=5;
%set the number of timesteps
nstep=200;
%set the number of nodal points
n=400;
%calculate the number of elements
ne=n-1;
%construct initial function dependent on iopt selection
switch iopt
case 1
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
for ie=1:ne
h(ie)=(0.5+rand*0.75);
%construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%the Courant number
Csafe=u*dt/min(h)
%construct the initial value of the function f across the domain (in this
%case a pulse)
for i=1:20
f1(i)=0;
end
for i=21:40
f1(i)=(i-20);
end
for i=41:80
f1(i)=20;
end
for i=81:100
f1(i)=(100-i);
end
for i=101:n
f1(i)=0;
end
%plot the initial f-function
plot(x,f1)
axis([a (b/1.5) 0 30])
case 2
%construct a vector of the element lengths of random length between
%0.5 and 1.25
x(1)=0.0;
for ie=1:ne
h(ie)=(0.5+rand*0.75);
%construct the spatial coordinate (x-axis) vector
x(ie+1)=x(ie)+h(ie);
end
%set the limits of the domain
a=x(1);
b=x(n);
%the Courant number
Csafe=u*dt/min(h)
%construct the initial value of the g-function across the domain (in this
%case a sine wave)
for i=1:n
f1(i)=sin(0.05*x(i));
end
%plot intial value of the function
figure
plot(x,f1)
axis([a (b-50) -2 2])
end
%create an (nstep x n) matrix of f-functions, each row is an f-function at
%a given timestep,first row set to initial value of f
for i=1:n
f(1,i)=f1(i);
end
%loop over time steps
%LAXWENDROFF
for itime=1:(nstep-1)
%set the upwind method weightings
w1=1;
w2=1;
%first LAXWENDROFF step
```

```
for ie=1:ne
ile=ie;iri=ie+1;
Fle=u*f(itime,ile);
Fri=u*f(itime,iri);
fh(ie)=(1/(w1+w2))*(w1*f(itime,ile)+w2*f(itime,iri))-(0.5*dt/h(ie))*(Fri-Fle);
end
%second LAXWENDROFF step
for k=2:n-1
iri=k;ile=k-1;
Fle=u*fh(ile);
Fri=u*fh(iri);
fnew(k)=f(itime,k)-(dt/0.5*(h(k)+h(k-1)))*(Fri-Fle);
end
%SET BOUNDARY CONDITIONS
%left hand boundary:
fnew(1)=0.0;
%right hand boundary:  unchanged
fnew(n)=0.0;
%update the solution
for k=1:n
f(itime+1,k)=fnew(k);
end
end
figure
plot(x,f)
axis([a (b-50) -2 2])
%plot the evolution every x timesteps
for t10=1:10
t=t10*20;
for i=1:n
newfplot(t10,i)=f(t,i);
end
end
figure
plot(x,newfplot)
axis([a (b-50) -2 2])
end
end
```

Appendix D

Program Flow Chart for BGK / Full Boltzmann Solver

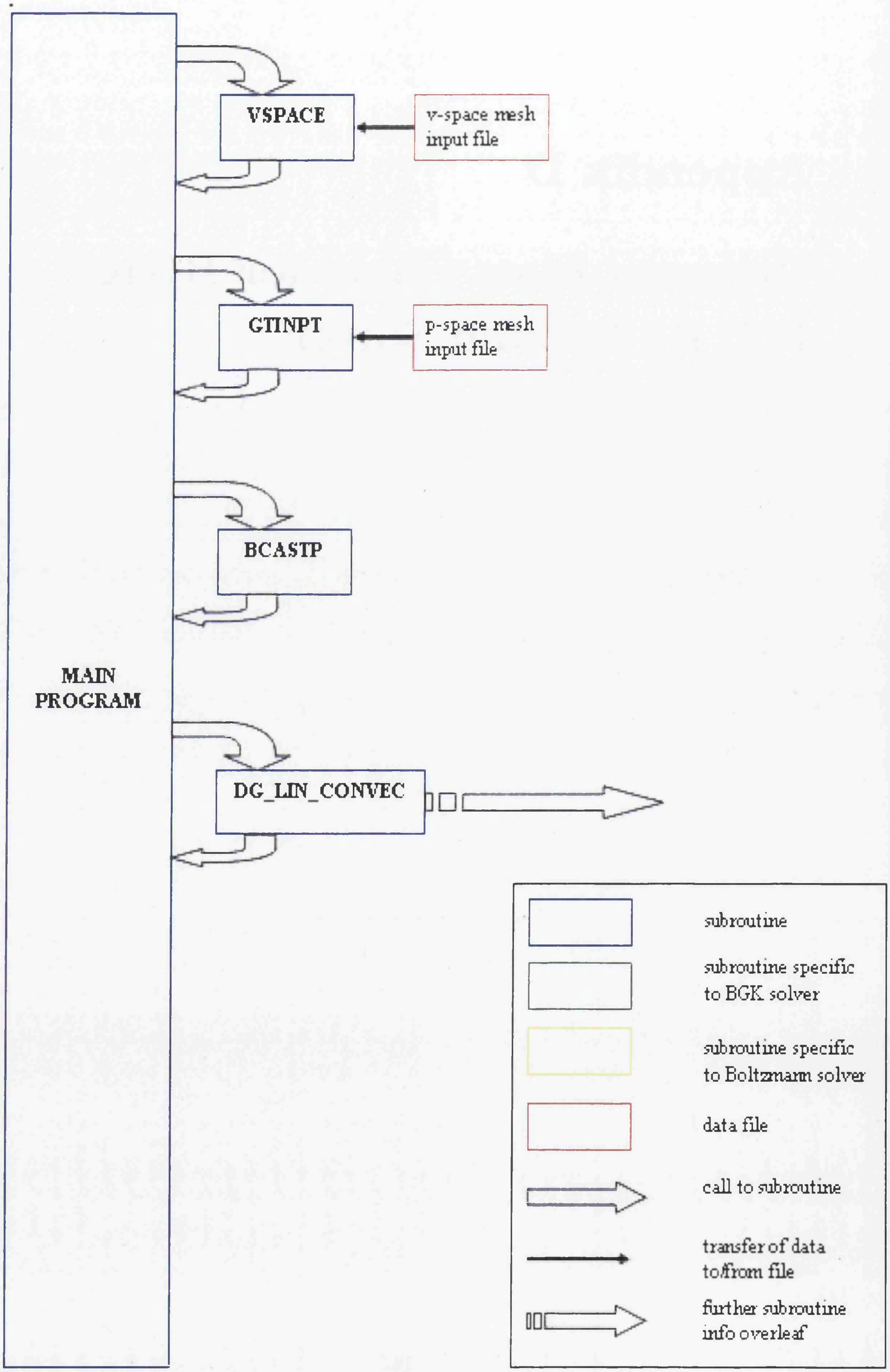

VSPACE
v-space mesh input file
GTINPT
p-space mesh input file
BCASTP
MAIN PROGRAM
DG_LIN_CONVEC
subroutine
subroutine specific to BGK solver
subroutine specific to Boltzmann solver
data file
call to subroutine
transfer of data to/from file
further subroutine info overleaf

Figure D.1: Program Flow Chart (part 1)

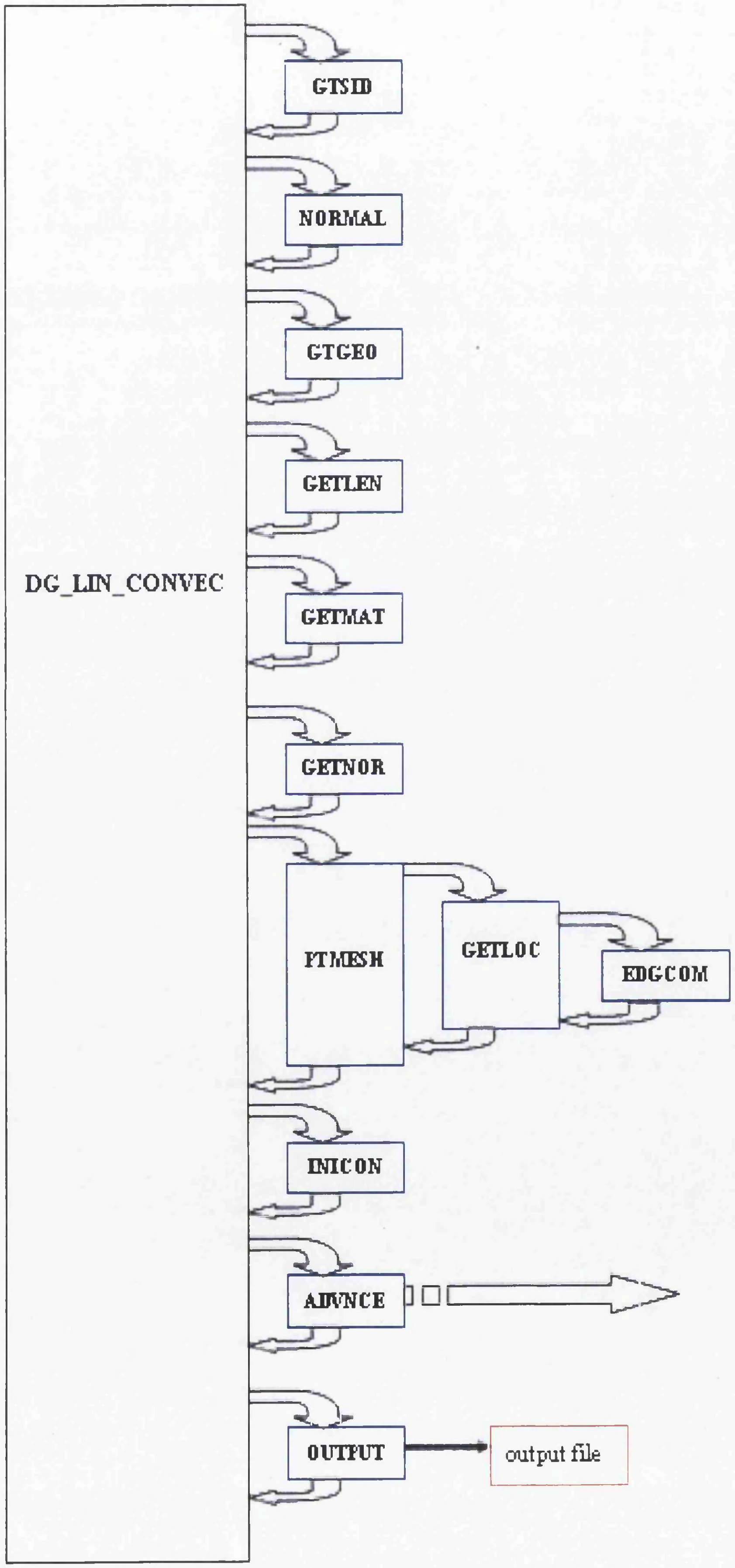


Figure D.2: Program Flow Chart (part 2)

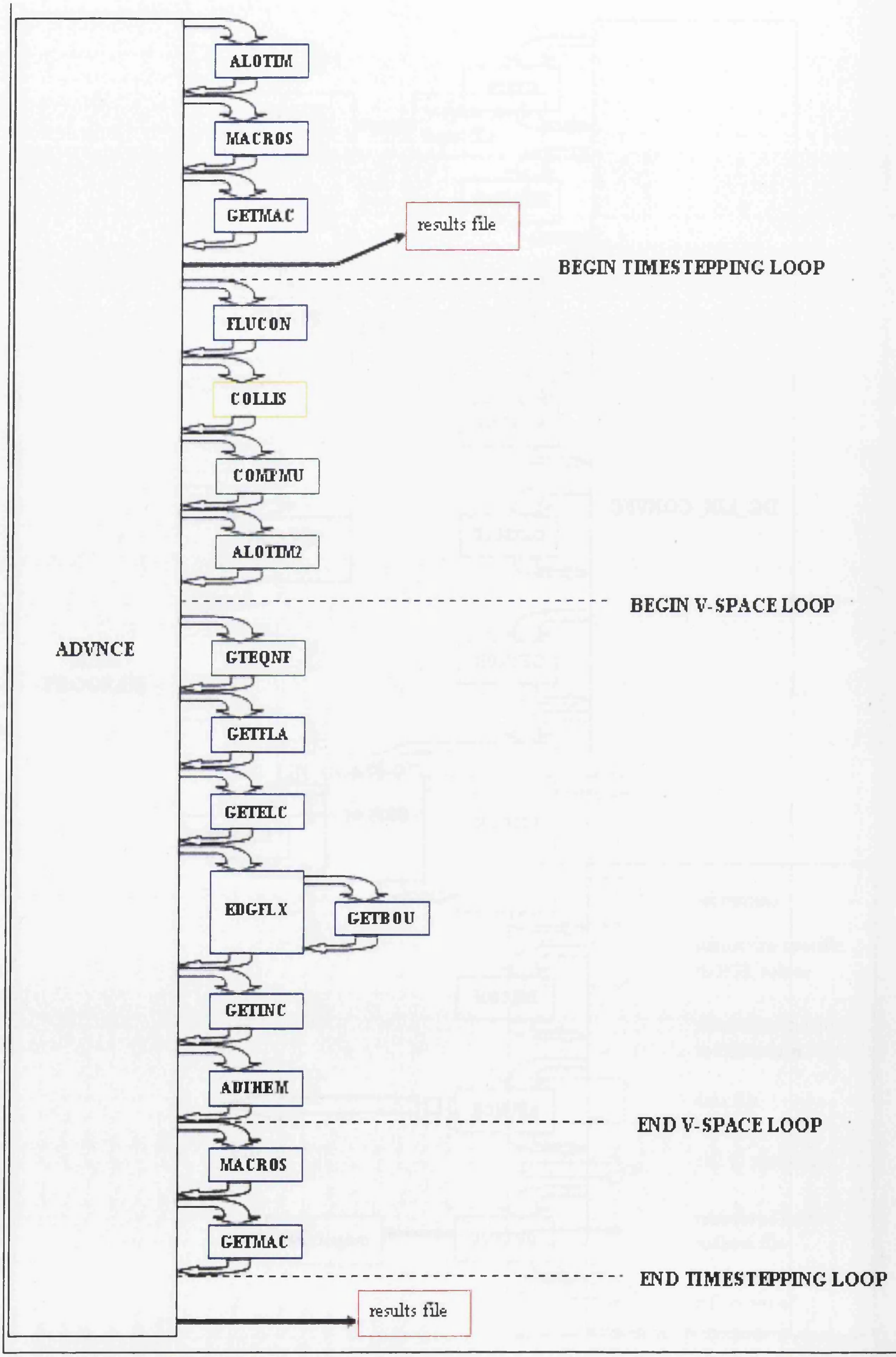
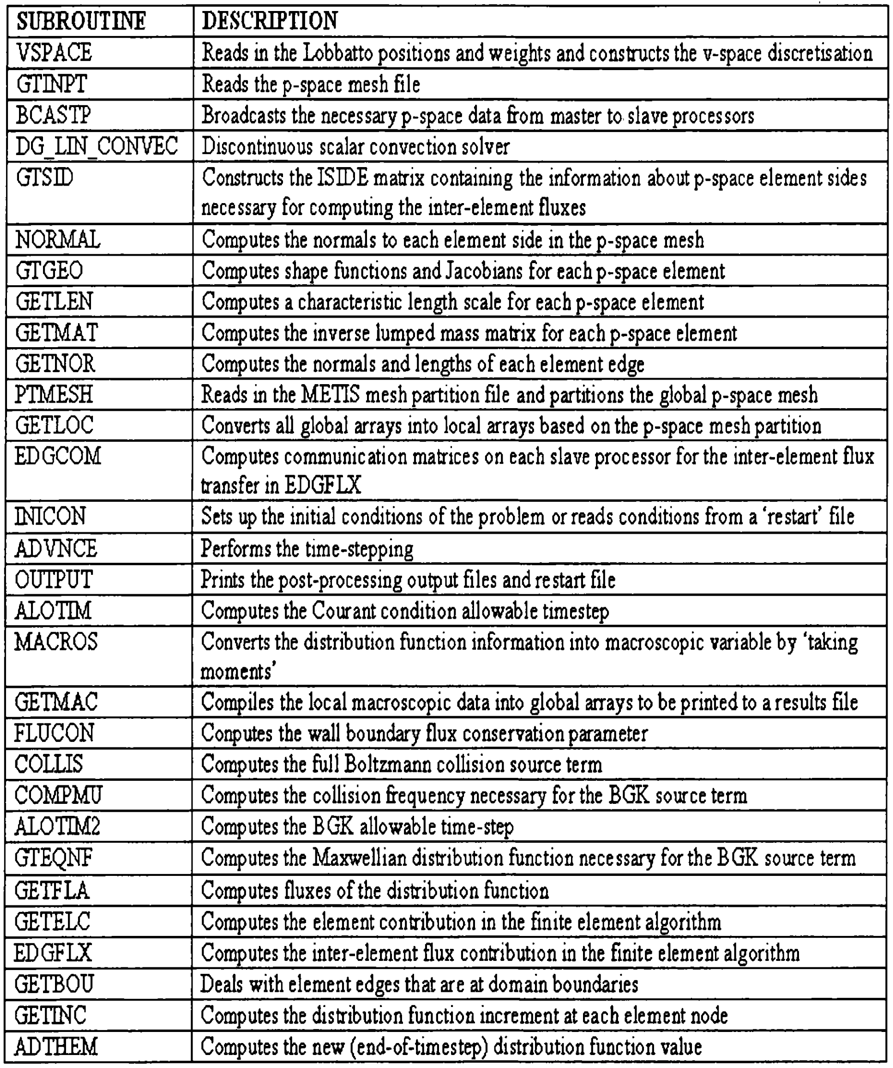


Figure D.3: Program Flow Chart (part 3)



SUBROUTINE	DESCRIPTION
VSPACE	Reads in the Lobbatto positions and weights and constructs the v-space discretisation
GTINPT	Reads the p-space mesh file
BCASTP	Broadcasts the necessary p-space data from master to slave processors
DG_LIN_CONVEC	Discontinuous scalar convection solver
GTSID	Constructs the ISIDE matrix containing the information about p-space element sides necessary for computing the inter-element fluxes
NORMAL	Computes the normals to each element side in the p-space mesh
GTGEO	Computes shape functions and Jacobians for each p-space element
GETLEN	Computes a characteristic length scale for each p-space element
GETMAT	Computes the inverse lumped mass matrix for each p-space element
GETNOR	Computes the normals and lengths of each element edge
PTMESH	Reads in the METIS mesh partition file and partitions the global p-space mesh
GETLOC	Converts all global arrays into local arrays based on the p-space mesh partition
EDGCOM	Computes communication matrices on each slave processor for the inter-element flux transfer in EDGFLX
INICON	Sets up the initial conditions of the problem or reads conditions from a 'restart' file
ADVNCE	Performs the time-stepping
OUTPUT	Prints the post-processing output files and restart file
ALOTIM	Computes the Courant condition allowable timestep
MACROS	Converts the distribution function information into macroscopic variable by 'taking moments'
GETMAC	Compiles the local macroscopic data into global arrays to be printed to a results file
FLUCON	Conputes the wall boundary flux conservation parameter
COLLIS	Computes the full Boltzmann collision source term
COMPMU	Computes the collision frequency necessary for the BGK source term
ALOTIM2	Computes the BGK allowable time-step
GTEQNF	Computes the Maxwellian distribution function necessary for the BGK source term
GETFLA	Computes fluxes of the distribution function
GETELC	Computes the element contribution in the finite element algorithm
EDGFLX	Computes the inter-element flux contribution in the finite element algorithm
GETBOU	Deals with element edges that are at domain boundaries
GETINC	Computes the distribution function increment at each element node
ADTHEM	Computes the new (end-of-timestep) distribution function value

Figure D.4: Subroutine Descriptions

Part II

References

Bibliography

[1] S.A. Trugman A.J.Taylor, *Analytic solution of the Boltzmann equation with applications to electron transport in inhomogeneous semiconductors*, Phys. Rev. B **33** (1986), no. 8, 5575 – 5584.

[2] F.J. Regan S.M. Anadakrishnan, *Dynamics of atmospheric re–entry*, AIAA Education Series, American Institute of Aeronautics and Astronautics, Inc., New York, 1993.

[3] V.V. Aristov, *Direct methods for solving the Boltzmann equation and study of nonequilibrium flows*, Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001.

[4] Q. Behrend, *Solid-fluid boundaries in particle suspension simulation via the lattice-Boltzmann method*, Phys. Rev. E. **52** (1995), 1164–1175.

[5] G.A. Bird, *Molecular gas dynamics and the direct simulation of gas flows*, Clarendon Press, Oxford, 1994.

[6] A. Palczewski J. Schneider A.V. Bobylev, *A consistency result for a discrete-velocity model of the Boltzmann equation*, SIAM J. Numer. Anal. **34**.

[7] Ludwig Boltzmann, *Lectures on gas theory*, Dover Publications, New York, 1964.

[8] R. Chen R.K. Argawal F.G. Cheremisin Y. Bondar, *A Comparative Study of Navier-Stokes, Burnett, DSMC, and Boltzmann Solutions for Hypersonic Flow Past 2-D Bodies* , 45th AIAA Aerospace Sciences Meeting and Exhibit, 07-0205, 2007.

[9] R. Lohner K. Morgan M. Vahdati J. P. Boris D. L. Book, *FEM-FCT: Combining unstructured grids with high resolution*, Communications in Applied Numerical Methods **4** (1988), 717–729.

[10] A. Bouhadji M. Braza, *Organised modes and shock-vortex interaction in unsteady viscous transonic flows around an aerofoil part ii: Reynolds' number effect*, Computers and Fluids **32** (2003), 1261–1281.

[11] A. Brubeau P. Sagaut Ch.-H. Bruneau, *A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods*, J. Comp. Phys. **169**.

[12] A. Burbeau P. Sagaut Ch.-H. Bruneau, *A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods*, J. Comp. Phys. **169** (2001), 111–150.

[13] L. Pareschi R.E. Caflisch, *An implicit Monte Carlo method for rarefied gas dynamics*, J. Comp. Phys. **154** (1999), 90–116.

[14] M. K. Gobbert T. S. Cale, *A kinetic transport and reaction model and simulator for rarefied gas flow in the transition*, J. Comp. Phys. **213**.

[15] S.G. Webster M.K. Gobbert T.S. Cale, *Transient 3d/3d transport and reactant-wafer interactions: Adsorption and desorption*, Rapid Thermal and Other Short-Time Processing Technologies III **2002** (2002), 81–88.

[16] A. Jameson D.A. Caughey, *A finite volume method for transonic potential flow calculations*, Proc. AIAA Third Computational Fluid Dynamics Conference, Albuquerque, 1977, pp. 35–54.

[17] C. Cercignani, *Mathematical methods in kinetic theory*, Plenum Press, New York, 1969.

[18] C. Cercignani, *The Boltzmann equation and its applications*, Springer-Verlag, New York, 1987.

[19] C.K. Chu, *Kinetic-theoretic description of the formation of a shock wave*, Physics of Fluids **8** (1965), 12–22.

[20] P.G. Ciarlet, *The finite element method for elliptic problems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.

[21] B. Cockburn, *Discontinuous Galerkin methods*, Journal of Applied Mathematics and Mechancics **83**.

[22] B. Cockburn, *Discontinuous Galerkin methods for convection-dominated problems, in: High-order methods for computational physics, edited by T. Barth and H. Deconink, lecture notes in computational science and engineering*, vol. 9, pp. 69–224, Springer-Verlag, Berlin, 1999.

[23] S. Chapman T.G. Cowling, *The mathematical theory of non-uniform gases, 3rd ed.*, Cambridge University Press, Cambridge, 1970.

[24] J.P. Boris D.L.Book, *Flux-corrected transport, i, SHASTA - a fluid transport algorithm that works*, J. Comp. Phys. **11** (1973), 38.

[25] J. Donea, *A Taylor-Galerkin method for convective transport*, Int. J. Numer. Meth. Fluids. **20** (1984), 101–120.

[26] S. Chen G.D. Doolen, *Lattice Boltzmann method for fluid flows*, Annu. Rev. Fluid Mech. **30** (2001), 329–364.

[27] R.A. East, *A reusable space-rescue vehicle: re-entry simulation*, Phil. Trans. R. Soc. Lond. A **357** (1999), 2177–2195.

[28] A. Einstein, *On the motion - required by the molecular kinetic theory of heat - of small particles suspended in a stationary liquid*, Annalen der Physik **17** (1905), 549–560.

[29] S.M. Deshpande et. al., *A 3-dimensional upwind Euler solver using kinetic flux vector splitting method*, Proceedings of 13th International Conference on Computaional Fluid Dynamics (1992).

[30] C. Greenhough R.F. Fowler, *Partitioning methods for unstructured finite element meshes*, Report RAL-94-092, Rutherford Appleton Laboratory, Didcot (1994).

[31] N.C. Freeman, *Non-equilibrium flow of an ideal dissociating gas*, J. Fluid Mech. **4** (1958), 407.

[32] J.K.Harvey M.A. Gallis, *Review of code validation studies in high-speed low-density flows*, Journal of Spacecraft and Rockets **37** (2000), 8–20.

[33] J.A. CARRILLO C. Cercignani I.M. Gamba, *Steady states of a Boltzmann equation for driven granular media*, Phys. Rev. **62** (2000), 7700–7707.

[34] H. Grad, *Encyclopaedia of physics*, vol. XII, Springer-Verlag, Berlin, 1958.

[35] W.G. Habashi G. Guevremont M.F. Peeters M.M. Hafez, *Finite–element solutions of the compressible Navier–Stokes equations*, AIAA **25** (1987), 944–948.

[36] S. Harris, *An introduction to the theory of the Boltzmann equation*, Holt, Rinehart, and Winston, New York, 1971.

[37] B. Evans K. Morgan O. Hassan, *A discontinuous Taylor-Galerkin finite element approach for solution of the collisionless Boltzmann equation*, Proceedings of the 14th ACME Conference: Queen's University, Belfast (2006), 135–138.

[38] J. Grooss J.S. Hesthaven, *A levelset discontinuous Galerkin method for free surface flows*, Comput. Methods Appl. Mech. Engrg. **195** (2006), 3406–3492.

[39] W.H. Reed T.R. Hill, *Triangular mesh methods for the neutron transport equation*, Tech. Rep. LA-UR-73-479 (1973).

[40] E.R. Hillje, *Entry aerodynamics at lunar return conditions obtained from the flight of Apollo 4(AS-501)*, NASA TN D-5399 (1969).

[41] C. Hirsch, *Numerical computational of internal and external flows: Vol 1*, John Wiley and Sons, New York, 1990.

[42] Jr H.J. Allen A.J. Eggers, *A study of the motion and aerodynamic heating of ballistic missiles*, NACA Report 1381 (1958).

[43] D.A. Caughey A. Jameson, *Progress in finite volume calculations for wing-fuselage combinations*, AIAA Journal **18** (1980), 1281–1288.

[44] G. May A. Jameson, *Improved gaskinetic multigrid method for three-dimensional computation of viscous flow*, 17th AIAA Computational Fluid Dynamics Conference, Toronto (2005), 1–18.

[45] Y. Ruan A. Jameson, *Gas-kinetic BGK method for three-dimensional compressible flows*, AIAA-2002-0550 (2002).

[46] J.D. Anderson Jr., *A survey of modern research in hypersonic aerodynamics*, 17th American Intitute of Aeronautics and Astronautics, Plasma Dynamics and Lasers Conference (1984).

[47] John D. Anderson Jr., *Hypersonic and high temperature gas dynamics*, AIAA, Reston, Virginia, 2000.

[48] W.G. Vincenti C.H. Kruger, *Introduction to physical gas dynamics*, John Wiley and Sons, Inc.

[49] A.J.C. Ladd, *Numerical simulation of particular suspensions via a discretised Boltzmann equation. Part 1. theoretical foundation*, J. Fluid Mech. **271** (1994), 285–309.

[50] A.J.C. Ladd, *Numerical simulation of particular suspensions via a discretised Boltzmann equation. Part 2. numerical results*, J. Fluid Mech. **271** (1994), 311–339.

[51] N.P. Weatherill O. Hassan K. Morgan J.W. Jones B. Larwood, *Towards fully parallel aerospace simulations on unstructured meshes*, Engineering Computations **18** (2001), 347–375.

[52] P. le Tallec F. Mallinger, *Coupling Boltzmann and Navier-Stokes equations by half fluxes*, J. Comp. Phys. **136** (1997), 51–67.

[53] C.E. Leith, *Methods in computational physics*, 4 ed., Academic Press, 1965.

[54] R. Cornubert D. Dhumieres D. Levermore, *A Knudsen layer theory for lattice gases*, Physica D **47** (1991), 291–297.

[55] B. Khonalatte P. Leyland, *New finite-element-based fluctuation splitting schemes*, Int. J. Num. Meth. Eng. **27** (1998), 229–239.

[56] Y. Xuan K. Yu Q. Li, *Investigation on flow and heat transfer of nanofluids by the thermal lattice Boltzmann model*, Prog. Comp. Fluid Dyn. **5**.

[57] M.J. Lighthill, *Dynamics of a dissociating gas. Part i. equilibrium flow*, J. Fluid Mech. **2**.

[58] H. Li X. Lu H. Fang Z. Lin, *Simulation of multi-particle suspensions in a quasi-two-dimensional symmetric stenoic artery with lattice Boltzmann method*, Prog. Comp. Fluid Dyn. **5**.

[59] J. Reese M. Gallis D. Lockerby, *New directions in fluid dynamics: non-equilibrium aerodynamic and microsystem flows*, Phil. Trans. R. Soc. Lond. A **361** (2003), 2967–2988.

[60] K. Morgan J. Peraire R. Lohner, *Adaptive finite element flux corrected transport techniques for CFD*, ICASE Symposium on Finite Elements Theory and Applications (1986).

[61] P. Tallec F. Mallinger, *Coupling Boltzmann and Navier-Stokes equations by half fluxes*, J. Comp. Phys. **136** (1997), 51–67.

[62] L. Darius P. Gonzalez-Vera F. Marcellan, *Gaussian quadrature formulae on the unit circle*, Journal of Computational and Applied Mathematics **140** (2002), 159–183.

[63] N.P. Weatherill J.S. Mathur M.J. Marchant, *An upwind kinetic flux vector splitting method on general mesh topologies*, Int. J. Num. Meth. Eng. **37** (1994), 623–643.

[64] H. Yamaguchi N. Tsuboi Y. Matsumoto, *3D DSMC simulation of rarefied hypersonic flow*, RAREFIED GAS DYNAMICS: 22nd International Symposium. AIP Conference Proceedings **585** (2001), 764–771.

[65] K. Koura H. Matsumoto, *Variable soft sphere model for inverse-power-law or Lennard-Jones potential*, Phys Fluids A **3** (1991), 2459–2465.

[66] K. Koura H. Matsumoto, *Variable soft sphere model for air species*, Phys Fluids A **4** (1992), 1083–1085.

[67] J.C. Maxwell, *On the dynamic theory of gases*, Phil. Trans. Soc.

[68] L. Mieussens, *Discrete-velocity models and numerical schemes for the Boltzmann-BGK equation in plane and axisymmetric geometries*, J. Comp. Phys. **162** (2000), 429–426.

[69] P.L. Bhatnagar E.P. Gross M.Krook, *Model for collision processes in gases, i small amplitude processes in charged and neutral one-component systems*, Phys. Rev. **94** (1954), 511–524.

[70] M.M.Kuznetsov, *Analystical solution of the Boltzmann equation in a Knudsen layer*, J. App. Mech. Tech. Phys. **4**.

[71] J. Peraire J. Peiro K. Morgan M. Vahdati R.C. Molina, *Hypersonic flow computation around reentry vehicles*, AGARD 70th Fluid Dynamics Panel Meeting nd Symposium Theoretical and Experimental Methods in Hypersonic Flows, Turin (1992).

[72] J.J. Monaghan, *Smoothed particle hydrodynamics*, Rep. Prog. Phys. **68** (2005), 1703–1759.

[73] R. Thareja J.R. Stewart K. Morgan, *Euler and Navier-Stokes solutions for hypersonic flows*, Proceedings of the Seventh International Conference on Finite Element Methods in Flow Problems, Huntsville, AL (1989), 180–186.

[74] J.N. Moss and G.A. Bird, *Direct simulation of transitional flow for hypersonic re-entry conditions*, AIAA **Paper no. 84-0223** (1984).

[75] H.M. Mott-Smith, *The solution of the Boltzmann equation for a shock wave*, Physical Review **82** (1951), no. 6, 885–893.

[76] E.P. Muntz, *Rarefied gas dynamics*, Annual Rev. Fluid Mech. **21** (1989), 387.

[77] E. Fatima F. Odeh, *Upwind finite difference solution of Boltzmann equation applied to electron transport in semiconductor devices*, J. Comput. Phys. **108** (1993), 209–217.

[78] N. Satofuka K. Morinishi T. Oishi, *Numerical solution of the kinetic model equations for hypersonic flows*, J. Comp. Mech. **11** (1993), 452–464.

[79] K. Morgan J. Peraire, *Unstructured grid finite-element methods for fluid mechanics*, Rep. Prog. Phys. **61** (1998), 569–683.

[80] P.-O. Persson J. Peraire, *Sub-cell shock capturing for discontinuous Galerkin methods*, 44 th AIAA Aerospace Sciences Meeting and Exhibit (2006).

[81] J.B. Perrin, *Brownian movement and molecular reality*, Dover Phoenix Editions, 2005.

[82] C. Johnson U. Navert J. Pitkaranta, *Finite element methods for hyperbolic problems*, Comput. Methods Appl. Mech. Engrg. **45** (1984), 285–312.

[83] D.I. Pullin, *Direct simulation methods for compressible inviscid ideal gas flow*, J. Comput. Phys. **34** (1978), 231–244.

[84] L.F. Richardson, *The approximate arithmetic solution by finite differences of physical problems involving differential equations with application to stresses in a masonry dam*, Trans. Roy. London Ser. A 210 **307** (1910).

[85] J.A. Fay F.R. Riddell, *Theory of stagnation point heat transfer in dissociated air*, J. Aero. Sci. **25** (1958), 73.

[86] C. Ringhofer, *Sapce-time discretization of series expansion methods for the Boltzmann transport equation*, SIAM . Numer. Anal. **38** (2000), 442–465.

[87] F.Filbet G. Russo, *High order numerical methods for the space nonhomogeneous Boltzmann equation*, J. Comp. Phys. **186** (2003), 457–480.

[88] L. Pareschi G. Russo, *Numerical solution of the Boltzmann equation i: Spectrally accurate approximation of the collision operator*, SIAM J. Numer. Anal. **37** (2000), 1217–1245.

[89] S-C. Sheen, *Computation of convective-diffusion problems differenced with high-order upwind schemes*, Numerical Heat Transfer, Part B: Fundamentals **40** (2001), 163–178.

[90] J. Qiu M. Dumbser C-W. Shu, *The discontinuous Galerkin method with Lax-Wendroff type time discretisations*, Comput. Methods Appl. Mech. Engrg **194** (2005), 4528–4543.

[91] M. Zhang C-W Shu, *An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods*, Computers and Fluids **34** (2005), 581–592.

[92] W. Gropp E. Lusk A. Skjellum, *Using MPI:portable parallel programming with the message passing interface*, The MIT Press, Cambridge, MA, 1994.

[93] N. Rossi S. Ulbertini G. Bella S. Succi, *Unstructured lattice Boltzmann method in three dimensions*, Int. J. for Numer. Meth. Fluids **49** (2005), 619–633.

[94] S. Succi, *The lattice Boltzmann equation for fluid dynamics and beyond*, p. 72, Oxford University Press, Oxford, 2001.

[95] K. Xu M. Mao L. Tang, *A multidimensional gas-kinetic BGK scheme for hypersonic viscous flow*, J. Comp. Phys. **203** (2005), 405–421.

[96] O. Hassan K.Morgan E.J. Probert R.R. Thareja, *Adaptive unstructured mesh methods for steady viscous flow*, 10th AIAA Computational Fluid Dynamics Conference, Honolulu (1991), 125–133.

[97] V. Thomee, *Galerkin finite element method for parabolic problems*, 2 ed., vol. 25, Springer Series in Computatinal Mechanics, 2006.

[98] M.J. Turner R.W. Clough H.C. Martin L.J. Topp, *Stiffness and deflection analysis of complex structures*, J. Aero. Sci. **23** (1956), no. 9, 805–823.

[99] L. Pareschi S. Trazzi, *Numerical solution of the boltzmann equation by time reed Monte Carlo (TRMC) methods*, Int. J. Numer. Meth. Fluids. **48** (2005), 947–983.

[100] L. Trefethen, *Spectral methods in MATLAB*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2000.

[101] M. Delfour F. Trochu, *Discontinuous Galerkin methods for the approximation of optimal control problems governed by hereditary differential systems*, Distributed Parameter Systems: Modelling and Identification (1978), 256–271.

[102] J.-S. Wu K.-C. Tseng, *Parallel DSMC method using dynamic domain decomposition*, Int. J. Num. Meth. Eng. **63** (2005), 37–76.

[103] R. Lohner K. Morgan J. Peraire M. Vahdati, *Finite element flux-corrected transport (fem-fct) for the Euler and Navier-Stokes equations*, Int. J. for Numer. Meth. Fluids **7** (1987), 1093–1109.

[104] C.E. Siewert D. Valougeorgis, *The temperature-jump problem for a mixture of two gases*, Journal of Quantitative Spectroscopy and Radiative Transfer **70** (2001), 307–319.

[105] A. Harten P. Lax B. van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM **25** (1983), 35–61.

[106] L.H. Abbott A.E. von Doenhoff, *Theory of wing sections*, Dover, 1959.

[107] R. Illner W. Wagner, *A random discrete velocity model and approximation of the Boltzmann equation*, J. Stat Phys. **70**.

[108] M. Hafez E. Wahba, *Simulations of viscous transonic flows over lifting aerfoils and wings*, Computers and Fluids **36** (2007), 39–52.

[109] B. Shi N. He N. Wang, *A unified thermal lattice BGK model for Boussinesq equations*, Prog Comput Fluid Dyn **5**.

[110] K. Morgan P.J. Brookes O. Hassan N.P. Weatherill, *Parallel processing for the simulation of problems involving scattering of electromagnetic waves*, Comput. Methods Appl. Mech. Engrg **152** (1998), 157–74.

[111] D.M. Hawken H.R. Tamaddon-Jahromi P. Townsend M.F. Webster, *A Taylor-Galerkin based algorithm for viscous incompressible flow*, Int. J. Num. Meth. Fluids **10** (1990), 327–351.

[112] P.D. Lax B. Wendroff, *Difference schemes for hyperbolic equations with high order of accuracy*, Comm. Pure App. Math. **17** (1964), 381–398.

[113] D. Yu R. Mei W.Shyy, *Improved treatment of the open boundary in the lattice Boltzmann equation*, Prog. Comp. Fluid Dyn. **5**.

[114] www.llnl.gov.

[115] X. Zhong, *High-order finite-difference schemes or numerical simulation of hypersonic boundary-layer transition*, J. Comp. Phys. **144** (1998), 662–709.

[116] D.P. Ziegler, *Boundary conditions for lattice Boltzmann simulations*, J.Stat. Phys. **71** (1993), 1171–1177.