Micro-Meso Coupling for Hydrodynamic Simulations

A Thesis
Submitted for the Degree of
Master of Science (Engineering)

by

Akshay Chandran

Engineering Mechanics Unit
Jawaharlal Nehru Centre for Advanced Scientific Research
(A Deemed University)
Bangalore – 560 064

NOVEMBER 2021

Dedicated to my family

DECLARATION

I hereby declare that the matter embodied in the thesis entitled "**Micro-Meso Coupling for Hydrodynamic Simulations**" is the result of investigations carried out by me at the Engineering Mechanics Unit, Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore, India under the supervision of **Prof. Santosh Ansumali** and that it has not been submitted elsewhere for the award of any degree or diploma.

In keeping with the general practice in reporting scientific observations, due acknowledgment has been made whenever the work described is based on the findings of other investigators.

Akshay Chandran

CERTIFICATE

I hereby certify that the matter embodied in this thesis entitled "**Micro-Meso Coupling for Hydrodynamic Simulations**" has been carried out by **Mr. Akshay Chandran** at the Engineering Mechanics Unit, Jawaharlal Nehru Centre for Advanced Scientific Research, Bangalore, India under my supervision and that it has not been submitted elsewhere for the award of any degree or diploma.

Santosh Ansumali

Prof. Santosh Ansumali
(Research Supervisor)

Acknowledgements

This thesis could not be completed without the support of a few people. Firstly, I would like to thank Prof. Santosh Ansumali for giving me this opportunity and for introducing me to the world of research. His constant guidance has been important in the many projects I have had the opportunity to work on. I would also like to thank him for his support in academic and personal situations as well. This project would not have been a possibility without the idea and the motivation from Prof. Berni J. Alder. His comments and suggestions have been invaluable in the development of this project and in this field in general. I am also indebted to Prof. Sauro Succi for his optimism and his feedback from time to time, without which this work would not have attained this form. Lastly, I am grateful to Praveen "RD3Q41" Kolluru who has been a part of the development of this project from its inception.

I am also grateful to Prof. Vishwanathan Kumaran for his valuable inputs on the DSMC simulations. I would also like to thank the PARAM Yukti Computing facility at JNCASR and Sankhyasutra labs for providing access to their clusters.

I would like to thank Prof. Ganesh Subramanian, Prof. K.R. Sreenivas, Prof. Meheboob Alam and Dr. Diwakar S. Venkatesan for their wonderful courses.

I thank my friends - Anjana, Sabarish, Mohit, Janaky and Ashish for making my life easier at JNCASR. My DNS lab members - Praveen, Shaurya, Samarth, Atif, Rafi and Vybhav for maintaining a balance in the lab.

Finally, I am grateful for my parents and my sister for their unconditional support and being with me through all the ups and downs.

I would also like to thank our administrative officer Mr Joydeep Deb, former administrative officer Mr A.N. Jayachandra, along with the academic coordinator Dr. Panneer Selvam, employees of the library, CompLab, hostel office, and other JNC staff for their cooperation.

Abstract

We present a particle-continuum coupling for simulations in the laminar and turbulent flow regimes. The two-way coupling in the hybrid model is achieved via a direct Simulation Monte Carlo layer near walls and a Lattice-Boltzmann RD3Q41 model in the bulk. The lifting of macroscopic to microscopic dynamics is done via a novel projection scheme based on the idea of quasi-equilibrium manifold incorporating stress and heat flux as fields. The coupled solver is validated by performing simulations on the canonical plane Poiseuille and Couette flows. Turbulent flow simulation performed using minimal compute resources shows the ability of the solver to predict the regeneration cycles of coherent structures. This hybrid coupling represents a novel method for accurate description of flow physics in various hydrodynamic regimes, where stand-alone solvers fall short due to high computational expense like in DSMC or due to lack of high Knudsen flow physics in LB.

List of Figures

List of Tables

Contents

Chapter 1
Introduction

At the continuum scale, fluid dynamics is well described by the Navier-Stokes-Fourier (NSF) equations(Batchelor, 2000). For dilute gases, which can be safely assumed to be composed of structureless point particles, this description is valid only when the system size is much larger than the mean free path(Batchelor, 2000; Chapman & Cowling, 1990). This is typically characterized in terms of the Knudsen number, defined as $\mathrm{Kn} = \lambda/L$, where λ is the mean free path and L is the characteristic length of the system. The NSF description, where one assumes locally thermodynamic equilibrium, is valid only in the limit of $\mathrm{Kn} \to 0$. A more general description of dilute gas dynamics is provided by the Boltzmann equation which describes the dynamics even far away from the thermodynamic equilibrium and is valid at all Knudsen numbers(Chapman & Cowling, 1990). Thus, the Boltzmann equation provides a statistical description of the time evolution of the point particle system in terms of the single-particle distribution function, with its evolution represented as a sequence of free flight and binary collision described by an integro-differential term.

In the last two decades, interest in the numerical solution of the Boltzmann equation has been revived. This is partially driven by MEMS/NEMS applications (Nie *et al.*, 2002), where characteristic flow scales are no longer much larger than the molecular mean free path, requiring modeling beyond the Navier–Stokes level of description. In this regard, an important tool is the direct simulation Monte Carlo (DSMC) method (Bird, 1963; Bird & Brady, 1994), a particle-based, numerical scheme for solving the nonlinear Boltzmann equation. In this method, unlike force-based (Alder & Wainwright, 1959) or event-driven Molecular Dynamics (MD) (Rapaport, 2004; Lubachevsky, 1991) for the Hard-Sphere (HS) gas, one does not track individual collisions. Instead, DSMC models particle motion as a sequence of free flight and random collisions. In this method, one generates collisions stochastically with scattering rates and post-collisional velocity distributions determined from the kinetic theory of a dilute HS gas. Because of this simplification made in DSMC for elevating particles to a state further upstream in time and by not representing physical collisions like in hard-sphere MD, the computational cost required for DSMC is typically an order of magnitude lower than that of an MD simulation.

As the spatial domain is discretized into computational cells which contains particles that can collide with any other particle within the cell, the size of a cell is maintained to be atmost $\Delta x \sim \lambda$. For low Knudsen number simulations, the number of cells and thence the number of particles increases, and therefore the computational cost scales as Kn^{-3}. Additionally, with restrictions on the time stepping of the simulation scaling as the Mean free time (τ), there is a restriction on the Δt of the simulation. Thus, DSMC simulations in the continuum regime (where $\mathrm{Kn} \ll 1$) are computationally expensive. In the hydrodynamic regime, one is often interested in probing high Reynolds number flows, so traditionally the role of DSMC was limited for high Knudsen simulations. At this point, it should be reminded that in the continuum limit, two relevant parameters are the Reynolds number Re and the Mach number Ma and they are related to the Knudsen number as $\mathrm{Kn} \sim \mathrm{Ma}/\mathrm{Re}$. Thus, out of the three only two are independent parameters. It is also evident that as Ma is raised at a fixed Re, Knudsen number increases, and thus the computational need for DSMC decreases. Furthermore, the importance of the statistical scatter in the method ($\sim O(1/\sqrt{N})$ with N being the number of particles in a computational cell) decreases as the mean flow is much more dominant at higher Mach. With $\lambda \ll l_k$ for incompressible flows, the gap between the Kolmogorov length scale (l_k), the relevant length scale for turbulence, and the mean free path shrinks as the Mach number is raised. To study the effect of this, recently DSMC simulations were used for high Mach number turbulent flows

(Pradhan & Kumaran, 2016; Gallis *et al.*, 2021).

Along similar lines, it was argued recently that due to increased computing power and with better softwares, DSMC simulations of turbulence at finite Mach number is a distinct possibility. However, large-scale computing resources were required for decay to turbulence and minimal turbulent Couette flow (MCF) simulations(Gallis *et al.*, 2018, 2017) at high Reynolds number and finite Mach number. These simulations were performed at a moderate Reynolds number of 400 and a Mach number of 0.3, with billions of particles in the domain and have given good initial results after using millions of CPU hours on a multi peta-scale HPC cluster.

These works essentially show that DSMC simulations are becoming more powerful due to an increase in computing power. However, still a substantial speed-up (a few orders of magnitude) is required for these to become practical for low Mach number flows. As a DSMC code is often compute bound, assuming that Moore's law(Moore *et al.*, 1965) will hold for the next few years one may expect that in the next decade an order of magnitude improvement might come from an increase in computing power itself. However, one may need to consider other routes to increase the performance of the method further. In this regard, one may ask are DSMC codes optimally written so far? In recent years, a lot of progress has been made with respect to memory optimization of scientific codes(Shet *et al.*, 2013). One may ask whether community software frameworks for DSMC such as SPARTA®(Klothakis *et al.*, 2016) can be optimized further?

One simple and popular method is through parallelization, with each processor owning a portion of the domain. Such methods are typically termed to be domain decomposed. With a 2D decomposition known as pencil or block decomposition, one is able to allocate a much larger number of cores than a 1D or slab decomposition. And through this procedure one can gain significantly by using an MPI parallelized pencil decomposed particle code. Such codes need to be efficiently parallelized for optimal performance and to avoid redundant communications between processors.

Another way of improving performance of particle simulations is by decreasing the resolution by increasing cell sizes and correct for the changes in transport parameters that arise because of an increase in the effective mean free path (Alexander *et al.*, 1998). In open-source DSMC solvers like SPARTA®, such viscous and thermal corrections are not implemented. A similar approach in reparametrization is by increasing the time stepping (Δt) of the simulation and using similar corrections arising thereof (Garcia & Wagner, 2000). The latter approach allows one to reach steady states much faster and with an increase in the effective time of averaging, one can achieve a higher level of noise reduction for the same number of ensembles averaged. For example, using our inhouse DSMC code, we obtained an order of magnitude decrease in the total time of the simulation by increasing the time stepping by a factor 40 for a domain containing 50 Million particles. Since most particle codes are written using a parallel architecture, larger time steps could result in a decreased overall efficiency due to much higher particle crossings across processor boundaries, thereby increasing the communication load on parallelization. Significant gains can still be made in such scenarios by analyzing the time split by subroutines and through efficient algorithms. Other than such possible intrinsic improvements in DSMC itself, one may think of coupling it with other mesoscopic schemes that are computationally more efficient.

For example, in recent years Lattice-Boltzmann (LB) (Succi, 2001; Chen & Doolen, 1998) has emerged as an efficient alternative to conventional fluid dynamics solvers. Even though initially, the method was limited to low Mach number and low Knudsen number flows, in recent years the method has matured to handle high Mach and high Knudsen flows(Ansumali *et al.*, 2007; Frapolli *et al.*, 2015; Namburi *et al.*, 2016; Atif *et al.*, 2018). It is therefore natural to ask, whether one can gain from such a coupling. Any such coupling would require a communication strategy between the two solvers at the interface of the coupling. There are several initial attempts to couple LB with a particle-based method. However, all of them have worked with lower-order LB models. Furthermore, they have done only a one-way communication between

the solvers (Montessori *et al.*, 2020; Di Staso *et al.*, 2016) and their coarser solvers run throughout the domain rather than restricting them to a particular region of interest. Such methods could get away with particle fluctuations at the interface of the coupling arising because of a free surface boundary for the particle solver. Additionally, such coupling have restricted their mesoscopic solvers also to have a resolution of $\Delta x \approx \lambda$ (Montessori *et al.*, 2020; Di Staso *et al.*, 2016). This does not utilise the efficient features provided by mesoscopic solvers.

In this thesis, we investigate two questions: How to improve the intrinsic performance of DSMC codes through inherent physics and data-structure based changes? And how to introduce a two-way coupling for LB and DSMC and what is the performance gain in doing so. This report highlights the progress made in this direction so far.

Finally, we describe the progress in coupling the DSMC-LB code. In this coupling, DSMC is implemented in a region near the walls and the LB solver is implemented in the bulk. This domain-restriction is computationally less expensive than a full-domain DSMC simulation and also provides the accuracy of a higher-order LB Model. These coupled fluid solvers are also used for Turbulent and High Knudsen flow simulations where the computational efficiency of LB and the ability of DSMC to capture wall layer phenomena is utilized. We shall discuss the communication routine that makes this coupling possible and benchmark this solver against standard cases.

1.1 DSMC - Lattice Boltzmann Coupling

The idea behind a particle-continuum coupling is to be able to delve in regimes which are typically hard to numerically adapt either a particle or a continuum solver individually. A stand-alone DSMC solver is computationally expensive for simulations in the turbulent flow regime and requires excessive compute resources (Gallis *et al.*, 2018, 2017). With modest resource availability, particle methods are extremely under-resolved for such simulations. Similarly, mesoscopic methods like standard LBM suffer from drawbacks in the high Mach or Knudsen number regimes leading to numerical instabilities. One resorts to a compromise by using double distribution function models with an unfaithful representation of the particle nature of kinetic theory (Shan, 1997; He *et al.*, 1998). Although, higher-order models (Ansumali *et al.*, 2007; Atif *et al.*, 2018; Namburi *et al.*, 2016; Yudistiawan *et al.*, 2010) have shown good results in such regimes.

Recent studies performed on such a micro-meso coupling using DSMC-LB (Di Staso *et al.*, 2016) and MPCD-LB (Montessori *et al.*, 2020) have shown good initial results for laminar and rarefied regimes. But these studies do not impose any restrictions on the domain of control for individual solvers and have not shown the capability of such a coupling in the turbulent regime and have also shown small deviations at the centerline from DSMC simulations in the transitional regime. Another drawback of such methods has been a grid ratio (defined as the ratio of number of LB nodes to DSMC cells) of 1 along all directions. Here, we show that for distances varying upto $\mathcal{O}(10\lambda)$, one can work with using a grid ratio less than 1.

Therefore, one can view this coupling as an alternative, yet more realistic implementation of a microscopic boundary condition necessary for an accurate description of hydrodynamics for mesoscopic solvers. In our simulations discussed in this thesis, we will be benchmarking our solvers on the standard plane channel flow geometry for canonical Poiseuille and Couette flows. Such a coupling can easily be extended to other geometries through the interface communication method reviewed in this thesis. Fig.(1.1) shows a sketch of the Knudsen layer formed near the wall in a plane Couette flow geometry with the DSMC region exerting control of this layer. The domain restriction of the two solvers with an interface for communication can be seen.

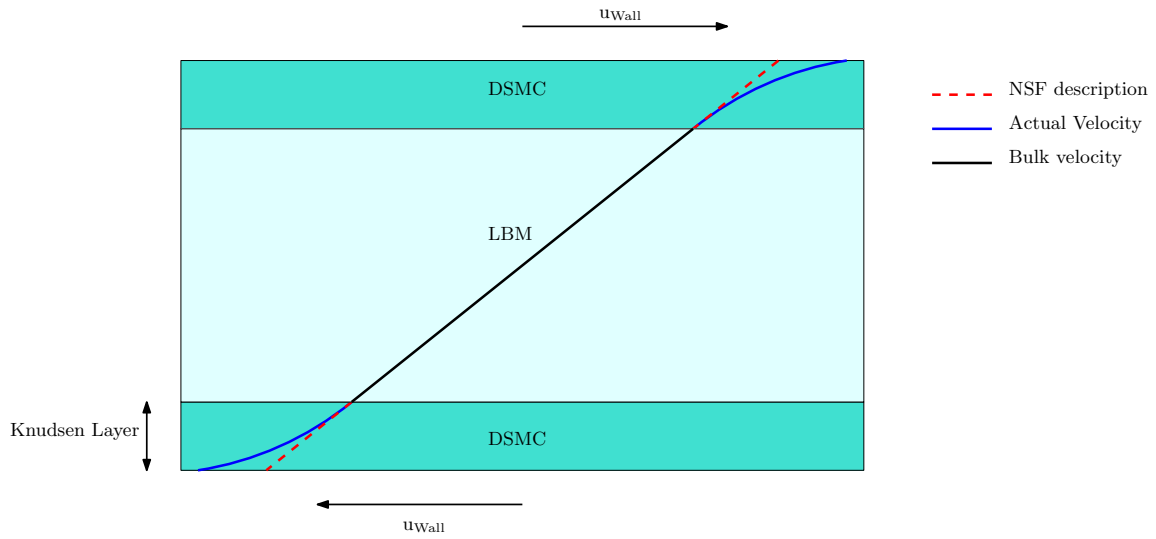

Figure 1.1: A schematic showing the near wall Knudsen layer in a plane Couette flow and the region of control for the two solvers.

To emphasize the importance of providing a wall layer microscopic solver, we look at the approximate velocity profile (Yudistiawan *et al.*, 2010) for a plane Couette flow at moderate Knudsen numbers,

$$u_y = \frac{U_1 + U_2}{2} + \frac{y}{\Theta}\frac{U_2 - U_1}{L} + \frac{1}{K_1}(U_2 - U_1)\sinh\left(\frac{y}{K_2\,\mathrm{Kn}\,L}\right). \tag{1.1}$$

Here, U_1, U_2 are the wall velocities, L is the channel width and K_1, K_2 & Θ are constants dependent on the choice of the discrete velocity model used to obtain the profile. With the Knudsen layer dominated by non-equilibrium effects, the Navier-Stokes description consequently fails in this layer. As a result, the velocity profile has an additional Knudsen dependent sine hyperbolic term quantifying this deviation. This term arises because of the anti-symmetry of the system around the centreline ($y = 0$) and thence the "sinh" term. Therefore, a microscopic picture should be able to capture this effect. With increasing Knudsen numbers, the width of this region increases and with it the contribution of the deviation to the velocity profile.

Therefore, we see that to have a molecular solver in the Knudsen layer that can capture this deviation can help us couple a micro and mesoscopic solver efficiently for flows beyond the continuum regime.

To summarize, we talk about the following points in our coupling scheme:

- How to communicate at the interface of a particle-continuum coupling?
- How does one control the instability arising in noise-free LBM due to the inherent thermal noise of DSMC?
- How can one extend the coupling to the various hydrodynamic regimes?

1.2 Outline of the thesis

In this thesis, we describe about a micro-meso coupling between DSMC and a Lattice Boltzmann model. In order to simulate flows in the turbulent regime, we try to understand how one can accelerate particle codes through parallelization and through appropriate transport coefficient reparametrization. We perform simulations on canonical test cases to benchmark our DSMC solver before moving on to a hydrodynamic coupling. Finally, we describe how such a coupling

can be achieved through a communication scheme based on projection of macroscopic fields into a quasi-equilibrium manifold.

This thesis is organized into the following chapters:

In Chapter 2, we provide a brief introduction to the kinetic theory of gases and discuss the properties of the Boltzmann equation. We also discuss on the molecular and turbulent scales of motion and the relationship between them.

In Chapter 3, we introduce the particle-based method used in our coupling. We briefly describe the connection between DSMC and the Boltzmann equation by means of the collision frequency. The methodology involved in performing a DSMC simulation is then described. We also introduce how one can improve the performance of the algorithm through existing modifications to transport parameters and through a parallelization scheme based on domain decomposition. Finally, we present benchmark results for our DSMC solver in various hydrodynamic regimes.

In Chapter 4, we discuss the particle-continuum coupling. We introduce why such a coupling is needed and the improvements one can achieve against either a particle or a standard mesoscopic solver. We discuss about how a coupling is performed and introduce the projection scheme used in ours. The steps involved in coupling and how one deals with the coupling interface is discussed. We present results in the laminar no-slip, rarefied and the turbulent regimes as a bechmark for our coupled solver.

In Chapter 5, we present the summary and conclusions of our study.

Chapter 2
Kinetic theory of gases

A typical system of molecules in random motion in the thermodynamic limit exhibit collective behavior in terms of measurable macroscopic quantities varying on much larger time scales than that of the molecular scale of motion. Macroscopic quantities such as temperature, pressure, etc., are well described at equilibrium via classical thermodynamics. Since the collective motion of these particles cause the dynamics of the macroscopic properties, one could find a link between macroscopic and microscopic dynamics. The kinetic theory therefore provides a statistical description of macroscopic dynamics from the molecular picture (Callen, 1998) for equilibrium and non-equilibrium systems.

One approach to understand this macroscopic description is through describing the motion of the 6N-dimensional phase space point, signifying the positions and velocities of the N particles. Such a description is governed by the Liouville equation. Alternatively, for dilute gases, one could define the motion of a single particle through its probability density using a single particle distribution function in its 6 dimensional phase space. The evolution of this single particle distribution function is governed by what is known as the Boltzmann equation (Grad, 1949).

The Boltzmann equation has been used as a route to understand macroscopic thermohydrodynamics even for systems far away from equilibrium. Equations governing fluid flow like the Euler and Navier-Stokes system of equations are obtained at the zeroth and first order in the smallness parameter Kn^{-1} through the Chapman-Enskog Expansion (Chapman & Cowling, 1990). Here $\mathrm{Kn} = \lambda/L$ is the Knudsen number defined as the ratio of the molecular mean free path and the macroscopic length scale. The Navier-Stokes description of fluid flow is valid in the continuum limit where $\mathrm{Kn} \to 0$. For flows in the rarefied regime, one resorts to the kinetic description for accurate prediction.

In this chapter, we briefly discuss the basic elements of kinetic theory and of molecular parameters important in understanding particle simulations. We start with a desciption of the length and time scales at the molecular level in Section 2.1. In Section 2.2, we introduce the single particle distribution function and its moments, used in describing macroscopic dynamics. In Section 2.3, we discuss the Boltzmann Equation, its properties and the Maxwell-Boltzmann distribution. Finally, we discuss the relationship between the microscopic scales with the scales in the turbulent regime in Section 2.4.

2.1 Mean Free Path and Mean Free Time

In this section, we briefly review the notion of the mean free path and mean free time for a hard-sphere system. From the kinetic theory description of elastic hard-spheres, one assumes a system composed of point particles to interact with each other only through collisions. The path taken by the hard sphere molecule between successive collisions is termed as the free path. For such a system, as shown in Figure 2.1 when two molecules of diameter d collide, their centers are separated by a distance equal to d. In other words, if there is any other molecule within a distance d from the center of the target molecule we say that a collision occurs. The effective area πd^2 of this circle of radius d is termed as the collisional cross-section.

The mean free path can be estimated by considering a simplified system where a single particle is moving with a speed, $\bar{v}$ in a sea of stationary point particles with number density as n. As shown in Figure 2.2, in time t, this particle will sweep a volume of $\pi d^2 \bar{v} t$. Thus, $n\pi d^2 \bar{v} t$ gives the number of gas molecules inside this volume and is, therefore, the estimated number

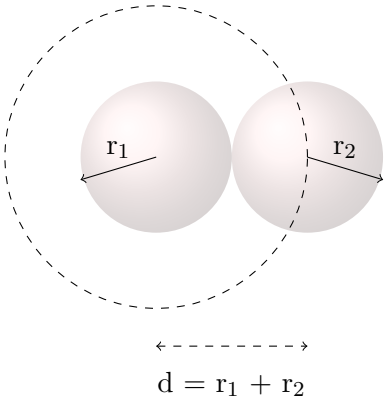

Figure 2.1: A representation of the collisional cross-section. r_1 and r_2 represent the radiuses of the individual molecules and d represents the collisonal cross-section.

of collisions in time t for the moving particle. Therefore, the Mean free path (λ) defined as the average distance that the particle travels between successive collisions can be found as the ratio of the length of the cylinder and the average number of collisions when particle sweeps the cylinder. In particular

$$\lambda = \frac{\bar{v}t}{n\,\pi d^2 \bar{v}t} = \frac{1}{n\,\pi d^2}. \tag{2.1}$$

We arrive at the above expression with the assumption that the target molecules are stationary. A more elaborate calculation, taking into account the average relative speeds of the particles, $\bar{v}_{rel} = \sqrt{2}\bar{v}$, leads to a more accurate estimate as follows,

$$\lambda = \frac{1}{\sqrt{2}\,n\,\pi d^2}. \tag{2.2}$$

As an example, for Air at STP conditions, using the ideal gas assumption for dilute gases, we have the number density,

$$n = \frac{p}{kT} = \frac{10^5 N/m^2}{1.38 \times 10^{-23} J/K \times 300K} = 2.5 \times 10^{25} \text{molecules}/m^3. \tag{2.3}$$

Using this, and the diameter of Air as the diameter of Nitrogen, d = 3.6A^o, we get the mean free path from Equation 2.1 as follows,

$$\lambda = \frac{1}{3.1415 \times 2.5 \times 10^{25} \times 21.6 \times 10^{-20}} \approx 60\text{nm}. \tag{2.4}$$

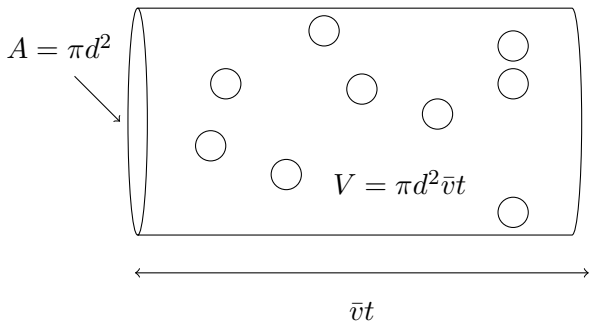

Figure 2.2: Classical picture of the collisional tube in a hard-sphere system. In time t, the average distance traveled by any particle is $\bar{v}\,t$, where $\bar{v}$ is the average speed

The time scale of interest for the dynamics of the distribution function is the Mean free time, (τ), which is the average time spent by the particle between successive collisions and is defined as,

$$\tau = \frac{\lambda}{\bar{v}} = \frac{1}{n\,\pi d^2}\sqrt{\frac{m}{2k_{\mathrm{B}}\,T}}, \tag{2.5}$$

where m is the mass of the particle, k_B is the Boltzmann's constant, T is the temperature of the gas and we have used the most probable speed, $\bar{v} = \sqrt{2k_BT/m}$. Thus, the mean free time for air can be found from Equation 2.5 as follows,

$$\tau = \frac{60\mathrm{nm}}{300\mathrm{m/s}} \approx 0.2\mathrm{ns}. \tag{2.6}$$

Fluid	d (in pm)	λ (in nm)	τ (in ns)
Air	400	60	0.2
Helium	260	115	0.38
Argon	340	67	0.22
Xenon	396	50	0.16

Table 2.1: Typical values of the mean free path and mean free time for various gases

Table 2.1 gives the values of the mean free path and time for various gases. We now define how these scales relate to the corresponding macroscopic ones. This will allow us to estimate the computational cost for molecular simulations.

As stated in the introduction, the non-dimensional parameter that compares the microscopic and the macroscopic length scales is the Knudsen number, defined as the ratio of the mean free path and the macroscopic length scale, L, i.e.,

$$\mathrm{Kn} = \frac{\lambda}{L}. \tag{2.7}$$

It helps in validating the continuum hypothesis and the fluid description of the Navier-Stokes-Fourier Equations. A high Knudsen number, ($\mathrm{Kn} > 0.01$) would mean that the molecular mean free path is comparable to the macroscopic length scale and would break down the assumption of treating fluids as a continuum. MEMS/NEMS typically fall in this category where the mean free path and the macroscopic length scale are comparable. Numerical modeling of such systems therefore requires schemes able to model beyond the Navier-Stokes equations.

The macroscopic transport quantities such as the viscosity (ν) and thermal conductivity (κ) arise as a consequence of molecular collisions exchanging momentum. The relation (Chapman & Cowling, 1990) between viscosity and the mean free path for a gas with elastic hard-sphere molecules is defined as,

$$\lambda = \frac{16}{5}\nu\left(\frac{m}{2\pi k_{\mathrm{B}}T}\right)^{\frac{1}{2}}. \tag{2.8}$$

The relationship between the Reynolds (Re), Mach (Ma) and Knudsen numbers found from the above relation is therefore

$$\mathrm{Kn} = \frac{16}{5}\sqrt{\frac{\gamma}{2\pi}}\frac{\mathrm{Ma}}{\mathrm{Re}}, \tag{2.9}$$

where, $\mathrm{Ma} = U_c/c_s$ is the ratio of the macroscopic velocity scale U_c and the sound speed $c_s = \sqrt{\gamma k_{\mathrm{B}}T/m}$, $\mathrm{Re} = U_cL/\nu$ is the ratio of the inertial to the viscous time scales and γ is the adiabatic gas constant.

In the incompressible limit, i.e, $\mathrm{Ma} < 0.3$, and for a high Reynolds number, ($\mathrm{Re} = \mathcal{O}(10^3)$),

we would have a Knudsen number of Kn $\approx 5 \times 10^{-4}$. This indicates that the mean free path is orders of magnitude smaller than the macroscopic length scale and as the number of computational cells scales as Kn^{-3}, therefore the representation of such scales on molecular simulations is a computationally expensive task.

2.2 Distribution function and its moments

In this section, we introduce the concept of the probability distribution function in the context of kinetic theory. For an N-particle system with number density n, one could sample all possible velocities of particles $n\ d\boldsymbol{x}$ within the differential volume $d\boldsymbol{x}$ creating a distribution of velocities. Such a distribution would depend on the position $\boldsymbol{x}$, time t and the location in the velocity space $\boldsymbol{v}$. Therefore, we could describe a distribution function of the form $f^{(N)}(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \cdots \boldsymbol{x}_N, \boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \cdots \boldsymbol{v}_N, t)$, which is known as the N-particle distribution function. For dilute gases, a much coarser description in terms of only the phase space coordinates of a single particle is sufficient. This single-particle distribution fucntion, f is the central quantity of interest in Boltzmann kinetic theory of dilute gases. Figure 2.3 shows the differential position and velocity elements in phase space of the single particle. It describes the probability of finding a molecule in the velocity range $\boldsymbol{v}$ to $\boldsymbol{v} + d\boldsymbol{v}$ and in the position range $\boldsymbol{x}$ to $\boldsymbol{x} + d\boldsymbol{x}$. With a prefactor to denote the presence of N particles, f satifies the normalization condition

$$\int d\boldsymbol{x} \int d\boldsymbol{v} f\left(\boldsymbol{x}, \boldsymbol{v}, t\right) = N, \tag{2.10}$$

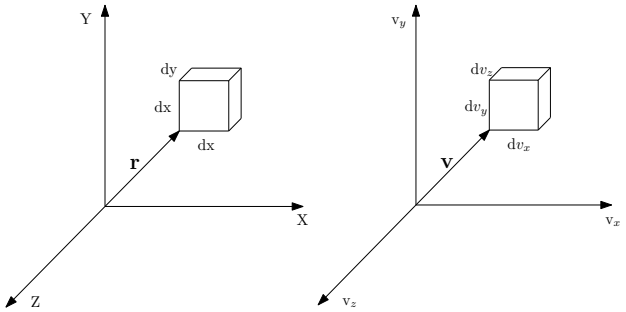

Figure 2.3: Left: A configuration space volume element $dr = dxdydz$ at spatial position r. Right: The equivalent velocity space element. Together they constitute a volume element $d\Omega = drdv$ at position $(\mathbf{r}, \mathbf{v})$ in phase space

The evolution of the distribution function could be related to the macroscopic fields through its moments. In D dimensions, hydrodynamic variables, the mass density ρ, the fluid velocity $\boldsymbol{u}$ and temperature T, are related to the distribution function as (Chapman & Cowling, 1990)

$$\rho = \int f m d\boldsymbol{v}, \quad \rho u_\alpha = \int m v_\alpha\, f\, d\boldsymbol{v}, \quad \frac{\rho}{2} u^2 + \frac{D \rho k_B T}{2m} = \int \frac{m v^2}{2}\, f\ d\boldsymbol{v}, \tag{2.11}$$

where k_B is the Boltzmann constant and m is the mass of the particle and is scaled with unity henceforth.. The thermodynamic pressure p is related to the scaled temperature $\theta = k_B T/m$ through the ideal gas equation of state $p = \rho\theta$.

Similar to Equation 2.11, we could define the average of any quantity $\phi(\boldsymbol{v})$ as,

$$\langle \phi(\boldsymbol{v}) \rangle = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} d\boldsymbol{v} \phi(\boldsymbol{v}) f(\boldsymbol{x}, \boldsymbol{v}, t). \tag{2.12}$$

Often, it is convenient to express the macroscopic moments in the co-moving reference frame

defined in terms of the peculiar velocity $\boldsymbol{\xi} = \boldsymbol{v} - \boldsymbol{u}$. In this coordinate system, the internal energy ρe therefore takes the form

$$\rho e \equiv \frac{D\rho k_B T}{2m} = \left\langle \frac{\xi^2}{2}, f \right\rangle. \tag{2.13}$$

Other quantities of interest in hydrodynamics is the stress tensor $\sigma_{\alpha\beta}$. It is defined as the traceless part of the momentum flux tensor in the co-moving frame. Similarly, the heat flux q_α is the energy flux in the comoving system. Thus

$$\sigma_{\alpha\beta} = \left\langle \overline{\xi_\alpha \xi_\beta}, f \right\rangle, \qquad q_\alpha = \left\langle \frac{\xi^2}{2}\xi_\alpha, f \right\rangle, \tag{2.14}$$

where the traceless form of any symmetrized second order tensor $A_{\alpha\beta}$ in terms of the kronecker delta $\delta_{\alpha\beta}$ is defined as,

$$\overline{A}_{\alpha\beta} = \frac{1}{2}\left(A_{\alpha\beta} + A_{\beta\alpha}\right) - \frac{1}{D}A_{\gamma\gamma}\delta_{\alpha\beta}. \tag{2.15}$$

Similarly, the traceless symmetrized third order tensor $Q_{\alpha\beta\gamma}$ $(= \langle \xi_\alpha \xi_\beta \xi_\gamma, f \rangle)$ is defined as,

$$\overline{Q}_{\alpha\beta\gamma} = Q_{\alpha\beta\gamma} - \frac{2}{D+2}(q_\alpha \delta_{\beta\gamma} + q_\beta \delta_{\alpha\gamma} + q_\gamma \delta_{\alpha\beta}), \tag{2.16}$$

which appears in the evolution equation for the stress tensor $\sigma_{\alpha\beta}$.

2.3 Boltzmann Equation

The time evolution of the single particle distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ under the action of external forces and internal collisions is governed by the Boltzmann equation. The Boltzmann equation provides a statistical description of dilute gases in the hydrodynamic and rarefied regimes even far away from equilibrium. This integro-differential equation is of the form (Cercignani, 1975)

$$\partial_t f + c_\alpha \partial_{x_\alpha} f + g_\alpha \partial_{c_\alpha} f = \Omega^{\mathrm{B}}(f, f), \tag{2.17}$$

where g_α is the acceleration due to external forces and $\Omega^{\mathrm{B}}(f, f)$ represents the collision kernel describing internal collisions. The left hand side describes the change in the distribution function resulting from the free flight of the molecules across the phase space differential volumes $d\boldsymbol{x}$ and $d\boldsymbol{c}$. The collision kernel represents binary collisions between particles and assumes that their velocities to be uncorrelated prior to the collision. This assumption is termed *"Stosszahlansatz"* or the Molecular chaos assumption (Cercignani, 1975).

$$f_{12}(\boldsymbol{x}_1, \boldsymbol{c}_1, \boldsymbol{x}_2, \boldsymbol{c}_2, t) = f(\boldsymbol{x}_1, \boldsymbol{c}_1, t) f(\boldsymbol{x}_2, \boldsymbol{c}_2, t). \tag{2.18}$$

One can define the Boltzmann collision operator Ω^{B}, a bi-linear function of f using,

$$\Omega^{\mathrm{B}}(f, f) = \int d\boldsymbol{c}' \int d\boldsymbol{c}_1 \int d\boldsymbol{c}_2 \left\{ w(\boldsymbol{c}, \boldsymbol{c}' | \boldsymbol{c}_1, \boldsymbol{c}_2) f(\boldsymbol{c}_1) f(\boldsymbol{c}_2) - w(\boldsymbol{c}_1, \boldsymbol{c}_2 | \boldsymbol{c}, \boldsymbol{c}') f(\boldsymbol{c}) f(\boldsymbol{c}') \right\}, \tag{2.19}$$

with $w(\boldsymbol{c}, \boldsymbol{c}' | \boldsymbol{c}_1, \boldsymbol{c}_2) \equiv w \geq 0$ as the probability that as a result of collision between molecules with velocities $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ acquires velocities $\boldsymbol{c}'$ and $\boldsymbol{c}$ according to the laws of elastic collisions:

$$\begin{aligned} \boldsymbol{c} + \boldsymbol{c}' &= \boldsymbol{c}_1 + \boldsymbol{c}_2, \\ \boldsymbol{c}^2 + \boldsymbol{c}'^2 &= \boldsymbol{c}_1{}^2 + \boldsymbol{c}_2{}^2. \end{aligned} \tag{2.20}$$

A few basic properties of the Boltzmann equation are as follows:

1. **Collisional invariants:** Binary elastic collisions between point particles conserves mass, momentum and energy. This is reflected by the condition:

$$\langle \Omega^{\mathrm{B}}, \{1, \boldsymbol{c}, \boldsymbol{c}^2\} \rangle = 0. \tag{2.21}$$

This property of the collision operator holds true for any form of the kernel function. Thus, changes in mass, momentum and energy is a result of particle redistribution in the position space. The Boltzmann collision term is therefore said to be local in position space.

2. **Conservation laws:** Taking the moments of Equation 2.17, one can arrive at the macroscopic equations governing the evolution of mass, momentum and energy.

$$\begin{aligned} \partial_t \rho + \partial_\alpha j_\alpha &= 0, \\ \partial_t \left(\rho u_\alpha\right) + \partial_\beta \left(\rho u_\alpha u_\beta + \rho \frac{k_B T_0}{m} \delta_{\alpha\beta} + \sigma_{\alpha\beta}\right) &= 0, \\ \partial_t \left(\frac{1}{2}\rho u^2 + \frac{D k_B T}{2m}\right) + \partial_\beta \left(\frac{1}{2}\rho u^2 u_\beta + \rho(D+2)\frac{k_B T_0}{2m} u_\beta + \sigma_{\alpha\beta} u_\beta + q_\beta\right) &= 0. \end{aligned} \tag{2.22}$$

One can see that these equations are obtained irrespective of the form of the collision operator as long as it conserves mass, momentum and energy. Thus, different collision forms of the collision term can be used to obtain the same set of conservation laws.

3. **Zero Point of the collision:** It would be important to understand the solution of the equation

$$\Omega^{\mathrm{B}}(f, f) = 0. \tag{2.23}$$

Such a region arises when the following condition (from Equation 2.19) is satisfied,

$$f(\boldsymbol{c}_1) f(\boldsymbol{c}_2) = f(\boldsymbol{c}) f(\boldsymbol{c}^{'}). \tag{2.24}$$

This is known as the detailed balance condition. It implies that for every pair of particles with pre-collisional velocities as $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ and post-collisional velocities as $\boldsymbol{c}$ and $\boldsymbol{c}^{'}$, there exists an inverse partner. Taking logarithms on both sides of Equation 2.24, one can show that $\ln f$ is an additive invariant. As a consequence, at thermodynamic equilibrium $\ln f$ must be a function of the collision invariants (Grad, 1949). This gives rise to the form:

$$\ln f = \lambda + \beta_\alpha c_\alpha + \gamma c^2, \tag{2.25}$$

where λ, β_α and γ are the lagrange multipliers for the mass, momentum and energy constraints repectively. The solution of Equation 2.25 represents the Maxwell-Boltzmann distribution given by,

$$f^{\mathrm{MB}}(\boldsymbol{c}) = \rho \left(\frac{m}{2\pi k_B T}\right)^{3/2} \exp\left(-\frac{m}{2k_B T}(\boldsymbol{c} - \boldsymbol{u})^2\right). \tag{2.26}$$

4. **Entropy Production:** Boltzmann defined a non-equilibrium generalization of the entropy, termed as the $\mathcal{H}$ function (Cercignani, 1975)

$$\mathcal{H} = \int d\boldsymbol{c}\ f \ln f, \tag{2.27}$$

where $S_B = -k_B\mathcal{H}[f]$ is the thermodynamic entropy at equilibrium. During the time evolution, the entropy increases locally and it is evident from the evolution equation for the $\mathcal{H}$ function

$$\partial_t\mathcal{H} + \partial_\alpha J_\alpha^H = -\sigma^{(\mathrm{B})} \leq 0, \tag{2.28}$$

where entropy flux term and positive definite entropy production terms are

$$J_\alpha^H = \int d\mathbf{c}\; c_\alpha(f \ln f), \quad \sigma^{(\mathrm{B})} = -\langle \Omega^{\mathrm{B}}, \ln f \rangle \geq 0. \tag{2.29}$$

Equation 2.29 shows that the production term results in an increase in the entropy with time. And zeros of the collision term satisfy this inequality, particularly the Maxwellian with the entropy production ($\sigma^{(\mathrm{B})}$) being zero at equilibrium. This is considered as a generalization of the second law of thermodynamics.

2.4 Length scales in Isotropic 3D Turbulence

In this section, we focus on comparing the scales at the molecular level (λ) and the turbulence scale. We focus on an idealized set of 3D homogenous turbulence problem. In this set-up all boundary influences are ignored and it is assumed that there is no mean flow and the statistics are spatially isotropic and homogeneous. The absense of mean flow and the fact that rotation and buoyancy are not important leads to a statistically isotropic flow. The flow is said to be homogeneous if there are no spatial gradients in any averaged quantity. Such idealization is simulated on computers via a periodic boundary conditions which mimics an infinite domain.

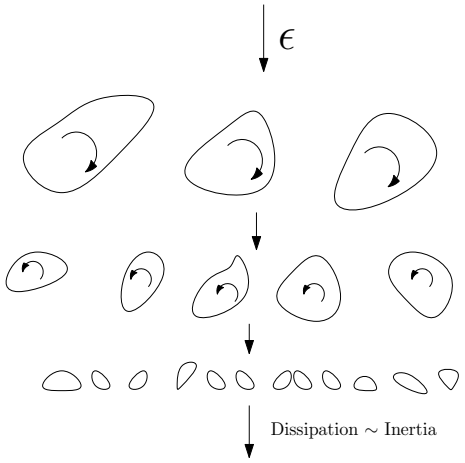

Figure 2.4: A schematic of the energy cascade process as hypothesized by Lewis Richardson

Lewis Richardson (Richardson, 1922) provided a first picture of the energy cascade that is observed in developing turbulent motions of fluid flow. His idea of the cascade involved the transfer of Kinetic energy at the macroscopically observable flow scales down to smaller scales through an inviscid mechanism. These smaller scales were later quantified by Kolmogorov in 1941 (AN, 1941).

The kinetic energy of large eddies of the size of the macroscopic length scale cascade into smaller eddies until they are dissipated by viscosity. Denoting the large scale velocity by $a = \sqrt{\langle \mathbf{u}_i\mathbf{u}_i \rangle}$, we define the rate of energy transport (ϵ) as,

$$\epsilon = \frac{a^2}{t_0} = \frac{a^3}{L}, \tag{2.30}$$

where $t_0 = L/a$ represents the time scale at the top of the hierarchy and L represents the macroscopic length scale. With the Kolmogorov theory (Pope, 2001) being valid for large Reynolds numbers, the viscous time scales are very large compared to the inertial and hence we obtain the kolmogorov scales when the two time scales are comparable. Mathematically,

$$\mathrm{Re}_\eta = \frac{u_\eta l_\eta}{\nu} \sim 1, \tag{2.31}$$

where, l_η, u_η and Re_η are the length, velocity and the Reynolds number respectively defined in the Kolmogorov scale. At this scale, viscous dissipation is balanced by energy transfer from larger scale. Thus,

$$\epsilon \approx \frac{\nu u_\eta^2}{l_\eta^2}. \tag{2.32}$$

Using this in Equation 2.31, we have the Kolomogorov length, velocity and time scales as

$$l_\eta = \left(\frac{\nu^3}{\epsilon}\right)^{1/4}, \qquad u_\eta = (\epsilon\nu)^{1/4}, \qquad \tau_\eta \equiv \frac{l_\eta}{u_\eta} = \left(\frac{\nu}{\epsilon}\right)^{1/2}. \tag{2.33}$$

2.4.1 Relation to microscopic length scale

We now estimate the relationship between the mean free path and the kolmogorov length scale. Their relation would give us an idea about the computational expense in performing a direct numerical simulation (DNS) of turbulence using DSMC. In using particle methods like DSMC for turbulent flow simulations, where computational expense scales as Kn^{-3}, an estimate of the l_η/λ can tell how well resolved a simulation is.

From the relation between the mean free path and the viscosity for a dilute Hard-Sphere gas (Equation 2.8), we can now relate the microscopic and the Kolmogorov length scales. In particular, we have

$$\frac{l_\eta}{\lambda} = \frac{1}{\lambda}\left(\frac{\nu^3}{\epsilon}\right)^{1/4} = \frac{k_1\, c_s}{\nu}\left(\frac{\nu^3}{\epsilon}\right)^{1/4} = \frac{k_1}{\mathrm{Ma}_\lambda}, \tag{2.34}$$

where $k_1 (= \frac{5}{16}\sqrt{2\pi/\gamma})$ is the prefactor in the relation in Equation 2.8. Ma_λ is the Mach number based on the Kolmogorov velocity scale (u_η from Equation 2.33). Similarly, the large and small scale Mach number can therefore be related in terms of large scale Reynolds number $\mathrm{Re}_a = La/\nu$ as

$$\mathrm{Ma}_a = \frac{a}{c_s} = \frac{(\epsilon L)^{1/3}}{c_s} = \frac{(\epsilon\nu)^{1/3}\,(L/\nu)^{1/3}}{c_s} = (\mathrm{Ma}_\lambda)^{4/3}\,(\mathrm{Re}_a)^{1/3}\left(\frac{1}{\mathrm{Ma}_a}\right)^{1/3}, \tag{2.35}$$

which allows us to write:

$$\mathrm{Ma}_\lambda = \frac{\mathrm{Ma}_a}{(\mathrm{Re}_a)^{1/4}}. \tag{2.36}$$

Therefore, we finally have

$$\frac{l_\eta}{\lambda} = \frac{k_1\,(\mathrm{Re}_a)^{1/4}}{\mathrm{Ma}_a}. \tag{2.37}$$

Rewriting this equation in terms of the Knudsen number,

$$\frac{l_\eta}{\lambda} = \frac{k_1 k_2}{\mathrm{Ma}_a^{3/4}\mathrm{Kn}}, \tag{2.38}$$

where k_2 is the prefactor obtained from Equation 2.9. We see, as the macro scale Mach number (Ma_a) increases, for turbulent flows, the mean free path and the Kolmogorov length

scale become comparable, therefore increasing the cost for a molecular simulation as it requires a higher cell resolution. Beyond a certain threshold, this linear increase saturates at a finite value with increasing Mach number (Pradhan & Kumaran, 2016).

Chapter 3
Direct Simulation Monte Carlo

In this chapter, we briefly discuss the microscopic method used in our coupled solver. Unlike Molecular Dynamics (MD) (Alder & Wainwright, 1959), where particle collisions are deterministic and provides an exact representation of particle trajectories. Particle based methods like Direct Simulation Monte Carlo (DSMC) provides a statistical description of fluid flow. In force-based Molecular dynamics, one only needs to specify the forces of interaction $F_{ij}(r_i, r_j, v_i, v_j)$ between any two particles apart from their phase space coordinates. The time-intensive nature of this force calculation in MD restricts its applicability to short time microscale simulations.

A less time consuming method is Event-driven MD (EDMD) (Alder & Wainwright, 1959; Lubachevsky, 1991) where the time consuming force-field calculations of force-based MD were replaced by elastic or inelastic collision rules for particles. Here, since the collisions take an infinitesimal duration, the need for computing the interactions and therefore the particle trajectories is done away by using a collision law characterized by the coefficients of restitution. Therefore, these simulations are much more efficient than force-based MD. But, the applicability of such methods is restricted to systems where particle streaming takes a much larger time against particle collisions.

For dilute gases, molecular description is provided by the Boltzmann equation. In this regard, Direct simulation Monte carlo has emerged as the most important tool for the direct simulation of the Boltzmann equation (Bird, 1970; Bird & Brady, 1994). Just like EDMD, one is not interested in computing the real particle trajectories and therefore provides a decoupled advection-collision representation in DSMC. This method divides the physical domain into a number of computational cells with the real gas represented by an N_c particle system (see Fig(3.1)) in each of these cells. As it is expensive to represent a physical scenario with trillions of particles, each particle in the domain represents an effective number of particles, N_e. The time evolution consists of local collision in the cell followed by free flight in which particles might cross over to neighboring cells. The differences between DSMC and MD methods are primarily associated with the procedures for handling collisions. Unlike MD, in DSMC the collisions are localized in the cell and statistical in nature. The requirement of a large number of particles in the domain for an effective transport of hydrodynamic quantities is replaced by the presence of effective number of particles, N_e in the collision sub-routine.

As an illustration, if we consider air for which $\lambda = 60$nm at $T_0 = 20^{\circ}C$, the sound speed $c_s \approx 343$ m/s. For Re $= 2000$ and Ma $= 0.2$ (giving us the macroscopic velocity scale to be $\approx 69\ m/s$) setup, in the relation for the Knudsen number (Eqn.(2.7)), we get the macroscopic length scale as $L_1 \approx 0.1868$mm. Therefore, the simulation for the given parameters with the working fluid being Air corresponds to a length scale in the millimetre scale. The number density of air at STP was found via Eqn(2.3). This when used to find the number of particles in a cube of size 2λ, we have,

$$N_0 = n \times (\Delta x)^3 = 2.5 \times 10^{25} \times (120 \times 10^{-9})^3 = 43200. \tag{3.1}$$

Therefore, the number of molecules at various cell sizes at STP for air can be calculated and is shown in Table(3.1). This gives us an estimate for N_e to be used through the ratio of the number of physical particles to the number of simulated particles.

For convenience, here onwards all length-scales are scaled with respect to the molecular diameter (d).

$$\tilde{L} = \frac{L}{d}, \qquad \tilde{\lambda} = \frac{\lambda}{d}, \qquad \tilde{n} = n\, d^3, \tag{3.2}$$

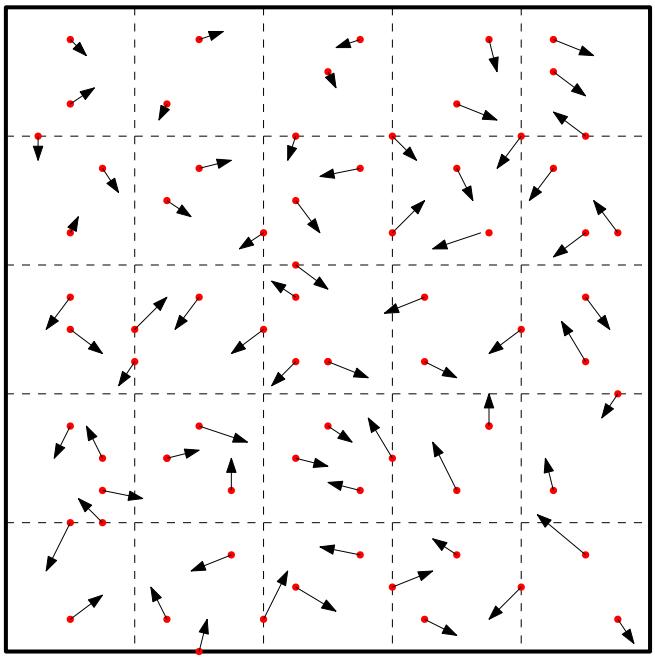

Figure 3.1: A 2D projection of a typical DSMC set-up. Dashed lines represent the cell boundaries

$\Delta x = 0.5\lambda$	N_0 (number of molecules)
2λ	43200
1λ	5400
0.5λ	675

Table 3.1: Typical values of the number molecules in a cubical volume of varying sizes at atmospheric pressures

which implies

$$\tilde{\lambda} = \frac{1}{\sqrt{2}\,\pi\,\tilde{n}}. \tag{3.3}$$

Here $\tilde{L}$, $\tilde{\lambda}$ and $\tilde{n}$ are the nondimensional length, mean free path and number density respectively. For air, with $d = 3.6A^o$ and $p = 231.48$Pa, we have $\tilde{\lambda} \approx 16.67$. For a cell size of $\Delta x = 2\tilde{\lambda}$ and one which has 100 particles in it, we have the nondimensional number density as,

$$\tilde{n} = \frac{100}{(2\tilde{\lambda})^3} = 2.7 \times 10^{-3}. \tag{3.4}$$

Typically, time scales are of the order of a few mean free time. Therefore, time steps in DSMC are taken to be a fraction of the mean free time $\Delta t = c\tau$, with $c \leq 1$.

Representing a full-domain molecular simulation for length scales of most modern applications is computationally very expensive and therefore such solvers usually find extensive use in microscale devices(Piekos & Breuer, 1995). We hence rely on DSMC simulations to represent physical systems of the order of the mean free path (λ) or a few orders of magnitude higher.

3.1 DSMC Methodology

In this section, we describe the implementation of the DSMC solver. As shown in Fig(3.1), the DSMC domain is discretized into a set of spatial cells in all three dimensions. Each cell is uniformly initialized with a certain number of particles, N_0. Collisions are computed in these

cells that range in length of the order of the mean free path. A flowchart representing the DSMC Algorithm is as shown in Fig(3.2). The details of individual steps are as follows:

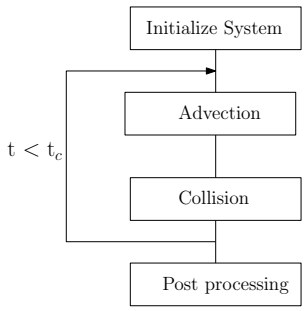

Figure 3.2: A flowchart showing the DSMC algorithm. Here, t_c represents the total time of the simulation.

- **Initialization**: The initial particle velocities are sampled from the Maxwell-Boltzmann distribution with the standard deviation of the sampled distribution being the temperature T_0 of the system and a required mean velocity **u**.
- **Advection**: Particles are advected as ghosts (see Fig(3.3)) with their motion in space described by the equations of motion for position and velocity updates. The positions ($\boldsymbol{x}$) and the velocities ($\boldsymbol{v}$) of the particles at time t are evolved as follows,

$$\boldsymbol{x}(\mathrm{t}+\Delta\mathrm{t}) = \boldsymbol{x}(\mathrm{t}) + \boldsymbol{v}\Delta\mathrm{t} + \frac{1}{2}\boldsymbol{g}\Delta\mathrm{t}^2, \tag{3.5}$$

$$\boldsymbol{v}(\mathrm{t}+\Delta\mathrm{t}) = \boldsymbol{v}(\mathrm{t}) + \boldsymbol{g}\Delta\mathrm{t}, \tag{3.6}$$

where $\boldsymbol{g}$ is the applied acceleration on the system.

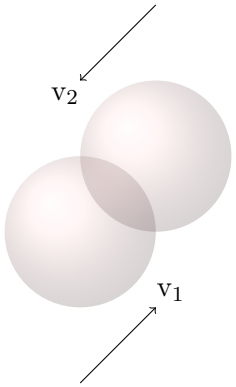

Figure 3.3: A schematic showing particles treated as ghosts during advection

Ocassionally, particles near walls collide with them. For particle collisions with the walls, we calculate the time for collision with the wall (t_{wall}) for any particle whose location and velocity are given by $\boldsymbol{x}$ and $\boldsymbol{v}$ as follows,

$$t_{wall} = \frac{(\boldsymbol{x}_{wall} - \boldsymbol{x}).\hat{\boldsymbol{n}}}{\boldsymbol{v}.\hat{\boldsymbol{n}}}, \tag{3.7}$$

where $\boldsymbol{x}_{wall}$ and $\hat{\boldsymbol{n}}$ are the position and normal vector of the wall plane respectively. Particles crossing cell boundaries are responsible for the transport of wall effects and other hydrodynamic quantities across the domain.

- **Boundary Conditions**: For our problem of interest and for an introduction to the method, we talk about a flow in a domain that is constricted by straight thermal walls with a fluid accelerating between them. The other two directions are periodic.

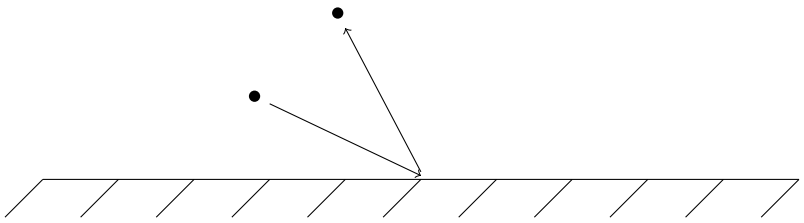

Figure 3.4: A schematic showing the diffusive nature of the wall

As an illustration, we consider particle collisions with a straight wall. We use diffuse boundaries for the top and bottom walls of a channel which assumes that the particle collides and equilibriates with the wall temperature and velocity and therefore has a wall equilibrium distribution. The boundary condition is implemented as follows,

$$v_x = \sqrt{\frac{k_\mathrm{B}T_w}{m}} r_G, \quad v_y = \sqrt{\frac{k_\mathrm{B}T_w}{m}} r_G^{'}, \quad v_z = \sqrt{-\frac{2k_\mathrm{B}T_w}{m}\log_e r}, \tag{3.8}$$

where T_w is the wall temperature, r_G and $r_G^{'}$ are gaussian distributed random numbers and r is a uniform random number between (0,1] generated from the Molecular Dice Algorithm(Agrawal *et al.*, 2018). For an isothermal simulation, we take the temperature of the bulk and the walls to be the same. The generated velocities represents the Maxwell-Boltzmann distributions with the mean as the wall velocity (zero for a stationary wall) and the standard deviation as the wall temperature (T_w) and is given as,

$$P_\perp(v_\perp) = \frac{m}{k_\mathrm{B}T_w} v_\perp e^{-mv_\perp^2/2k_\mathrm{B}T_w}, \quad P_\parallel(v_\parallel) = \sqrt{\frac{m}{2\pi k_\mathrm{B}T_w}} e^{-mv_\parallel^2/2k_\mathrm{B}T_w}, \tag{3.9}$$

where, $v_\parallel$ and $v_\perp$ represent the velocity components parallel and perpendicular to the wall respectively.

3.1.1 Collision process in DSMC

All collisions in DSMC are stochastic and are within a spatial cell of dimensions of the order of a mean free path. In order to calculate the number of particle collisions within a cell, one needs to determine the probability of a pair of particles to collide. For a pair of Hard-Sphere molecules, the probability that they would collide depends on their relative velocities, and therefore the collective probability of the set of particles is,

$$\boldsymbol{P}_\mathrm{coll}(i,j) = \frac{|\boldsymbol{v}_i - \boldsymbol{v}_j|}{\sum_{m=1}^{\mathrm{N_c}} \sum_{n=1}^{m-1} |\boldsymbol{v}_m - \boldsymbol{v}_n|}. \tag{3.10}$$

Calculating this distribution every time step is expensive because of the summation spanning over all ${}^{N}C_2$ collision pairs in a cell. Therefore, we use a criteria for the selection of a pair of particles for collision as (Alexander & Garcia, 1997)

$$\frac{|\boldsymbol{v}_i - \boldsymbol{v}_j|}{v_{r,\max}} > r, \tag{3.11}$$

where $v_{r,\max}$ is the expected maximum relative speed for the set of particles and r is a uniform random number generated between (0,1]. The maximum relative speed depends on the

temperature, T_0 of the gas. We let $v_{r,\max}$ to be a moderately high value ($\approx 30\sqrt{k_B T_0/M}$) in comparison to the particle mean velocities. This overestimation of $v_{r,\max}$ would lead to a higher number of rejections for every few pairs of particles selected for sampling. A smaller guess value would result in a higher acceptance and could lead to an ineffective representation of the collision process. To decide on the number of collisions within a cell, we take the number of collisions that could take place within a collision tube around a particle, i.e., a cylindrical volume of length λ and diameter of the order of the particle diameter. The number of candidates for collisions in this tube (of volume $\pi d^2 v_{r,\max}\Delta t$) of the total ${}^{N}C_2$ possibilities (in total volume V_c) would therefore be the fraction of total collisions in the collision tube. This is given as follows,

$$M_{\text{coll}} = \frac{N_0^2 \pi d^2 v_{r,\max} N_e \Delta t}{2V_{\text{c}}}, \tag{3.12}$$

where V_{c} is the volume of the cell. If a pair of particles passes our selection criteria, we can compute post-collisional velocities in terms of post collisional centre of mass and relative velocities as

$$\boldsymbol{v}_i^{'} = \boldsymbol{v}_{cm} + \frac{1}{2}\boldsymbol{v}_r^{'}, \qquad \boldsymbol{v}_j^{'} = \boldsymbol{v}_{cm} - \frac{1}{2}\boldsymbol{v}_r^{'}. \tag{3.13}$$

The centre of mass velocity and the relative speeds of the pair of particles remains unchanged because of momentum and energy conservation respectively. Only the directions that the particles traverse post-collision is altered. So, we may write

$$\boldsymbol{v}_{\text{cm}} = \frac{1}{2}(\boldsymbol{v}_i + \boldsymbol{v}_j), \quad \boldsymbol{v}_r^{'} = v_r[(\sin\theta\cos\phi)\hat{e_1} + (\sin\theta\sin\phi)\hat{e_1} + \cos\theta\hat{e_3}]. \tag{3.14}$$

The direction is decided based on the distribution of the particle velocities on a sphere of radius v_r and centred on $\boldsymbol{v}_{\text{cm}}$. Here, one assumes that the azimuthal angle ϕ is uniformly distributed on a spherical surface

$$P(\phi)d\phi = \frac{1}{2\pi}d\phi, \tag{3.15}$$

and therefore, $\phi = 2\pi r_2$, where r_2 is a uniform random number between (0,1).

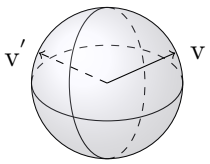

Figure 3.5: A schematic representing the collision process wherein the radius representing the relative velocity magnitude remains unchanged and the velocities randomly flip to another direction

However, there is a uniform probability that a particle is on a patch of area $dS = v_r^2 \sin\theta d\theta d\phi$ on the spherical surface. This gives on integrating out the dependence on the azimuthal angle,

$$P(\theta)d\theta = \frac{\sin\theta}{2}d\theta. \tag{3.16}$$

Therefore,

$$P(\cos\theta)d(\cos\theta) = \frac{1}{2}d(\cos\theta). \tag{3.17}$$

Hence, $\cos\theta$ is uniformly distributed on the sphere. Giving us,

$$\cos\theta = r_1, \qquad \sin\theta = \sqrt{1 - r_1^2}, \tag{3.18}$$

where r_1 is a uniform random number between [-1,1].

This process gives us the directions of the new relative velocity vectors using which one can obtain the post-collisional velocities of the particles through Equation 3.13. This stochastic procedure represents the collision process in DSMC and is repeated M_{coll} number of times for every time step.

3.2 Direct Simulation of the Boltzmann equation *

In this section, we describe how the simulation of real gas flows using DSMC also provides a solution to the equations governing the flow. We show how the particle advection and local collision provide a discrete description of the Boltzmann equation (Bird, 1970). In order to do so, we recognize how a multiple ensemble average of the DSMC particle density represents the single particle probability distribution function $f(\boldsymbol{x}, \boldsymbol{v}, t)$ and therefore the idea of the method is to determine the time-dependent distribution through a quasi-particle simulation (Pöschel & Schwager, 2005). Unlike, mesoscopic methods like Lattice-Boltzmann which deal directly with the distribution function and their evolution through a discretized Boltzmann equation, one derives the Boltzmann equation through the dynamics of the particles in DSMC.

To quantify the effects of the free-streaming and collision in DSMC, it is essential to understand how they translate to the change in the single particle distribution function, i.e. Δf. The change in the number of particles $f(\boldsymbol{x}, \boldsymbol{v}, t)d\boldsymbol{x}d\boldsymbol{v}$ in the phase space volume $d\boldsymbol{x}d\boldsymbol{v}$ in the absence of external forces would occur due to the following factors:

1. Particles with pre-collisional velocities $\boldsymbol{v} \pm d\boldsymbol{v}$ leave their velocity space volume acquiring a post-collisional velocity $\boldsymbol{v}' \neq \boldsymbol{v}$, thus reducing the number of particles in this velocity space volume. Particle collisions with velocities within $d\boldsymbol{v}$ does not result in a flux through $d\boldsymbol{v}$.

2. Particles with pre-collisional velocities $\boldsymbol{v}''$ enter the velocity space $\boldsymbol{v} \pm d\boldsymbol{v}$ post collision. This process increases the number of particles in $d\boldsymbol{v}$.

3. Free-streaming of particles through the faces of $d\boldsymbol{x}$ changes the number of particles in the position space volume around $\boldsymbol{x}$.

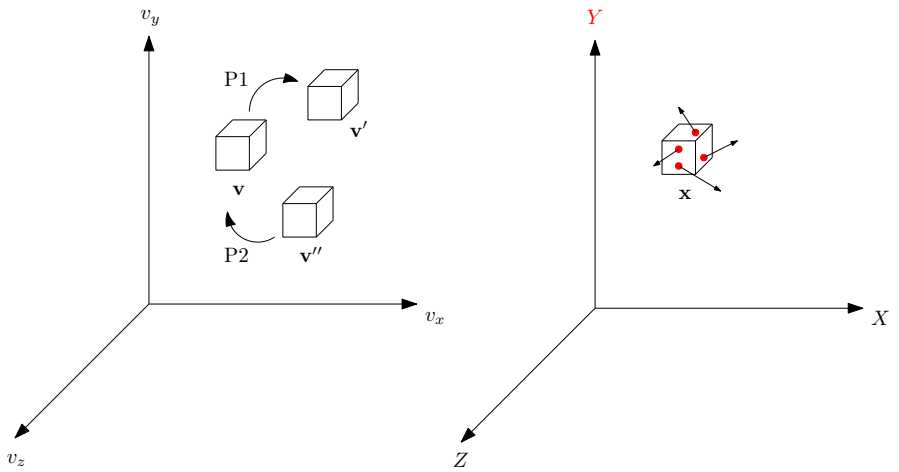

Figure 3.6: A schematic showing the various factors causing a change in the number of particles $f(\boldsymbol{x}, \boldsymbol{v}, t)d\boldsymbol{x}d\boldsymbol{v}$ in the 6-dimensional phase space. (a) Particles leaving the velocity space volume $d\boldsymbol{v}$ around $\boldsymbol{v}$ is shown as P1 (point 1 above) and particles entering this volume as P2 (point 2). (b) Particles leaving the position space volume $d\boldsymbol{x}$ around $\boldsymbol{x}$ through advection.

*This section is an adapted version from Pöschel & Schwager (2005)

To calculate the number of such changes through the phase space volume $d\boldsymbol{x}d\boldsymbol{v}$, we define the collision frequency $\nu^-(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{e})$ for point 1 which describe the number of collisions per time step Δt between particles with velocities $\boldsymbol{v}_1$ and $\boldsymbol{v}_2(\neq \boldsymbol{v}_1)$ along the unit vector $\boldsymbol{e} = (\boldsymbol{x}_2 - \boldsymbol{x}_1)/|\boldsymbol{x}_2 - \boldsymbol{x}_1|$. Similarly $\nu^+(\boldsymbol{v}_1'', \boldsymbol{v}_2'', \boldsymbol{e})$ is the collision frequency for point 2 between particles with velocities $\boldsymbol{v}_1''$ and $\boldsymbol{v}_2''$ that result in one of the particles having a post-collisional velocity of $\boldsymbol{v}_1$.

The changes in the probability distribution arising due to the negative flux (Point 1) can then be written as an integral over all possibilities of the colliding particle $\boldsymbol{v}_2$ and along all possible orientations in the position space $\boldsymbol{e}$.

$$\Delta f^-(\boldsymbol{v}_1)d\boldsymbol{v}_1 = -d\boldsymbol{v}_1 \int \nu^-(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{e})d\boldsymbol{v}_2 d\boldsymbol{e}\Delta t, \tag{3.19}$$

where the negative sign indicates the decrease in the number of particles. Similarly, the positive flux into the velocity space volume $d\boldsymbol{v}$ is given by integrating over all possibilities of $\boldsymbol{v}_1''$, $\boldsymbol{v}_2''$ and orientation $\boldsymbol{e}$.

$$\Delta f^+(\boldsymbol{v}_1)d\boldsymbol{v}_1 = \int \nu^+(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{e})d\boldsymbol{v}_1'' d\boldsymbol{v}_2'' d\boldsymbol{e}\Delta t. \tag{3.20}$$

Therefore. the total change in the distribution function is given by,

$$\Delta f(\boldsymbol{v}_1)d\boldsymbol{v}_1 = -d\boldsymbol{v}_1 \int \nu^-(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{e})d\boldsymbol{v}_2 d\boldsymbol{e}\Delta t + \int \nu^+(\boldsymbol{v}_1'', \boldsymbol{v}_2'', \boldsymbol{e})d\boldsymbol{v}_1'' d\boldsymbol{v}_2'' d\boldsymbol{e}\Delta t. \tag{3.21}$$

We can now derive a relation for the collision frequencies from a geometrical picture of the collision between particles with velocities $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$.

3.2.1 Collision frequency

Here, we revisit the relation between the collision frequencies and the single particle distribution function. We would obtain a bilinear form of the collision operator, similar in form to Equation 2.19.

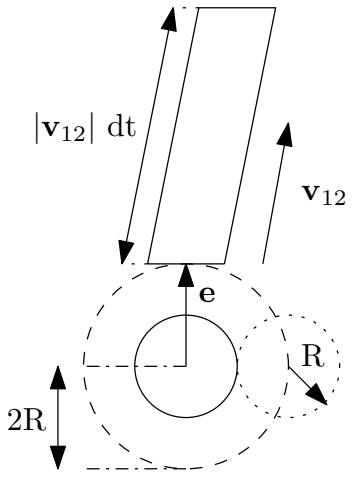

Figure 3.7: A 2D projection of the collision cross-section of radius $2R$ and the corresponding collision tube of height $|\boldsymbol{v}_{12}|dt$ oriented along $\boldsymbol{v}_{12} = \boldsymbol{v}_1 - \boldsymbol{v}_2$ and with base area $(2R)^2 d\boldsymbol{e}$. The dotted circle represents the projection of the colliding particle with velocity $\boldsymbol{v}_1$.

Figure 3.7 shows the collision tube from the frame of reference of particle 2. A collisional cross-section with radius $2R$ (equal to the distance between two particle centres) and a collision

tube of height $|\boldsymbol{v}_{12}|dt$ with the base area oriented along $\boldsymbol{e}$. The volume of this tube is given by

$$dV = (2R)^2 d\boldsymbol{e}|\boldsymbol{v}_{12} \cdot \boldsymbol{e}|dt. \tag{3.22}$$

The number of particles with velocity $\boldsymbol{v}_2$ is $f(\boldsymbol{v}_2)d\boldsymbol{v}_2$. The number of particles with velocity $\boldsymbol{v}_1$ in the collision tube is $H(\boldsymbol{v}_{12} \cdot \boldsymbol{e})f(\boldsymbol{v}_1)d\boldsymbol{v}_1 dV$. Here, $H(\boldsymbol{v}_{12} \cdot \boldsymbol{e})$ is the Heaviside function that makes the distinction betweem approaching ($\boldsymbol{v}_{12} \cdot \boldsymbol{e} < 0$) and departing ($\boldsymbol{v}_{12} \cdot \boldsymbol{e} > 0$) particles. We can find the number of collisions between $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ as,

$$\nu^-(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{e})d\boldsymbol{v}_1 d\boldsymbol{v}_2 d\boldsymbol{e}dt = dV H(\boldsymbol{v}_{12} \cdot \boldsymbol{e})f(\boldsymbol{v}_1)f(\boldsymbol{v}_2)d\boldsymbol{v}_1 d\boldsymbol{v}_2. \tag{3.23}$$

Simplifying the expression further, we arrive at,

$$\nu^-(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{e})d\boldsymbol{v}_1 d\boldsymbol{v}_2 d\boldsymbol{e} = f(\boldsymbol{v}_1)f(\boldsymbol{v}_2)(2R)^2|\boldsymbol{v}_{12} \cdot \boldsymbol{e}|H(\boldsymbol{v}_{12} \cdot \boldsymbol{e})d\boldsymbol{v}_1 d\boldsymbol{v}_2 d\boldsymbol{e}. \tag{3.24}$$

Similarly, the expression for the collision frequency contributing to the positive flux of the distribution function is as follows,

$$\nu^+(\boldsymbol{v}_1'', \boldsymbol{v}_2'', \boldsymbol{e})d\boldsymbol{v}_1'' d\boldsymbol{v}_2'' d\boldsymbol{e} = f(\boldsymbol{v}_1'')f(\boldsymbol{v}_2'')(2R)^2|\boldsymbol{v}_{12}'' \cdot \boldsymbol{e}|H(\boldsymbol{v}_{12}'' \cdot \boldsymbol{e})d\boldsymbol{v}_1'' d\boldsymbol{v}_2'' d\boldsymbol{e}. \tag{3.25}$$

The coefficient of restitution can be defined as follows,

$$\frac{|\boldsymbol{v}_{12}'' \cdot \boldsymbol{e}|}{|\boldsymbol{v}_{12} \cdot \boldsymbol{e}|} = \frac{1}{\varepsilon}. \tag{3.26}$$

For perfectly elastic collisions, the coefficient of restitution, $\varepsilon = 1$. Using the property of the Heaviside function, $H(\boldsymbol{v}_{12}'' \cdot \boldsymbol{e}) = H(-\boldsymbol{v}_{12} \cdot \boldsymbol{e})$ and with the change of variable $\boldsymbol{e} \to -\boldsymbol{e}$ in Equation 3.25, we have,

$$\nu^+(\boldsymbol{v}_1'', \boldsymbol{v}_2'', \boldsymbol{e})d\boldsymbol{v}_1'' d\boldsymbol{v}_2'' d\boldsymbol{e} = -f(\boldsymbol{v}_1'')f(\boldsymbol{v}_2'')(2R)^2|\boldsymbol{v}_{12} \cdot \boldsymbol{e}|H(\boldsymbol{v}_{12} \cdot \boldsymbol{e})d\boldsymbol{v}_1 d\boldsymbol{v}_2 d\boldsymbol{e}. \tag{3.27}$$

Using Equations 3.24 and 3.27 in Equation 3.21, we get the collision integral of the form,

$$I = (2R)^2 \int d\boldsymbol{v}_2 \int d\boldsymbol{e} H(\boldsymbol{v}_{12} \cdot \boldsymbol{e})|\boldsymbol{v}_{12} \cdot \boldsymbol{e}| \left[f(\boldsymbol{v}_1'')f(\boldsymbol{v}_2'') - f(\boldsymbol{v}_1)f(\boldsymbol{v}_2)\right]. \tag{3.28}$$

This integral is similar in form to the one shown in Equation 2.19. In the limit $\Delta t \to 0$, we get the Boltzmann equation of the form,

$$\frac{\partial}{\partial t} f(\boldsymbol{v}_1, t) = I. \tag{3.29}$$

We had neglected the free-streaming of particles in our derivation so far (Point 3). This corresponds to the change in the distribution arising due to its flux across the position space volume $d\boldsymbol{x}$. Accounting for this contribution as the convective term in the Boltzmann equation, we have,

$$\left(\frac{\partial}{\partial t} + \boldsymbol{v}_1 \cdot \nabla\right) f(\boldsymbol{v}_1, t) = I. \tag{3.30}$$

This shows how the local collision and free-streaming process of the DSMC algorithm leads to a direct simulation of the Boltzmann equation and would therefore recover appropriate macroscopic conservation equations as in Equation 2.22.

3.3 Modified Transport Parameters

As the collision process is stochastic in nature, the cell width Δx is maintained of the order of the Mean free path (λ) to ensure the correctness of transport parameters. Ideally, a well resolved DSMC simulation should work with $\Delta x \ll \lambda$ and $\Delta t \ll \tau$. Indeed, it is observed that DSMC requires that the cell sizes be below 0.5λ for an effective transport of hydrodynamic quantities (Bird & Brady, 1994). However, for engineering set-ups it is often not feasible to perform such resolved simulations. Thus, one would like to work with coarser cells and larger time-steps. However, using larger cells is equivalent to reducing the number of collisions that would happen in a smaller sized cell, and slowing the moment diffusion process. This is equivalent to a system having a reduced viscosity. Similarly, taking larger time steps results in lesser number of collisions and results in modified transport coefficients. Thus, one would expect that using a larger Δt and Δx should change the transport parameters to account for these reduced number of collisions.

Investigations (Alexander *et al.*, 1998; Garcia & Wagner, 2000) on the issue of working with larger Δx and Δt both as direct simulation based experiments and theoretically, they have found that the net effective correction added for a simulation running at a higher cell spacing and one that uses a larger time step jump is

$$\begin{aligned} \tilde{\nu} &= \nu_0 \left(1 + \frac{32}{135\pi}\frac{\Delta x^2}{\lambda^2}\right)\left(1 + \frac{32}{150\pi}\frac{\Delta t^2}{\tau^2}\right), \\ \tilde{\kappa} &= \kappa_0 \left(1 + \frac{32}{225\pi}\frac{\Delta x^2}{\lambda^2}\right)\left(1 + \frac{64}{675\pi}\frac{\Delta t^2}{\tau^2}\right). \end{aligned} \tag{3.31}$$

A simulation that was run with the above set of modified parameters in viscosity is as shown in Fig. 3.8. It is evident from this plot that once transport coefficient are properly re-parameterized the coarse simulations show as good result as that obtained from a finely resolved DSMC simulations.

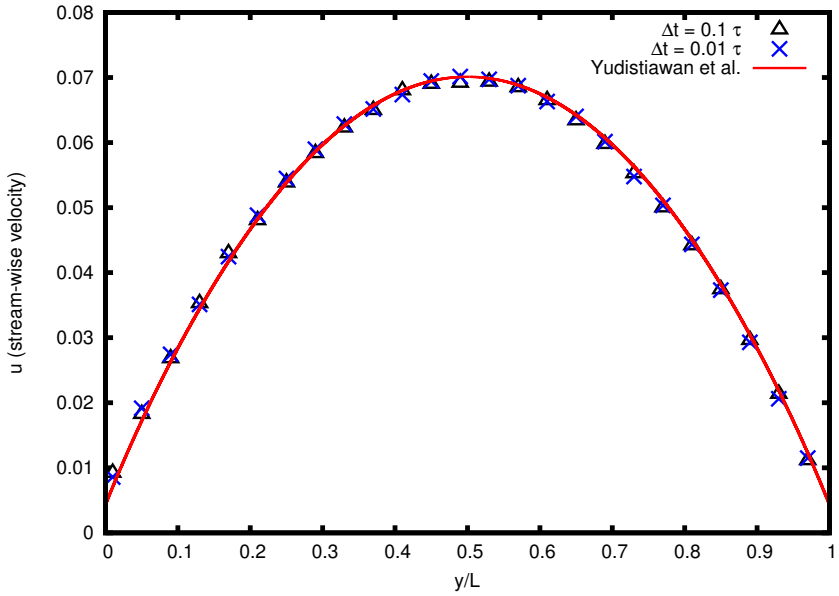

Figure 3.8: A channel flow simulation steady state velocity profile obtained using the modified parameters compared against (Yudistiawan *et al.*, 2010)

Similarly, we compare the steady state shear stress (σ_{xy}) profiles in Fig.(3.9) using a Δt of 0.1, 0.8 and 1 τ respectively. Once again, we observe that the higher order moments are also in good quantitative agreement after proper re-parametrization.

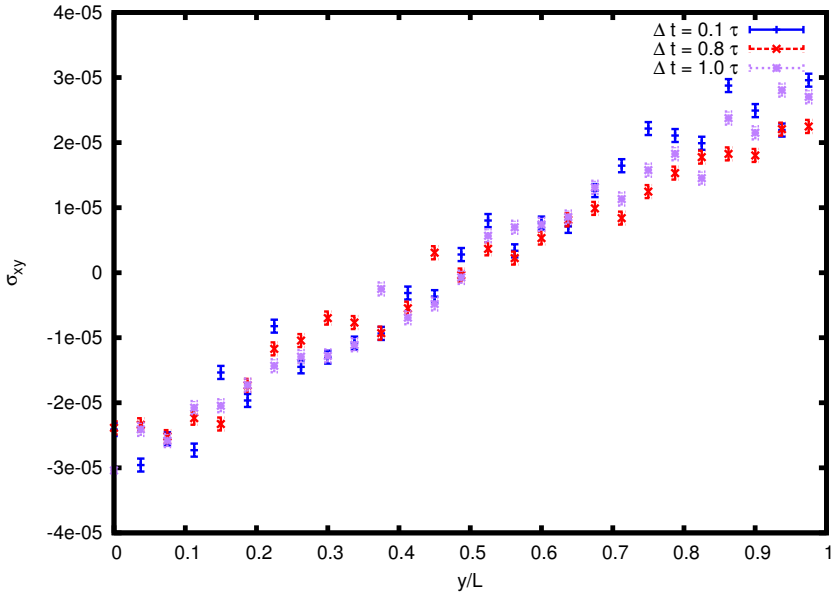

Figure 3.9: Steady state shear stress profile for an Re = 13, Ma = 0.2 simulation compared using various Δt after applying the viscous correction from Equation 3.31

We can understand the significance of increasing the time stepping by analyzing the change in the total time requirement per simulation. We can then estimate if such a maneuver could result in gains in simulation time. For example, high Reynolds number simulations requiring finer cell sizes for particle simulations (due to low Kn) would need a large number of particles in the domain. With limited computing resources, such simulations can become extremely time consuming. It is hence useful to be able to use larger time steps, to be able to attain a steady state sooner. Larger time steps causes a rise in the number of collisions to be processed per cell (as the number of collisions $\propto \Delta t$).

For our purposes, the time step maneuver could still be considered useful if the cost increase due to collisions and communications is not linearly dependent on $\Delta t_{new}/\Delta t_{old}$. That is, a factor 10 increase in Δt results only in a < 10 increase in the total time taken for processing all the routines per DSMC time step. This is possible as the time taken for the advection routine remains the same irrespective of Δt and the additional cost of collisions and communication rises slower than the ratio of time steps.

As a test case, we performed acceleration-driven channel flow simulations at Re $\approx$ 13 and Ma = 0.2. The domain consisted of 80 cells in each direction consisting of 100 particles in a cell, making a total of 51.2 Million particles in the domain. All simulations were performed using 200 processors with pencil decomposition of the channel geometry. Fig.(3.10) shows the split of the time taken for the above simulation by different subroutines for various values of the time step, Δt. We see that for a 40-fold increase in Δt, there is an 8-fold time saving for the simulation to reach one diffusion time. Although the time taken per step for collision and communication increases with Δt, the number of calls to these functions until 1 diffusion time has reduced.

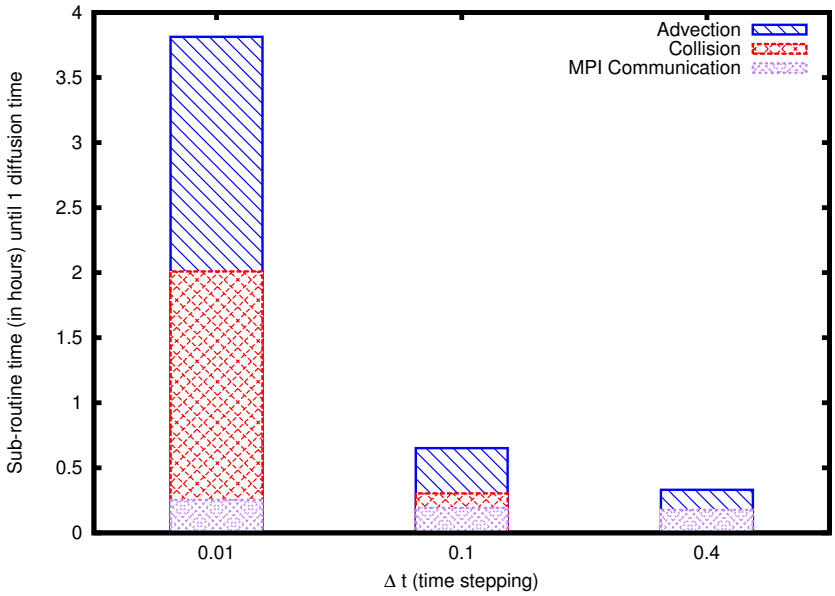

Figure 3.10: Time taken by the different sub-routines for the same simulation until 1 diffusion time using various values of Δt

We can now make huge savings in computational time by using a large Δt and through an appropriate viscosity correction.

3.4 MPI Parallelization

The performance of any particle code is expected to be compute bound. In other words, if CPU speed is doubled code speed also increases roughly by same factor. However, as the CPU speed has considerably gone up over years, now simulations are done with much higher number of particles. As the number of particle increases, at some point communication between CPU and RAM becomes a bottleneck. The performance of such codes are termed as memory bound. In other words, the code performance increases significantly if the memory bandwidth is increased via an increase in RAM speed or number of memory channels. Indeed, this is being increasingly observed that many scientific codes are becoming memory bound (Shet *et al.*, 2013).

As for an example, let us consider a flow simulation of $\mathrm{Re} = 10$ and $\mathrm{Ma} = 0.1$ simulation in a cubical domain. As, this setup correspond to $\mathrm{Kn} \sim 10^{-2}$, with a cell width of $\Delta x = \lambda$ we would need 100 cells in every direction. Thus, for a three dimensional simulation with number of particle in a cell $N_0 = 100$, we require 100 million particles. This would require a total memory of 4.8 GB with 6 phase space coordinates per particle. Table 3.2 shows the memory requirement as one moves to higher Reynolds numbers. In the advection step, we have around ~ 10 memory reads per particle and in the collision step, we have around $\sim 6N_0$ memory read operations per particle. Thus, approximately using an order of magnitude analysis, we can say that the number of memory operations in a single step of DSMC is $10N_0$ per particle. Similarly, a single collision event has approximately ~ 100 compute operations and there are $\sim N_0^2$ such collision events per cell. So, for approximately an order of magnitude analysis, we can say that the number of compute operations in a single step of DSMC is $100N_0^2$. Thus, for the present case, the total amount of memory transfer is $8 \times 10^6 \times 10 \times 100 \times 100$ which is ~ 1000GB of data transfer per time step. Similarly, we would require $10^6 \times 100 \times 100 \times 100$ which is ~ 1000 GFLOP per time step.

If we consider, AMD ROME (Sankhyasutra Labs cluster) or Intel XEON 8268 (PARAM Yukti cluster at JNCASR) (see Tables 3.3 and 3.4 for details), it has a per node memory bandwidth of $\sim$ 350GB/s and $\sim$ 175GB/s respectively (with $\sim$ 80% efficiency). Both have a

Re	Number of particles (in Billions)	Memory requirement (in TBs)
10	0.1	0.0048
100	100	4.8
1000	100000	4800

Table 3.2: Number of particles and memory requirement for a full-domain DSMC simulation with 100 particles per cell and $\Delta x = \lambda$

peak teraflop rating of around 4.5TFLOP per node. Thus, ideally compute should take $\sim$ 1s on both AMD ROME and intel XEON 8268. Similarly, $\sim$ 3s to $\sim$ 6s should take to perform the memory operations.

CPU	Intel	AMD
$n_{\text{freq}}^{(\text{RAM})}$	2.933 GHz	3.2 GHz
w	8 bytes	8 bytes
n_{channels}	6	8
n_{peak}	281.25 GB/s	409.6 GB/s

Table 3.3: RAM specifications of the Intel (PARAM Yukti) and AMD (Sankhyasutra labs) clusters

CPU	Intel	AMD
processors/node	48	128
$n_{\text{freq}}^{(\text{CPU})}$	2.9 GHz	2.25 GHz
FLOPS/cycle	32	16
r_{peak}	4.454 TFLOPS/s	4.608 TFLOPS/s

Table 3.4: CPU specifications of the Intel (PARAM Yukti) and AMD (Sankhyasutra labs) clusters

For simulations at higher Reynolds numbers or lower Knudsen numbers, the memory and compute requirement are huge and serial jobs would not be possible in realistic times. Parallelization allows us to distribute the work load among multiple processors and allows for particle simulations in such regimes to be done in reasonable times.

Parallelization can be either done by dividing the total number of particles in the domain among the different processors or via splitting the domain. With particles advecting over large distances, and with collisions being processed per cell, over time the simulation is bound to become slower with the memory addresses of colliding particles far from each other resulting in cache misses. Such an algorithm would require an efficient sorting algorithm. But, the time requirement would still be larger than an algorithm implemented based on domain decomposition.

Domain decomposition is the method by which the simulation geometry is divided into smaller domains, each of a similar shape. Each subdomain is then allocated to every processor. Fig.(3.11) illustrates one such technique known as slab decomposition where in the geometry is split along one dimension. Here, slabs are created along the wall normal direction with $\mathrm{N_p}$ such slabs, where $\mathrm{N_p}$ is the number of processors allocated.

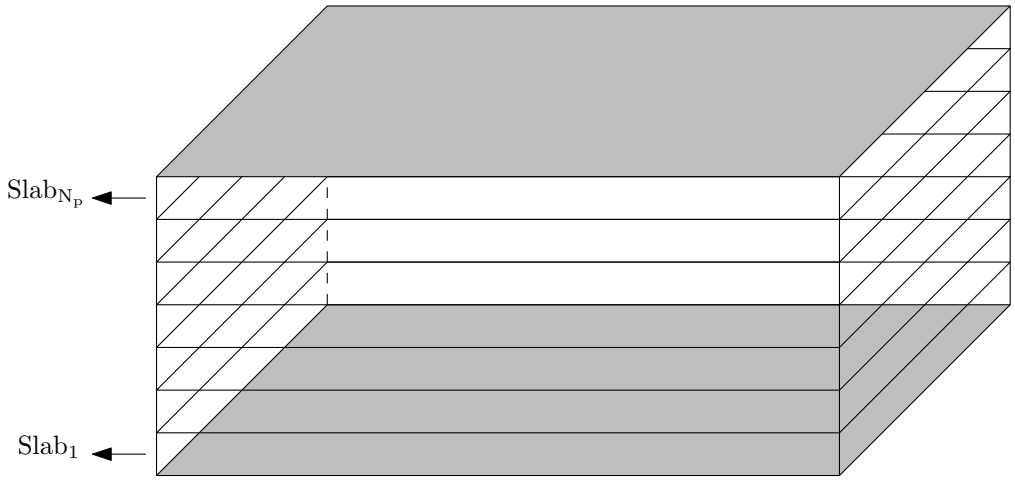

Figure 3.11: Slab decomposed channel flow geometry. N_p is the number of allocated processors

3.4.1 Pencil Decomposition

For our case of interest, we have divided a channel flow geometry into pencil shaped cuboids along the channel height. Fig.(3.12) shows a description of this domain decomposition technique. Each of these pencils represents the territory marked by a processor. Unlike the slab, pencil decomposition splits the domain along two-dimensions. Each pencil is then further divided into cells where particle collisions are processed. All the DSMC subroutines are performed independent of other pencils. In order to stitch back the domain, a communication routine among the neighboring regions is performed every time step. With the pencil decomposed domain having 4 neighbors per pencil (corresponding to each face) against only 2 neighbors in a slab decomped domain, the total number of communication calls between processors is increased in pencil as compared to slab while reducing the amount of data transfer per communication (as the individual volume per pencil is smaller than that of a slab).

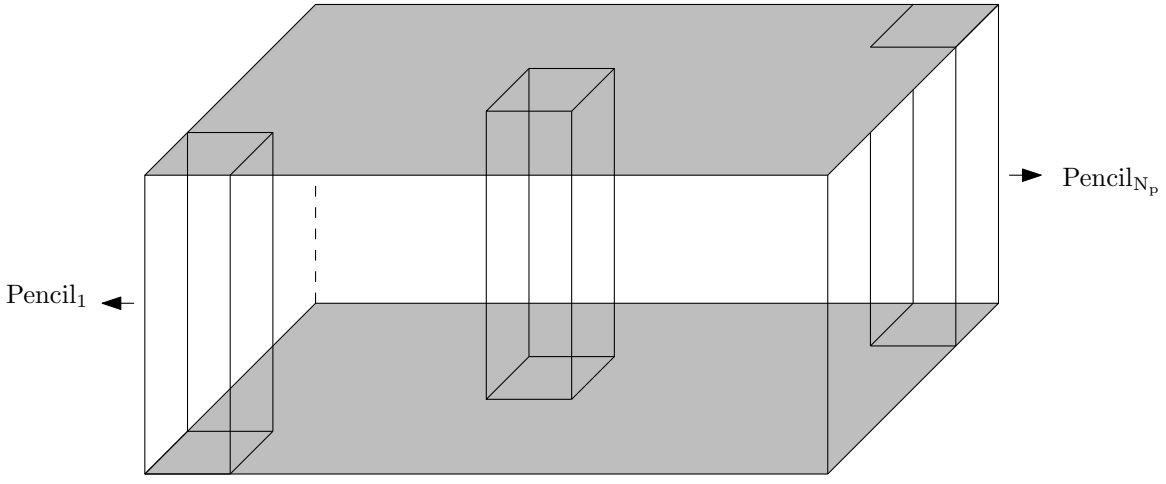

Figure 3.12: Pencil decomposed channel flow geometry. N_p is the number of allocated processors

We now describe how the parallelization is done. Fig.(3.13) shows the immediate neighbors of region P_0 as viewed from along the wall normal direction. Those bordered by dashed lines are the diagonal neighbors with a common edge, while the others have a common face with P_0. With every particle advection, particles cross over to its immediate neighbors and one needs to send the data corresponding to those particles to its neighbors. Every pencil requires a total of 4 communications, with its solid bordered neighbors (P_1, P_2, P_3 and P_4). Any communication

with the diagonal neighbors is accounted for due to dimensional independence.

Particles whose x coordinate has crossed over to the next processor after advection is sent over to P_1 (for particles traveling in the positive direction), irrespective of its new y position. Then the new y position is checked to know if a transfer to P_5 or P_8 is required, else the particle stays within P_1. The same rule applies to particles traveling in the negative direction. The particles that do not cross over along the x direction, are checked for transfer to either P_2 or P_4. In this way, communication to all neighbors is achieved. As the number of particles crossing over a face is usually larger than through an edge, this method would be much more efficient than communicating to all 8 neighbors.

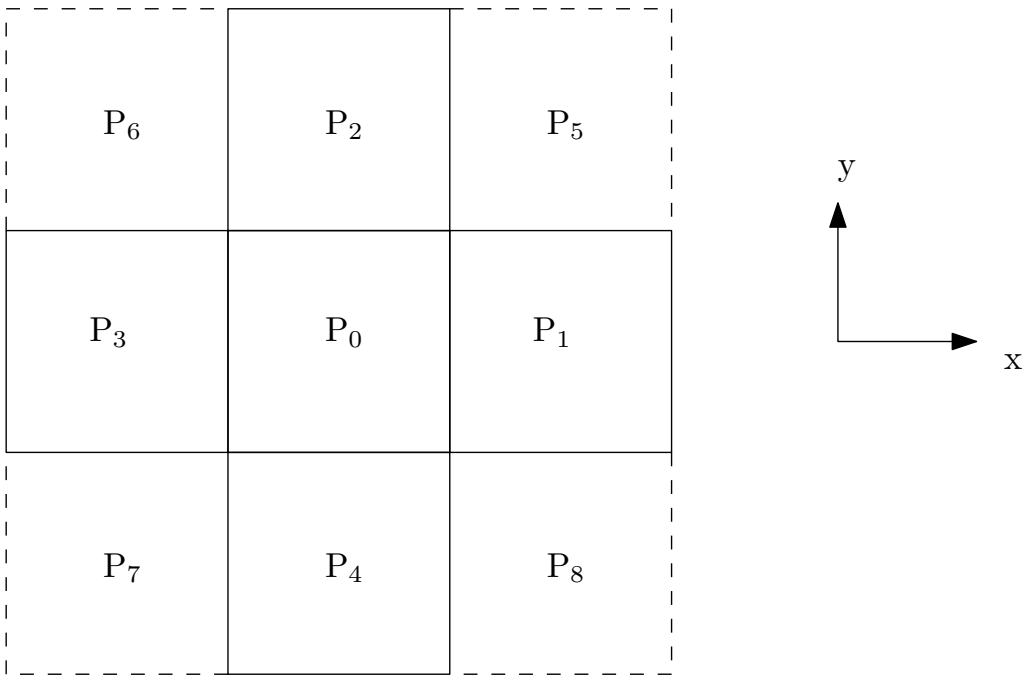

Figure 3.13: A cross-sectional view showing the processor layout in a plane of the Pencil decomposed geometry.

With this, the updated DSMC routine is shown in Algorithm 1.

Algorithm 1 Parallel DSMC Methodology

```
for i,j ϵ DSMC region do
    Initialize particles in domain
    while steps <n do
        Particle Advection
        Particle Collisions
        MPI Communication with neighbors
    end while
end for
```

3.4.2 Comparison to SPARTA

SPARTA (Gallis *et al.*, 2014) is an open-source DSMC code for rarefied particle simulations. With all the modifications to improve the performance of our DSMC solver, we performed similar simulations using the standard DSMC solver, SPARTA. We have done numerical studies to compare the scaling behavior of both solvers. With our limitations on the available computational resources, simulations were done using upto 800 processors to observe parallel scaling.

The total time taken to compute 1000 iterations was compared for every instance.

As shown in Fig.(3.14), our in-house DSMC solver has performed much faster and has shown a good scaling behavior upto 800 cores on an Intel Xeon Platinum 8268 cluster. Similar studies performed on an AMD EPYC 7742 cluster (Sankhyasutra labs) with 4096 processors has shown a linear scaling behavior of our solver.

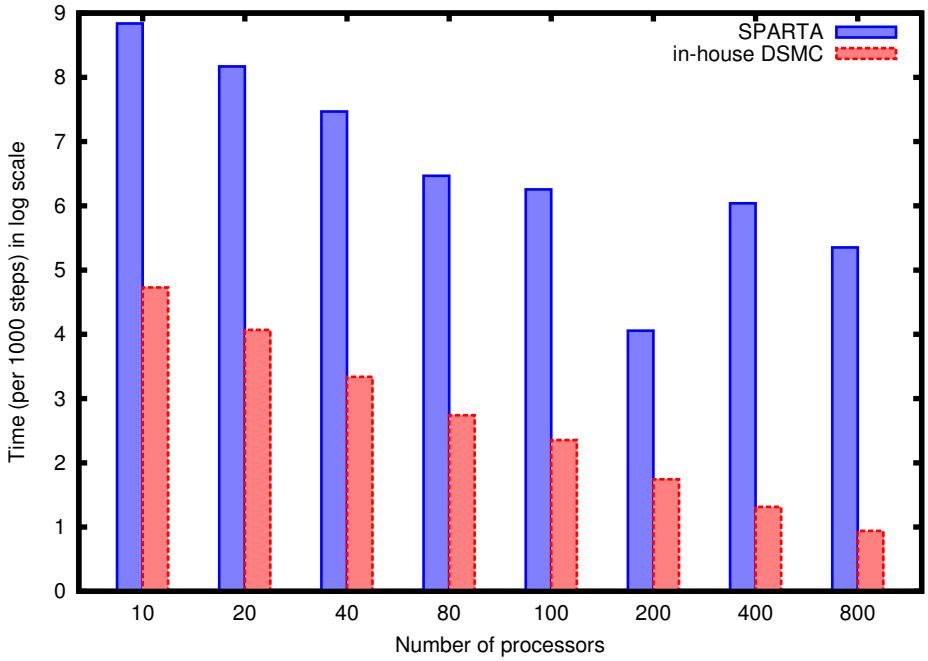

Figure 3.14: Comparison of our in-house DSMC solver against SPARTA for various number of processors on the Intel XEON 8268 (PARAM Yukti cluster at JNCASR)

Fig.(3.15) shows a comparison of the time taken for channel flow simulations using our DSMC solver at various Reynolds numbers. We see that with increasing Reynolds numbers, the time requirement for particle simulations makes it infeasible to perform moderate Re fluid simulations on moderate computing resources.

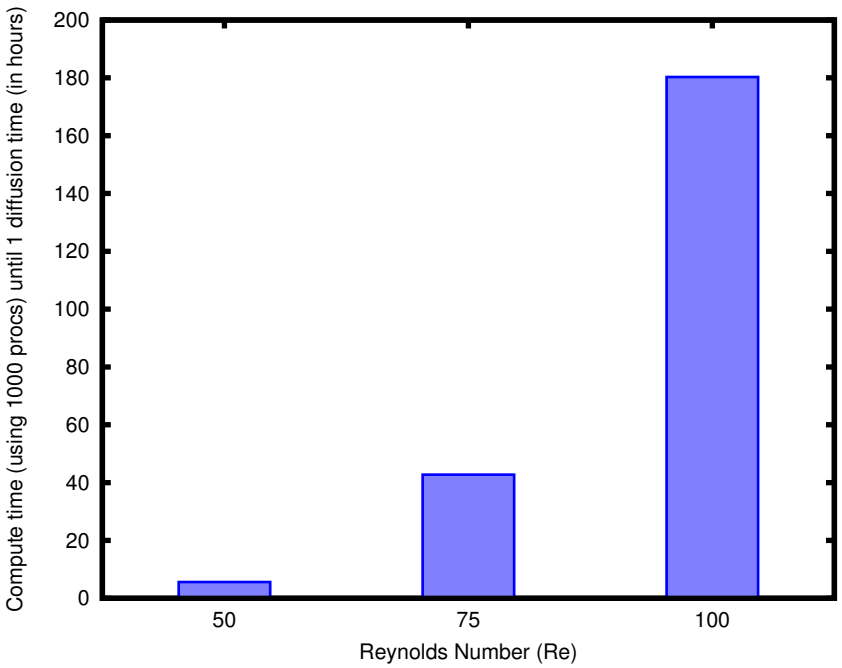

Figure 3.15: Time taken (in hours) by a Pure DSMC simulation for various Reynolds number and a system dimension of $6L \times L \times 3L$. With increasing Re, DSMC simulations take unrealistic times.

3.5 DSMC Benchmark

In this section, we present a few benchmark cases for validating our DSMC solver. The DSMC solver was initially tested on canonical cases such as the plane Couette and Poiseuille flow. It is shown in the literature that for moderate Knudsen number, the flow profile for a plane Poiseuille flow (see Fig(3.16)), where the fluid is accelerated by a pressure/force F between two parallel plates kept apart at a distance L is given by(Yudistiawan *et al.*, 2010).

$$u_y = F\frac{y^2}{2\tau} + A_2 + A_3 \cosh\left(\frac{y}{c_1 \mathrm{Kn}_1 L}\right), \tag{3.32}$$

where A_2, A_3 and c_1 are appropriate constants dependent on the discrete velocity model used. Here, Knudsen number is defined to be

$$\mathrm{Kn}_1 = \frac{\tau}{L}\sqrt{\frac{2k_\mathrm{B}T_0}{M}}. \tag{3.33}$$

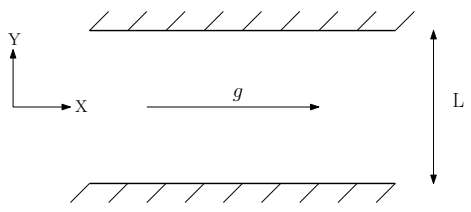

Figure 3.16: A sketch of the 2D plane poiseuille flow geometry with the fluid driven by a constant body force g along the streamwise direction

The cosh term in the velocity profile arises from symmetry of the flow around the centerline ($y = 0$). The moments from the simulations were averaged over a time scale of 100τ to reduce

statistical scatter. Fig.(3.17) shows the instantaneous and time-averaged profile for a transient plane Couette flow.

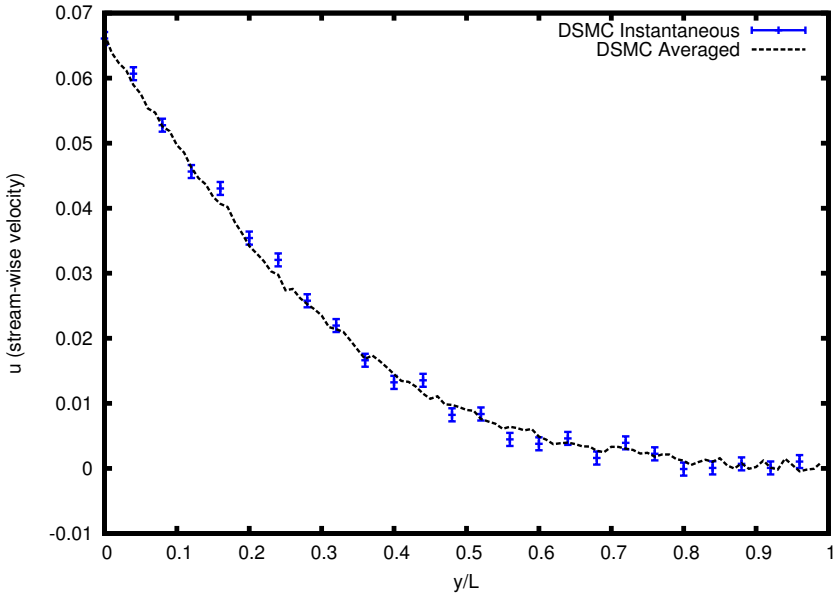

Figure 3.17: Steady State Re = 50, Ma = 0.1 fluctuations of the DSMC plane Couette flow streamwise velocity profile compared against a time-averaged profile

Poiseuille flow simulations were performed at Re ≈ 20 and Ma = 0.1 and the steady state flow profile(see Fig(3.18)) was compared against the profile from Equation 3.32.

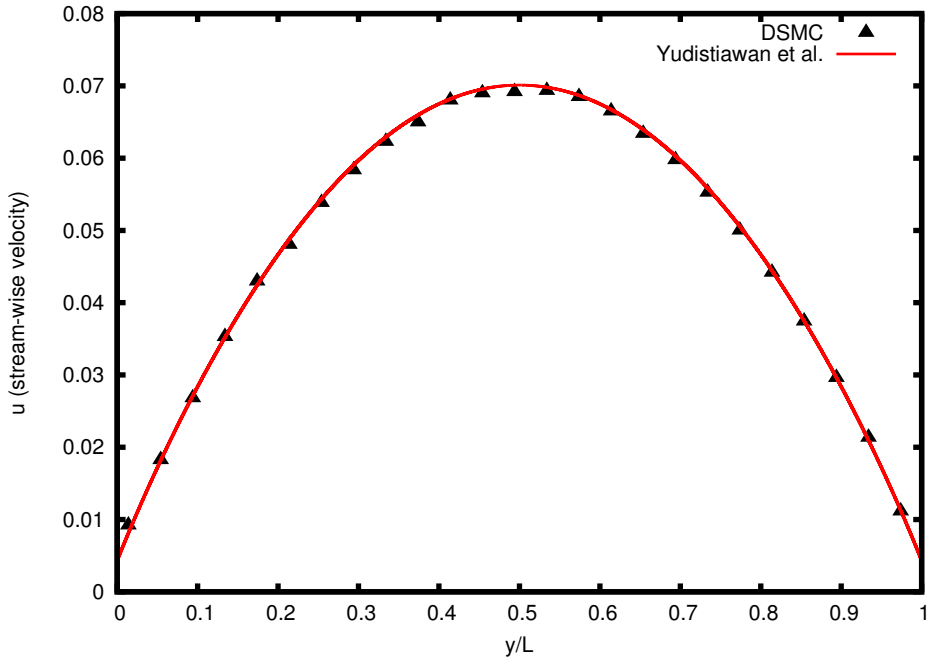

Figure 3.18: Steady State Re ≈ 20, Ma = 0.1 streamwise velocity profile compared against the approximate profile from ref. (Yudistiawan *et al.*, 2010)

Similarly, the flow profile for a plane Couette flow (see Fig(3.19)), where two plates are kept at a distance L and move with the velocities U_1 and U_2 is of the form (Yudistiawan *et al.*, 2010)

$$u_y = \frac{U_1 + U_2}{2} + \frac{y}{\Theta}\frac{U_2 - U_1}{L} + \frac{1}{K_1}(U_2 - U_1)\sinh\left(\frac{y}{K_2\,\mathrm{Kn}_1\,L}\right), \tag{3.34}$$

where Θ, K_1 and K_2 are constants whose details depends on the discrete velocity model used to obtain this profile.

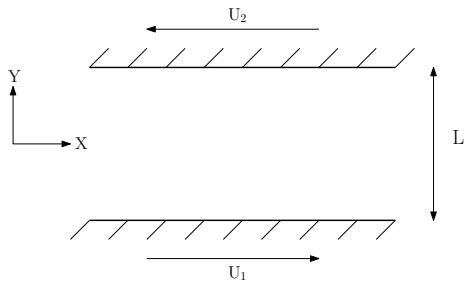

Figure 3.19: A sketch of the 2D plane couette flow geometry with the fluid driven by the anti-parallel movement of the walls with velocities U_1 and U_2 along the streamwise direction

Fig(3.20) shows the DSMC simulation results at a moderate Reynolds number of Re = 50 and at Ma = 0.2 for plane Couette flow compared against the analytical solution from Eqn(3.34). The slight difference in the velocity profiles as compared to Analytical and LB solutions are due to the ineffective representation of viscosity when using a cell size $> 0.5\lambda$. This mismatch can be resolved by using smaller cell widths.

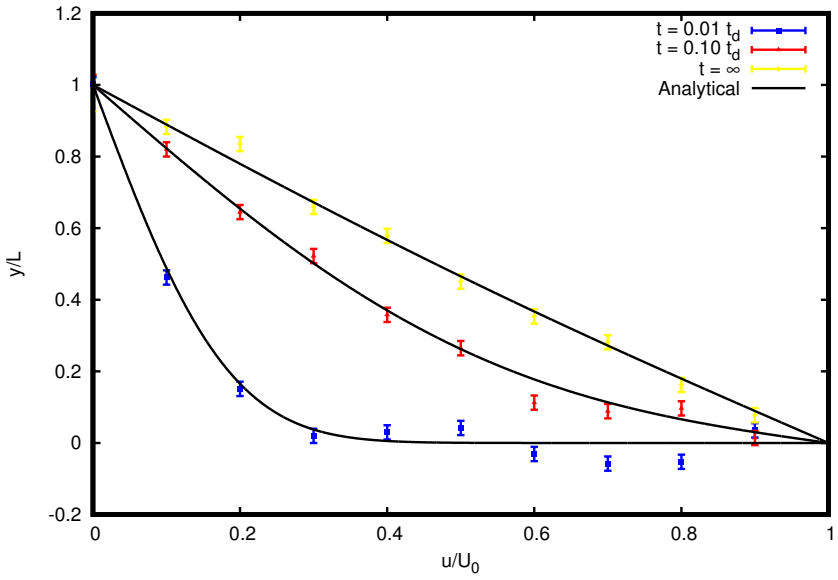

Figure 3.20: Nondimensional planar Couette developing flow profile compared with the Analytical solution at various diffusion times

DSMC simulations performed in the transitional regime at Kn = 0.1, Ma = 0.1 and the obtained streamwise velocity profile compared with Equation 3.32 is shown in Fig.(3.21).

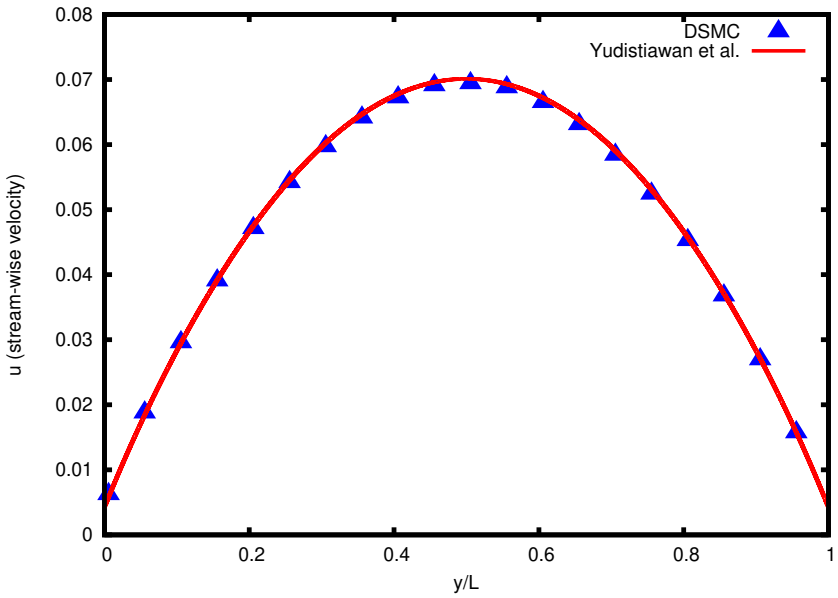

Figure 3.21: Steady State Kn = 0.1, Ma = 0.1 streamwise velocity profile compared with the profile from ref. (Yudistiawan *et al.*, 2010)

3.6 Outlook

In this chapter, we tried to demonstrate the capability and the methods used in improving the performance of traditional DSMC solvers. We have shown through numerous test cases that our model correctly captures hydrodynamic behaviour in various flow regimes. Simulations done in the laminar no-slip and in the rarefied regimes illustrate this point. Simulations were performed in a channel flow geometry for plane Poiseuille and plane Couette flows and have shown excellent agreement with their corresponding analytical and numerical profiles. We also discussed briefly about slab and pencil methods of domain decomposition. We also emphasized on the methods used in making computational gains through the use of modified transport parameters allowing for larger cell sizes (Δx) and time steps (Δt). Using a few test cases, we have demonstrated the huge time savings made in using this viscosity correction.

Chapter 4
DSMC - Lattice Boltzmann Coupling

In this chapter, we discuss the hybrid Direct Simulation Monte Carlo - Lattice Boltzmann (LB) scheme for fluid simulations in various hydrodynamic regimes. As the LB model we have used the recently proposed RD3Q41 model (Kolluru *et al.*, 2020). This model was chosen due to its ability to describe low Mach number flows and acoustics in a realistic fashion. We start with a brief discussion on the validity, usefulness and applicability of the proposed approach. We then describe the coupling strategy employed through the use of a domain-decomposition technique. The implemetation of a communication methodology through the use of a truncated Hermite polynomial expansion of the single particle distribution function is reviewed. Finally, we validate our approach by simulating flows in the laminar continuum, transitional and turbulent regimes.

4.1 Coupling necessity

In this section, we describe the purpose of such a method and the strategy used in devising the current coupling scheme. We review two basic questions in this section: Why does one need a Micro-Meso coupling? And how can one implement the same? We answer the former by showing the advantage of the solver from a physics and a computational point of view.

One generally quantifies the non-equilibrium nature of the flow through the use of the Knudsen number, Kn. Flows are classified into the continuum (Kn < 0.01), Slip flow ($0.01 <$ Kn < 0.1), transitional ($0.1 <$ Kn < 10) and free-molecular (Kn > 10) regimes based on this. Rarefied flows typically have large non-equilibrium contribution to the flow. Interest in such flows are due to its applicability in micro and nano scale flows such as in MEMS based devices (Nie *et al.*, 2002). One is interested in being able to predict such flows with a good amount of accuracy. Classical fluid solvers such as Finite Volume and others which employ a discretization of the Navier-Stokes equations are not capable of in providing an accurate description of hydrodynamics beyond the continuum regime. With corrections to the governing equations arising due to significant non-equilibrium effects, one requires a higher-order description of the flow to capture such deviations. Although, modifications to continuum solvers that reproduce a velocity slip and temperature jump at the wall are used, they do not faithfully reproduce the kinetic nature of such flows (Zhang *et al.*, 2012). Therefore, one often solves the Boltzmann equation that provides a solution from the rarefied to the continuum regimes.

Kinetic schemes such as the Lattice Boltzmann model (Chen & Doolen, 1998; Succi, 2001) have been successful in simulating flows in a wide variety of regimes, even though the method works at finite Knudsen numbers. But in the high Knudsen number range, it too suffers from the discrete nature of the model and its restricted isotropy. Obvious modifications through the use of higher-order models (Ansumali *et al.*, 2007; Atif *et al.*, 2018; Namburi *et al.*, 2016; Yudistiawan *et al.*, 2010) have shown good success. But particle-based methods such as DSMC (Bird & Brady, 1994) have remained the preferred choice in this regime. DSMC also provides a kinetic description through its approximation of the Boltzmann Equation (see Section 3.2). But due to its huge computational cost, its use has been restricted to small scale flows. As the computational cost of a DSMC simulation scales as Kn^{-3}, one requires multi-peta scale computing systems to perform a modest turbulent flow simulation (Gallis *et al.*, 2017, 2018) that is otherwise easily done using methods like LBM.

We can therefore look to strike a balance and extend the applicability of such kinetic solvers by using a hybrid solver where one can use DSMC in regions of the domain where

non-equilibrium effects persist and a LB model in the rest of the domain. Such a domain-decomposition technique employs the versatality of DSMC while also keeping the numerical expense low in contrast to particle methods. In our study, we use a class of LBM models known as crystallographic LBM. Such models are suited for optimal discretization of the spatial domain using a body-centered cubic (BCC) ordering of the grid points (Namburi *et al.*, 2016). We have used a 41-velocity crystallographic model, hereby referred to as the RD3Q41 model (Kolluru *et al.*, 2020). This model is well suited for weakly compressible aeroacoustic and thermoacoustic flow simulations. It was shown that this model ensures a higher order isotropy against the D3Q27 model and is therefore suitable for simulations in the moderate Knudsen number regime with appropriate boundary conditions. One can now delve into rarefied regimes with DSMC in control of the Knudsen layer. Alternate hybrid methods of coupling where the solution of a transient Boltzmann equation evolution is mapped to determine the evolution of the macroscopic velocity field (Al-Mohssen *et al.*, 2007) are used. Higher-order contributions from the stress tensor and heat flux are excluded in such methods. These schemes use a time-decomposed approach rather than a domain-decomposed method described in our coupling. In this two-way coupling, we have modeled a plane channel flow geometry with the DSMC solver near the wall and the relatively coarser LB solver in the bulk of the fluid.

As opposed to similar works using a DSMC-Lattice Boltzmann (Di Staso *et al.*, 2016) and MPCD-Lattice Boltzmann (Montessori *et al.*, 2020) coupling in literature which represented a hybrid particle-continuum coupling and where one of the solvers extends completely through the domain, we intend to restrict our solvers exclusively in our regions of interest. Such solvers which occupy the full domain have a relaxed stability criteria and are less affected by the thermal noise from the particle layer. In our case, restricting the region of control of the solvers allows the different solvers to be allocated in regions where they are most efficient in capturing the physics of the flow. Here, we restrict our DSMC solver only near the wall and the LB solver within the bulk with a buffer zone for communication between them. This gives us the flexibility of moving towards high Knudsen and turbulent flow regimes where the molecular wall layer plays an important role in capturing the boundary layer effects. This also plays to our advantage in being able to use lattice models with finite number of discrete velocities in moving towards such regimes. The schematic in Figure 4.1 gives a picture of the implemented coupling scheme. The buffer zone represents the region where a two-way DSMC-LB and LB-DSMC communication is executed.

Whenever thermal noise is desired, one works with a fluctuating LBM model (Adhikari *et al.*, 2005) or is artificially induced with thermal noise through the use of geometric modifications like a bump (Roy *et al.*, 2017) or through specific initial fluctuationg velocity conditions (Gallis *et al.*, 2018). With the noise for turbulent flow simulations provided by DSMC, we would like to know its influence on the transition process. With a DSMC layer near the wall, we now have a realistic depiction of the influence of micro-roughness on the flow. With important regions governing the flow dynamics covered by a particle-based method one could argue that it provides an accurate description of the flow from a kinetic point of view.

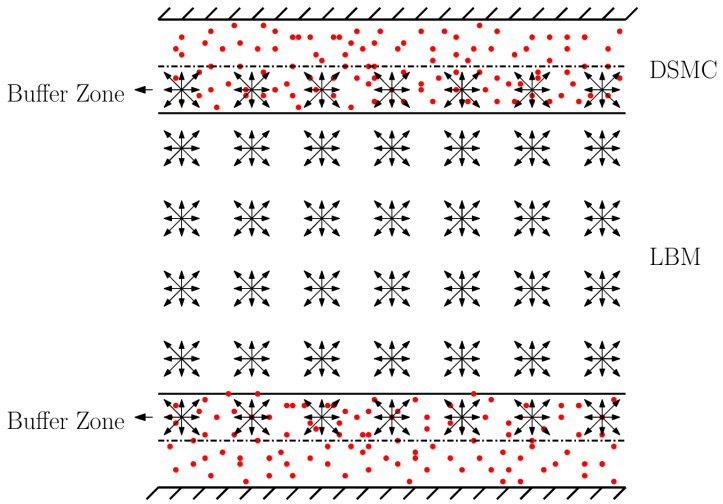

Figure 4.1: DSMC-LB Coupling Geometry. DSMC region is present only near the wall and the LB domain extends through the rest of the bulk. The buffer zone represents the region of communication between the two solvers.

In the following sections, we describe the communication routine at the interface of the two solvers, the time gain against a pure DSMC solver and validate the results obtained in various hydrodynamic regimes.

4.2 Coupling communication

In this section, we describe the communication routine employed in the buffer layer between the two solvers in order to transport wall effects into the bulk of the fluid and vice-versa. The basic idea is to use a projection scheme where in the macroscopic quantities are used as the basis of communication. The exchange is through the creation of a 13 moment Grad distribution function (Grad, 1949) expansion using the density (ρ), momentum density (ρu_α), stress tensor ($\sigma_{\alpha\beta}$) and the heat flux (q_α) as macroscopic field variables.

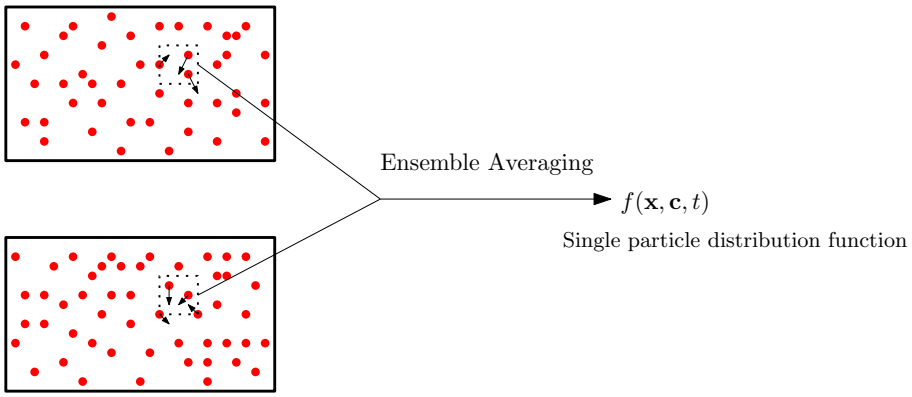

Figure 4.2: The conversion from a particle representation to a mesoscopic one. An ensemble average over particle systems generates the single particle probability distribution function.

In order to understand the mapping between a microscopic and a mesoscopic representation, we look at how the two are related. Fig.(4.2) shows the conversion between a particle and the single particle distribution function representations. At every point $\boldsymbol{x}_0$ in space and at time t_0, we can get the single particle distribution $f(\boldsymbol{x} = \boldsymbol{x}_0, \boldsymbol{c}, t = t_0)$ by ensemble averaging the probability of finding particles with different velocities $\boldsymbol{c}$. One can now relate the macroscopic properties to the micro and mesoscopic description as shown in Table 4.1.

Macroscopic	Mesoscopic	Microscopic
ρ	$\int f(\boldsymbol{x}, \boldsymbol{c}, t) d\boldsymbol{c}$	mN/V
$j_\alpha = \rho u_\alpha$	$\int f(\boldsymbol{x}, \boldsymbol{c}, t) c_\alpha d\boldsymbol{c}$	$m \sum_{i=1}^{N} c_{i\alpha}/V$
$P_{\alpha\beta}$	$\int f(\boldsymbol{x}, \boldsymbol{c}, t) c_\alpha c_\beta d\boldsymbol{c}$	$m \sum_{i=1}^{N} c_{i\alpha} c_{i\beta}/V$

Table 4.1: Representation of macroscopic quantities (density (ρ), momentum density (j_α) and pressure tensor ($P_{\alpha\beta}$)) in terms of the single particle distribution function $f(\boldsymbol{x}, \boldsymbol{c}, t)$ and the particle velocities c_i. Here m is the mass of every particle (taken to be unity) and V is the volume of a small positional space volume.

We can see that one can use the macroscopic variables as the basis for conversion between the microscopic and mesoscopic representations of the fluid. For such a task, it would be convenient to have a representation of the single particle distribution function in terms of the macroscopic moments it represents. Harold Grad (Grad, 1949) showed that using a combination of Hermite polynomials as orthonormal basis functions that span the space of the single particle distribution function, one obtains these macroscopic moments in the coefficients of this expansion.

4.2.1 Grad Hermite Expansion

The Hermite orthonormal polynomial expansion of the single particle distribution function $f(\boldsymbol{x}, \boldsymbol{\xi}, t)$ is as shown below,

$$f(\boldsymbol{x}, \boldsymbol{\xi}, t) = w(\boldsymbol{\xi}) \sum_{n=0}^{N} \boldsymbol{a}^{(n)}(\boldsymbol{x}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}), \tag{4.1}$$

where $\boldsymbol{\xi} = \boldsymbol{c} - \boldsymbol{u}$ is the peculiar velocity, $\mathcal{H}^{(n)}(\boldsymbol{\xi})$ are the hermite polynomials at n^{th} order, $\boldsymbol{a}^{(n)}(\boldsymbol{x}, t)$ are their corresponding coefficients and $w(\boldsymbol{\xi})$ is the weight/generating function for the various hermite polynomials (Grad, 1949). At various orders, the hermite polynomials obtained through Gram-Schmidt orthonormalization are as follows,

$$\begin{aligned}
\mathcal{H}^{(0)}(\boldsymbol{\xi}) &= 1, \\
\mathcal{H}^{(1)}_{\alpha}(\boldsymbol{\xi}) &= \frac{\xi_\alpha}{\sqrt{\theta_0}}, \\
\mathcal{H}^{(2)}_{\alpha\beta}(\boldsymbol{\xi}) &= \frac{(\xi_\alpha \xi_\beta - \theta_0 \delta_{\alpha\beta})}{\sqrt{2\theta_0^2}}, \\
\mathcal{H}^{(3)}_{\kappa}(\boldsymbol{\xi}) &= \frac{(\xi^2 \xi_\kappa - 5\theta_0 \xi_\kappa)}{\sqrt{10\theta_0^3}}.
\end{aligned} \tag{4.2}$$

The coefficients corresponding to these are determined using the property of orthonormality of these hermite polynomials.

$$\boldsymbol{a}^{(n)}(\boldsymbol{x}, t) = \int f(\boldsymbol{x}, \boldsymbol{\xi}, t) \mathcal{H}^{(n)}(\boldsymbol{\xi}). \tag{4.3}$$

Therefore, one obtains the following coefficients following Equation 4.3.

$$
\begin{aligned}
\boldsymbol{a}^{(0)} &= \rho, \\
\boldsymbol{a}^{(1)}_{\alpha} &= 0, \\
\boldsymbol{a}^{(2)}_{\alpha\beta} &= \frac{\sigma_{\alpha\beta}}{\sqrt{2\theta_0^2}}, \\
\boldsymbol{a}^{(3)}_{\kappa} &= 2\frac{q_{\kappa}}{\sqrt{10\theta_0^3}},
\end{aligned}
\tag{4.4}
$$

where $\sigma_{\alpha\beta}$ and q_{κ} are the stress and heat flux tensors respectively. Substituting the hermite polynomials and their coeffcients into Equation 4.1, we obtain the representation of the single particle distribution function as a truncated series in terms of the macroscopic moments.

$$
f(\boldsymbol{x},\boldsymbol{\xi},t) = F^{\text{grad}}(\boldsymbol{x},\boldsymbol{\xi},t) = w(\boldsymbol{\xi})\left[\rho + \frac{\sigma_{\beta\gamma}}{2\theta_0^2}(\xi_{\beta}\xi_{\gamma} - \theta_0\delta_{\beta\gamma}) + \frac{q_{\kappa}}{5\theta_0^3}(\xi^2\xi_{\kappa} - 5\theta_0\xi_{\kappa})\right]. \tag{4.5}
$$

Equation 4.5 represents the single particle distribution function as a combination of its macroscopic moments. Here, we have used $F^{\text{grad}}(\boldsymbol{x},\boldsymbol{\xi},t)$ to distinguish the continuous from the discrete grad function introduced in the next subsection.

4.2.2 Discrete Grad Expansion

In the LBM framework, one is interested in the discrete form of the hermite expansion (Shan & He, 1998). The weight for generating the polynomials is the discrete version of the Maxwell-Boltzmann distribution in the fixed reference frame, i.e., $\boldsymbol{u} = 0$. This would make this discrete representation suitable for low Mach number flows. The weights (w_i) corresponding to the discrete velocity set is therefore the discrete equilibrium function. Through a similar process of Gram-Schmidt orthonormalization, we obtain the discrete hermite polynomials as shown below,

$$
\begin{aligned}
\hat{\mathcal{H}}^{(0)} &= 1, \\
\hat{\mathcal{H}}^{(1)}_{\alpha} &= \frac{c_{i\alpha}}{\sqrt{\theta_0}}, \\
\hat{\mathcal{H}}^{(2)}_{\alpha\beta} &= \frac{(c_{i\alpha}c_{i\beta} - \theta_0\delta_{\alpha\beta})}{\sqrt{2\theta_0^2}}, \\
\hat{\mathcal{H}}^{(3)}_{\kappa} &= \frac{(c_i^2 c_{i\kappa} - 5\theta_0 c_{i\kappa})}{\sqrt{10\theta_0^3}},
\end{aligned}
\tag{4.6}
$$

where $\hat{\mathcal{H}}$ denotes the discrete hermite polynomial. The inner product of two discrete hermite polynomials $\phi(c_i)$ and $\psi(c_i)$ is of the form,

$$
\langle \phi(c_i), \psi(c_i) \rangle = \sum_{i=1}^{n} w_i \phi(c_i)\psi(c_i), \tag{4.7}
$$

where n stands for the number of discrete velocities. The coefficients of the discrete expansion are again determined through the property of orthogonality of the polynomials defined as,

$$
\hat{\boldsymbol{a}}^{(n)}(\boldsymbol{x},t) = \sum_{i} f_i \hat{\mathcal{H}}^{(n)}(\boldsymbol{c}_i). \tag{4.8}
$$

Using Equations 4.7 and 4.8, the coefficients obtained are as follows,

$$\begin{aligned}
\hat{a}^{(0)} &= \rho, \\
\hat{a}^{(1)}_{\alpha} &= \frac{j_\alpha}{\theta_0}, \\
\hat{a}^{(2)}_{\alpha\beta} &= \frac{(P_{\alpha\beta} - \rho\theta_0\delta_{\alpha\beta})}{\sqrt{2\theta_0^2}}, \\
\hat{a}^{(3)}_{\kappa} &= \frac{(Q_\kappa - 5\theta_0 j_\kappa)}{\sqrt{10\theta_0^3}}.
\end{aligned} \tag{4.9}$$

Higher order polynomials are functions of the discrete velocity model used (Chikatamarla *et al.*, 2006). Using these coefficients to write the discrete grad distribution function as a weighted combination of the discrete hermite polynomials, the expansion looks as follows,

$$f_i = w_i \left[\rho + \frac{j_\alpha c_{i\alpha}}{\theta_0} + \frac{(P_{\beta\gamma} - \rho\theta_0\delta_{\beta\gamma})}{2\theta_0^2}(c_{i\beta}c_{i\gamma} - \theta_0\delta_{\beta\gamma}) + \frac{(Q_\kappa - 5j_\kappa\theta_0)}{10\theta_0^3}(c_i^2 c_{i\kappa} - 5\theta_0 c_{i\kappa}) \right], \tag{4.10}$$

where f_i and w_i are the discrete distibution function and the weights associated with them corresponding to our RD3Q41 lattice (Kolluru *et al.*, 2020). We have expressed our distribution function as a projection of its macroscopic moments. This hermite polynomial expansion would serve as the basis of communication for our coupled solver.

4.2.3 DSMC to LB exchange

Here we describe the procedure followed in communicating from the DSMC solver to the LB solver in the bulk. As discussed in the previous subsection, we have devised a mechanism through the discrete Hermite polynomial expansion for communicating between the two solvers. Fig.(4.3) shows a schematic of this communication process between the two descriptions.

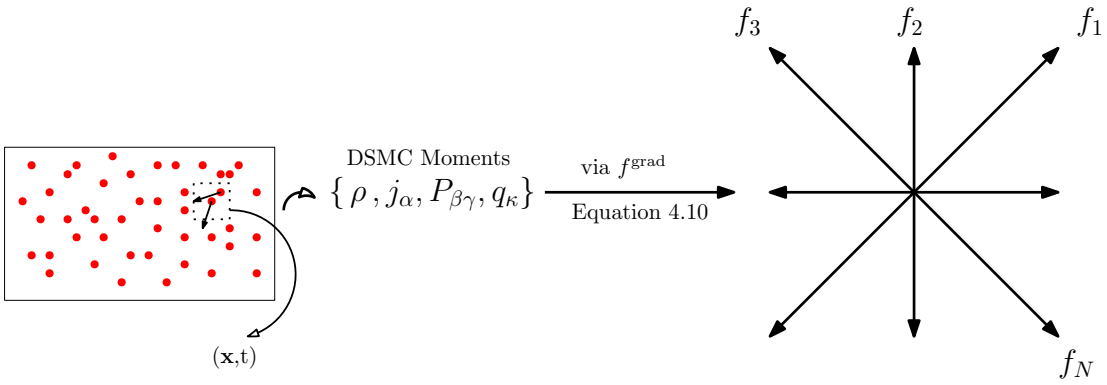

Figure 4.3: A schematic of the DSMC to LB exchange procedure. Particle velocities and their moments are averaged to obtain macroscopic quantities (as shown in column 3 of Table 4.1). These are then used to obtain the discrete grad distribution function (Equation 4.10) passed over to the LB node at the same $(\boldsymbol{x}, t)$ in the buffer layer.

Macroscopic quantities such as the density, momentum density, Pressure tensor, etc. are obtained through averaging the particle velocities and their moments as described in Table 4.1. These give a spatial data of the macroscopic moments from the DSMC region in the buffer layer. When these are plugged into the discrete version of the grad distribution function (Equation 4.10) we obtain the discrete populations of our 41-velocity model. We perform this procedure over a spatial region as shown in Fig.(4.8). This helps in adequate averaging of the moments

and improves the stability of the method by reducing the fluctuations in the moments obtained from the DSMC region in the buffer layer.

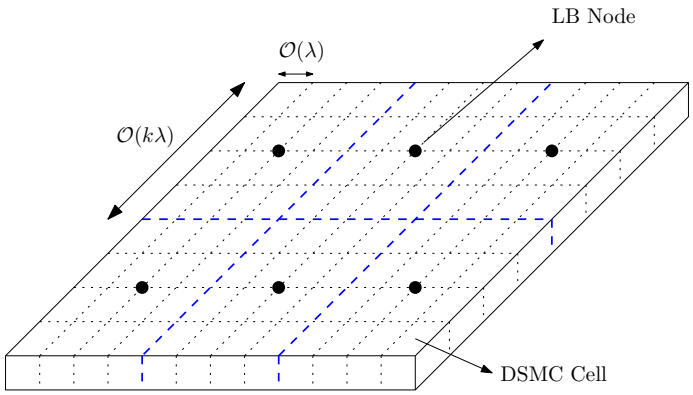

Figure 4.4: A schematic of the spatial averaging of the DSMC moments corresponding to every LB node in the domain. The blue dashed lines demarcate the boundaries of the LB nodes. The black dotted lines represent the DSMC cell boundaries. Here k represents the ratio of the number of DSMC cells corresponding to every LB node

$$k = \frac{\Delta x_{\mathrm{DSMC}}}{\Delta x_{\mathrm{LB}}} < 1. \tag{4.11}$$

The ratio of the number of DSMC cells to LB nodes in our domain is maintained to be larger than 1 (Equation 4.11). Therefore multiple DSMC cells correspond to a single LB node along all directions. The moments from these cells are then averaged over a region corresponding to the spatial extent of the node. Fig.(4.8) shows a schematic of this structure. All DSMC cells within the boundaries of the LB node (blue dashed lines) are averaged and are passed over to the corresponding node. This also ensures that the moments are sufficiently averaged to ensure the positivity of the equilibrium distribution in the LB domain. Once averaged these moments are projected onto the distribution function through Equation 4.10. The stochastic noise inherent to particle-based methods would make the coupling numerically unstable if communication is done without adequate averaging. We also temporally average over n steps (corresponding tot a physical time of around 100τ) before passing over to the LB nodes in the buffer layer. This ensures the smoothness of the moments and prevents any large gradients in the moments received by LB. This is physically justified because the timescales of the DSMC simulation are typically smaller than that of the mesoscopic solver.

4.2.4 LB to DSMC exchange

The domain separation into spatial cells as shown in Fig.(4.8) is used also for the reverse coupling. Similar to the DSMC to LB moment exchange procedure, we obtain moments from the LB nodes and are then used to generate particles within the spatial cells using a particle generation algorithm similar to the production of the Chapman-Enskog distribution (Garcia & Alder, 1998). The idea behind this particle generation being to sample particles randomly from a uniform distribution and filtering particles on the basis of an acceptance-rejection criteria. This criteria represents the distribution function obtained via the projection of the macroscopic moments obtained from the LB nodes as in Equation 4.5. Fig.(4.5) shows a schematic of the LB to DSMC exchange procedure. Algorithm 2 illustrates this particle generation procedure.

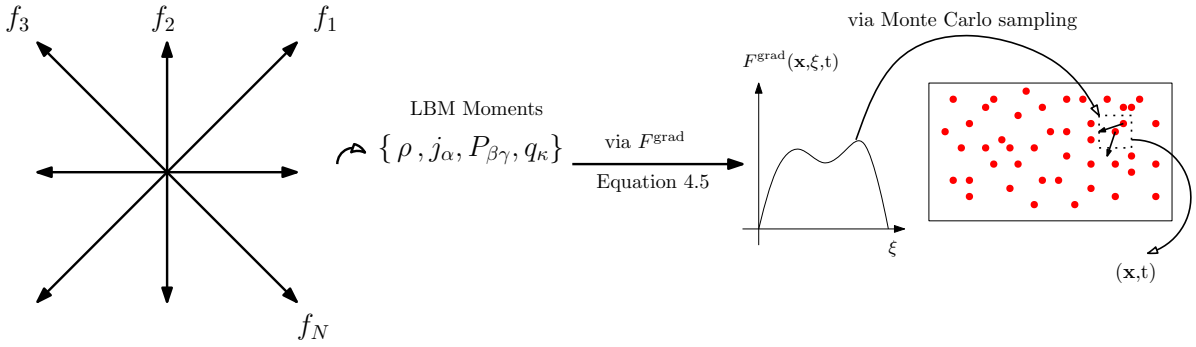

Figure 4.5: LB to DSMC moment exchange procedure showing the conversion of the discrete populations to the corresponding macroscopic moments. These moments are then used to obtain the continuous grad distribution function (Equation 4.5) which is subjected to a Monte Carlo sampling process to obtain the individual particle velocities that represent this set of macroscopic moments.

Algorithm 2 Particle Regeneration

1: **while** $N_i \leq N_{\text{bufferCells}}$ **do**
2: $n_{\text{poisson}} = \text{poissonDist}(\rho_{\text{LB}} V_c)$
3: **while** $n_0 \leq n_{\text{poisson}}$ **do**
4: Compute maxima of the coefficients of the Hermite polynomials

$$C = \max(|a^{(0)}_{\text{LB}}|, |a^{(1)}_{\text{LB},\alpha}|, |a^{(2)}_{\text{LB},\beta\gamma}|, |a^{(3)}_{\text{LB},\kappa}|)$$

5: Generate random particle velocities $\boldsymbol{\xi}$ from the Maxwell-Boltzmann distribution with temperature θ_{LB}
6: Select $\boldsymbol{\xi}$ **if** $C\mathcal{R} \leq F^{\text{grad}}(\boldsymbol{x}, \boldsymbol{\xi}, t)$, where $\mathcal{R}\ \epsilon\ (0,1)$; **else goto** step 5
7: Compute particle velocity using $c_\alpha = \xi_\alpha + u_{\text{LB},\alpha}$
8: **end while**
9: **end while**
10: rescale all c_α within cell to match $u_{\text{LB},\alpha}$ and θ_{LB}

The use of spatial cells for particle regeneration is advantageous as the number of particles for generation are increased, in turn increasing the accuracy of the resulting distribution represented by them.

4.2.5 Coupling Algorithm

In this subsection, we combine both the exchange procedures necessary for coupling. As mentioned earlier, we restrict our DSMC solver only near the walls. For this purpose, we use upto 4 DSMC cells from either walls, representing a distance of $\mathcal{O}(4\lambda)$ from the wall. The first LB node therefore starts after the first two DSMC cells. The steps followed for a single iteration of the coupling algorithm is as follows,

1. DSMC advection and collision for $n = \Delta t_{\text{LB}}/\Delta t_{\text{DSMC}}$ steps.
2. DSMC to LB moment exchange in the buffer layer through the Grad distribution function (Equation 4.10) followed by LB streaming and collision.
3. LB to DSMC moment exchange in the buffer layer with particle velocity regeneration using a Monte-Carlo sampling procedure of the generated Grad distribution function (Equation 4.5).

Algorithm 3 provides a pseudo code for clarity purpose.

Algorithm 3 Coupling methodology

```
for i,j ϵ DSMC region do
    while steps <n do
        Particle Advection
        Particle Collisions
        Calculate Moments in Buffer layer
    end while
end for
Calculate LB populations through Equation (4.10) in the buffer layer
for i,j ϵ LB region do
    Advect
    Collide
    Calculate Moments in Buffer layer
end for
Interpolate LB moments and regenerate particles using Algorithm 2
goto top
```

4.3 Code Performance

In this section we demonstrate the performance of our solver for parallel processing. In Fig.(4.6) the strong scaling behavior of the solver is illustrated at Re ≈ 65 and Ma $= 0.1$. The time taken to perform a single coupling step, which includes n DSMC and m LB steps is plotted against the number of processors. The solver has shown excellent scaling behavior upto 400 processors.

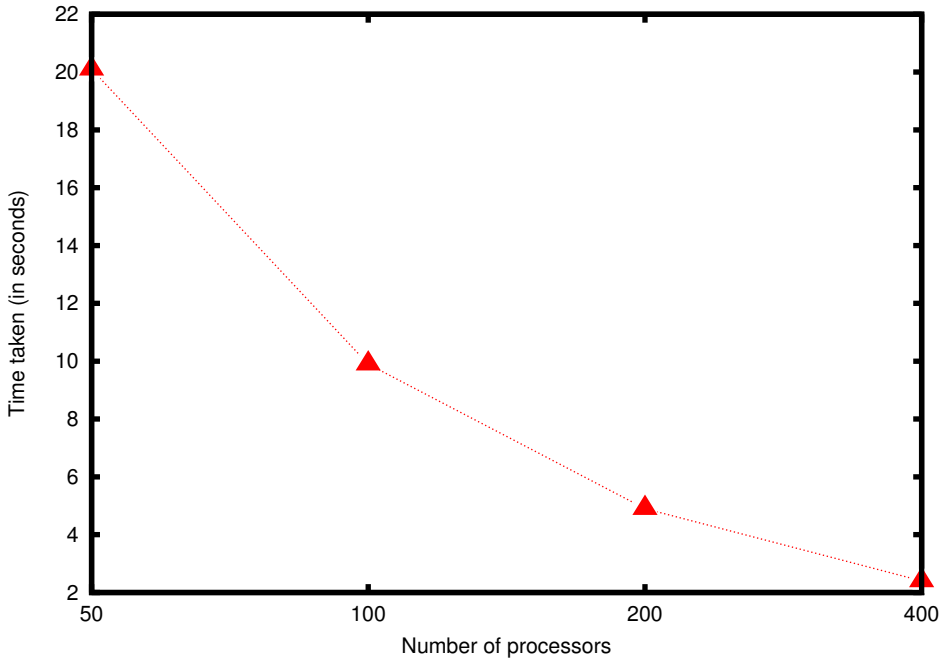

Figure 4.6: Strong scaling of our in-house coupled solver. The time taken for a single coupling step (encompassing multiple DSMC and LB substeps within) plotted against the number of processors

Similarly, we compare the time taken by our in-house DSMC solver and the coupled DSMC-LB solver at two different Reynolds numbers. Fig.(4.7) shows this comparison in log

scale. One can see that there is a significant saving in using a coupled solver against a full-domain DSMC, yet still being able to reproduce correct hydrodynamics. As mentioned earlier, this drastic reduction is a result of the rate-determining steps governed by the DSMC region in the coupled solver scaling as $\mathcal{O}(\mathrm{Kn}^{-2})$ (due to the fixed number of cells along the DSMC wall layer irrespective of the Knudsen number) against $\mathcal{O}(\mathrm{Kn}^{-3})$ for a pure DSMC solver.

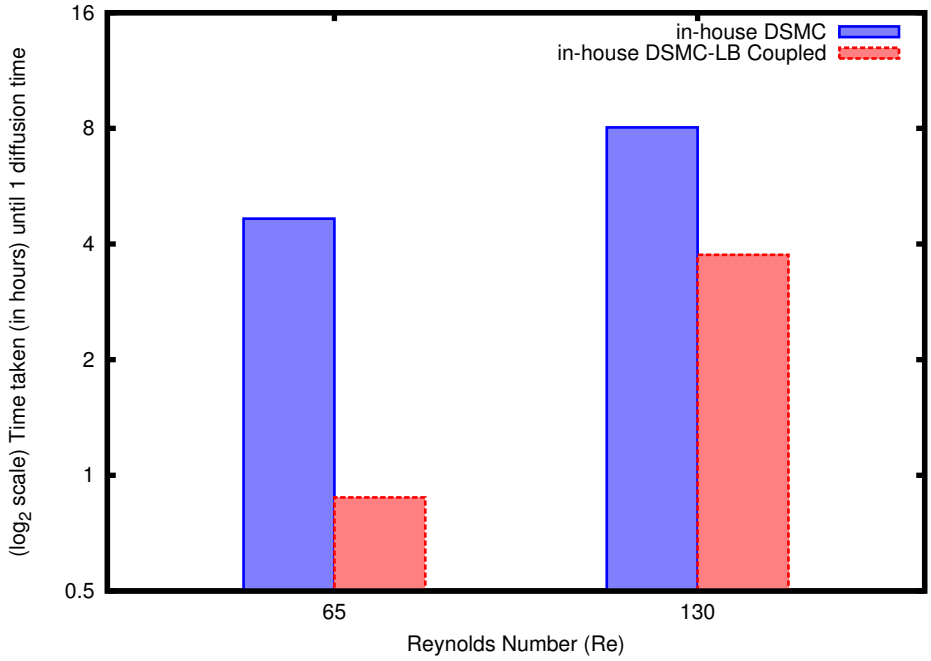

Figure 4.7: A log scale plot of the time taken by our pure DSMC and the coupled DSMC-LB solver to simulate flows at different Re. Simulations performed using 400 processors

4.4 Results

In this section, we present the results obtained from our coupled solver in the continuum, transitional and turbulent regimes. We have chosen the flow between two parallel plates as our simulation setup.

As mentioned earlier, we have restricted our DSMC layer to a distance of $\approx 4\lambda$ from either walls, beyond which our LB solver covers the rest of the domain. As a consequence of this, the LB nodes closest to the walls receive no information directly from the wall and rely on the information passed on from the DSMC cells farthest from the wall.

With the use of a thermal equilibrium LB model, our temperature is allowed to fluctuate about $\theta_0 = 0.294896$, the lattice temperature obtained from the RD3Q41 lattice model (Kolluru *et al.*, 2020). And our DSMC region is therefore initialized with this temperature. Fluctuations arising from the DSMC region are smoothened out using spatial averaging over a region of $5\lambda \times 5\lambda$ in the streamwise and spanwise directions and are also temporally averaged over a timescale, $t_{\mathrm{avg}} \approx 100\tau$.

All simulations were performed in the incompressible regime with a maximum Mach number of 0.2. The Reynolds number in all cases shown being defined as $\mathrm{Re} = U_c L/\nu$, where U_c, L and ν are the centerline velocity, full channel height and the kinematic viscosity respectively.

4.4.1 Continuum regime

Here, we discuss our numerical results for the canonical poiseuille (see Fig.(3.16)) and couette (see Fig.(3.19)) flow cases with moderate Reynolds numbers of $\mathrm{Re} \approx 100$. The Knudsen number

corresponding to this limit being Kn $\approx 10^{-3}$ for a Mach number of 0.2.

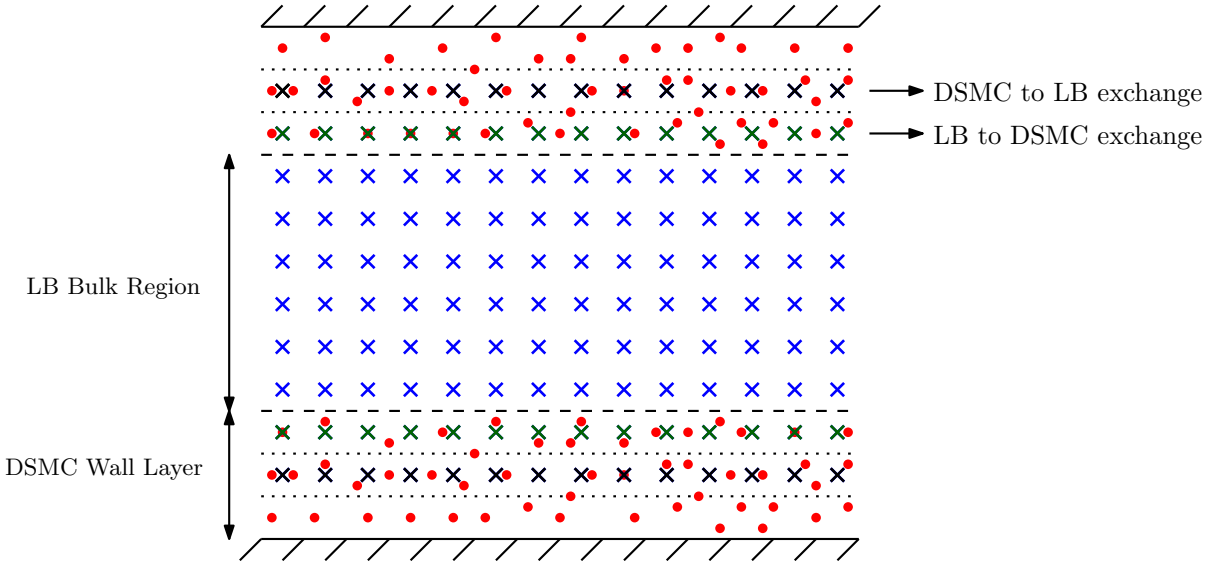

Figure 4.8: A schematic showing the layout of the LB nodes (shown as cross marks) and the DSMC region (particles shown as red dots). Two separate regions for LB to DSMC and DSMC to LB exchange are shown. The LB region does not directly receive any information from the wall. A DSMC layer a few mean free path away from the wall communicates information to the LB nodes.

The simulation was performed using $400 \times 8 \times 40$ (Streamwise $\times$ Wall-normal $\times$ Spanwise) cells in the top and bottom DSMC regions combined, with $n_0 = 100$ particles in each cell. The LB region consisted of $40 \times 40 \times 4$ points in the bulk. With $\Delta y_{\mathrm{LB}}/\Delta y_{\mathrm{DSMC}} = 2$, we interpolate the moments from the first two LB nodes along this direction and use it for particle generation in the LB to DSMC exchange step. Moments are also averaged over 100 DSMC cells before being passed on to the corresponding LB node in the buffer layer. Similarly, the moments received from LB are regenerated in a region consisting of 100 DSMC cells using Algorithm 2. With DSMC being the rate determining process with 128 Million particles in the wall layer, each simulation took approximately 800 CPU hours on an Intel XEON 8268 (PARAM Yukti cluster at JNCASR) to reach a steady state.

The transient analytical (Papanastasiou *et al.*, 2021) velocity profile for an acceleration driven plane channel flow in the continuum regime at any arbitrary time t is,

$$u_x(y,t) = -\frac{g_x L^2}{8\nu}\left[1 - \left(\frac{2y}{L}\right)^2 - \frac{32}{\pi^3}\sum_{k=1}^{\infty}\frac{(-1)^{k+1}}{(2k-1)^3}\cos\left(\frac{(2k-1)\pi y}{L}\right)\exp\left\{-\frac{(2k-1)^2\pi^2\nu t}{L^2}\right\}\right], \tag{4.12}$$

where g_x is the streamwise component of the acceleration.

Figure 4.9 shows the developing velocity profile from our numerical simulation for Re $\approx$ 132, Ma = 0.2 and compares with the analytical profile from Equation (4.12). Here $\bar{t} = t\nu/L^2$ represents the nondimensional time. The steady state velocity profile also shows an excellent agreement with ref. (Yudistiawan *et al.*, 2010).

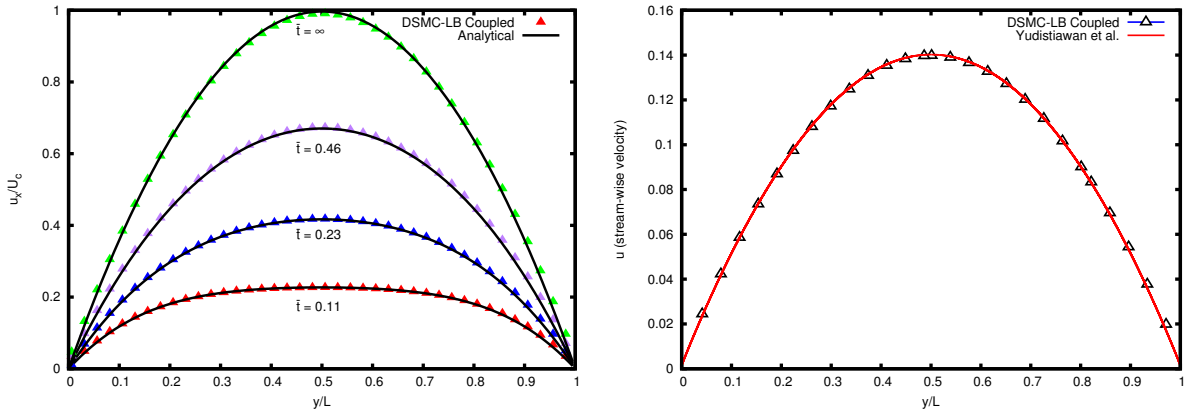

Figure 4.9: Developing Re ≈ 132, Ma = 0.2 result compared with Equation (4.12) and Steady state profile against (Yudistiawan *et al.*, 2010)

Similarly, Figure 4.10 compares the transient shear stress against the analytical derived from Equation (4.12). The numerical value of the wall shear stress, σ^0_{xy} is of $\mathcal{O}(10^{-6})$. This has made the fluctuations in the higher order moments near the wall to be significant. This is the result of the Mach number near the wall being low due to the small magnitude of the velocity near the wall. Therefore, to achieve a higher level of accuracy in the shear stress profile, one needs to use more particles in the near wall region.

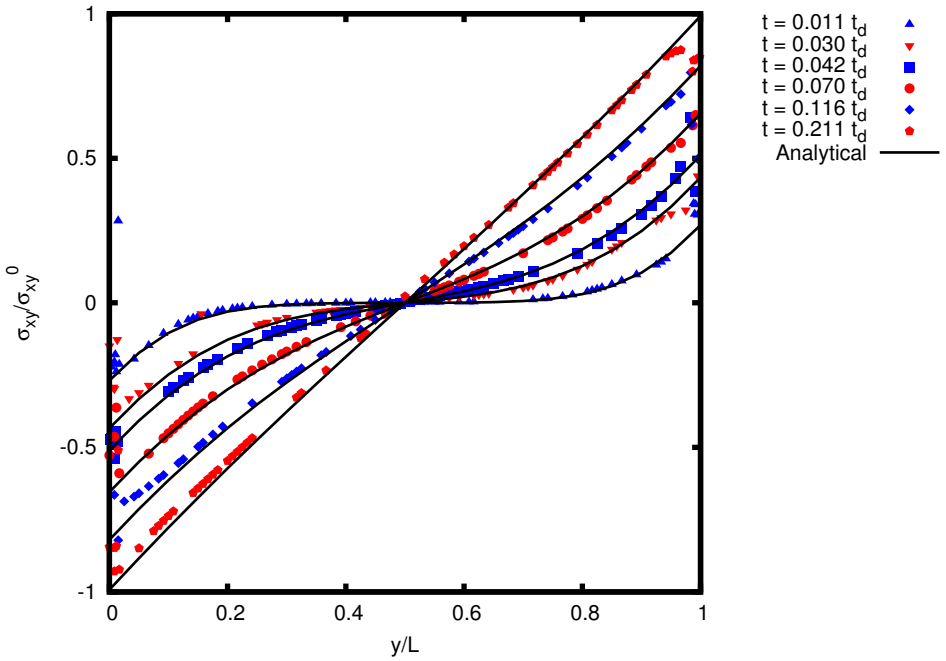

Figure 4.10: Nondimensionalized transient Shear stress profile obtained from the coupled solver at various diffusion times compared against the analytical

We now benchmark the coupled solver on the plane Couette flow geometry in the laminar regime for the same set of nondimensional and geometrical parameters as used in the plane Poiseuille flow setup. A sketch of the geometry is shown in Fig.(3.19).

The analytical (Leal, 2007) velocity profile for plane Couette flow in the laminar no-slip regime at any arbitrary time t is as follows,

$$u_x(y,t) = \left(1 - \frac{y}{L}\right) - \sum_{k=1}^{\infty} \frac{2}{k\pi} \exp\left(\frac{-k^2\pi^2 t\nu}{L^2}\right) \sin\left(\frac{k\pi y}{L}\right). \tag{4.13}$$

In Figure 4.11, we see that the transient velocity profiles in a Couette flow are in very good agreement to the analytical solution from Equation (4.13).

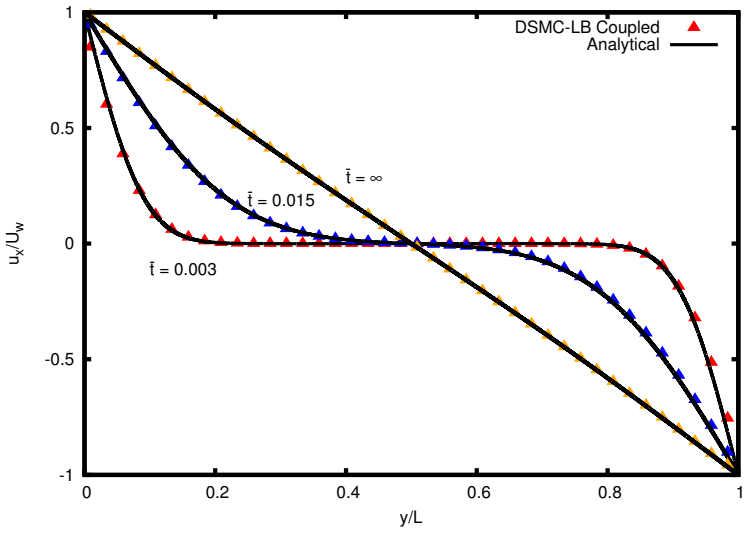

Figure 4.11: Transient velocity profile at various diffusion times compared against analytical(Equation (4.13)) for a planar Couette flow at Re $\approx$ 132, Ma = 0.2

4.4.2 Transitional flow regime

Finite Knudsen number flow dynamics display a dependence on higher order moments and any LB scheme would require a high symmetry of the lattice to satisfy the isotropy conditions. With the increase in the number of discrete velocities, one needs to ensure that the dynamics of the higher order moments is correct at least at the leading order, with the most critical higher order moments bring the stress ($\sigma_{\alpha\beta}$), the heat flux (q_α) and the flux of heat flux ($R_{\alpha\beta}$). The dynamics of these moments lead to prnounced effects in the Knudsen boundary layer. Therefore, one would hope that with a DSMC layer characterizing the boundary layer, the dynamics of these moments will be captured accurately. As a consequence, particle methods are still the preferred choice in the rarefied regime.

As methods like DSMC and MD do not suffer from this lack of isotropy, wall effects can be efficiently transported in such schemes. Hence, having a wall layer DSMC in our solver accounts for efficient thermalization with the wall, while also transporting these effects into the bulk via a moment exchange mechanism. A coupled solver could therefore be used in tandem with lattice models with lower-order isotropy while also avoiding the need to use different boundary schemes for flows in the slip regime.

Limitations arising at such regimes in the Lattice Boltzmann method can be seen in Figure 4.12, wherein at progressively higher Mach and Knudsen numbers, the deviations from the approximate solution (Yudistiawan *et al.*, 2010) increases.

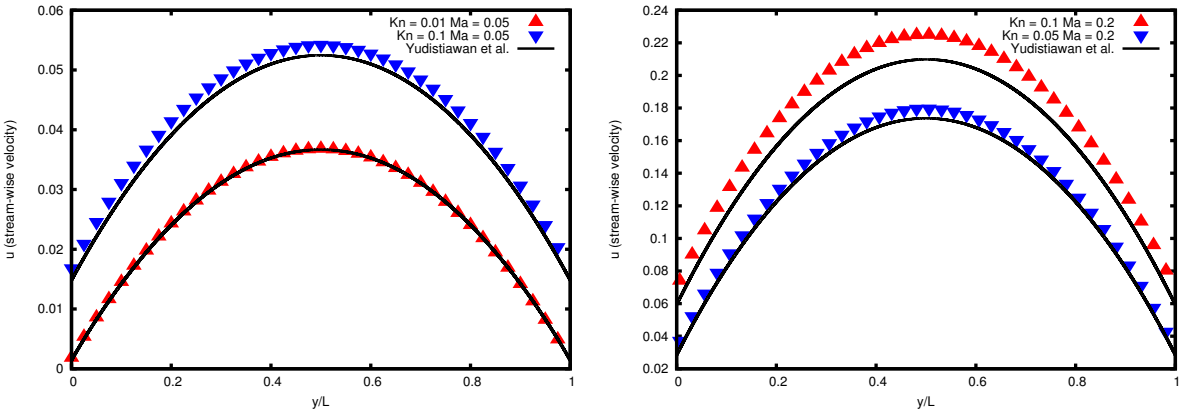

Figure 4.12: Steady state velocity (u_x) profiles for a poiseuille flow at various Knudsen and Mach numbers using an LBM RD3Q41 model (Kolluru *et al.*, 2020)

Although, the computational cost associated with the coupled solver is higher, our simulations at Kn = 0.1 for plane Poiseuille and Couette flows took approximately 200 CPU hours with the same grid size as used for the cases in the continuum regime.

Plane Couette flow simulation at Kn = 0.1 was performed and the steady state velocity profile is in good agreement with (Yudistiawan *et al.*, 2010). Figure 4.13 illustrates the same. The red and blue indicators mark the DSMC and LB regions of the domain respectively. One can observe that the overlap regions between them connects seamlessly.

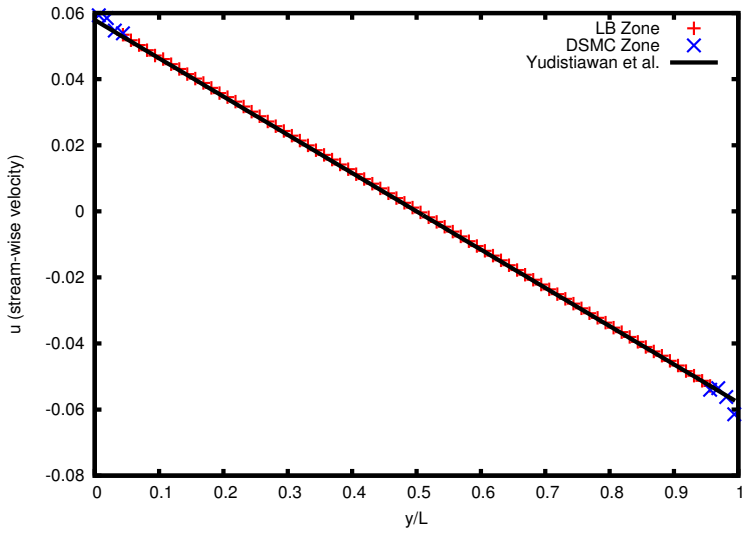

Figure 4.13: Steady state velocity (u_x) profile for a couette flow at Kn = 0.1

Figure 4.14 compares the steady state velocity and shear stress profiles against their analytical solutions for an acceleration driven channel flow at Kn = 0.1 and Ma = 0.2. Here the higher order moment, σ_{xy} shows a good match with the theoretical value.

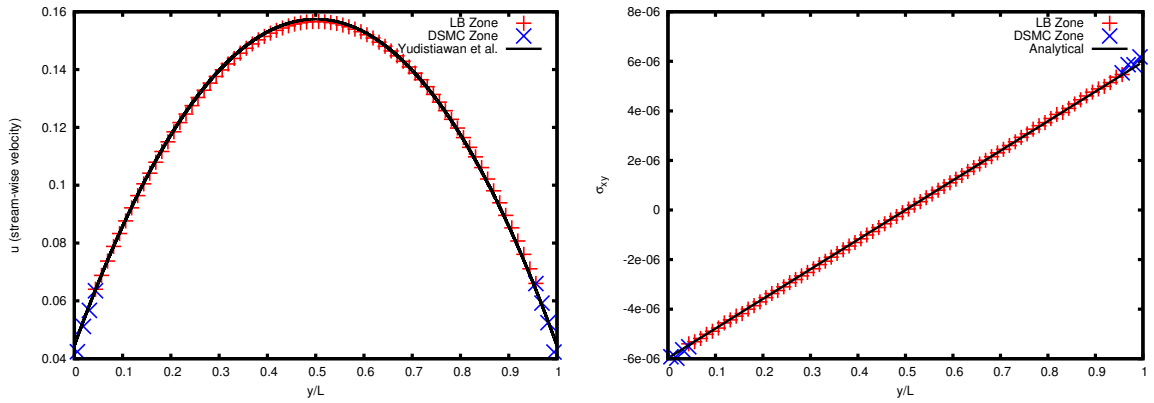

Figure 4.14: Steady state velocity (u_x) and Shear stress (σ_{xy}) profiles for a poiseuille flow at Kn = 0.1

Higher order moments ($\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$) as shown in Figure 4.15 show a smooth transition in the coupling region, with deviations in the DSMC region against the bulk arising at $\mathcal{O}(10^{-6})$. Meanwhile the temperature profile shows the continuous profile of the coupling and the relative cooling of the bulk as compared to the walls.

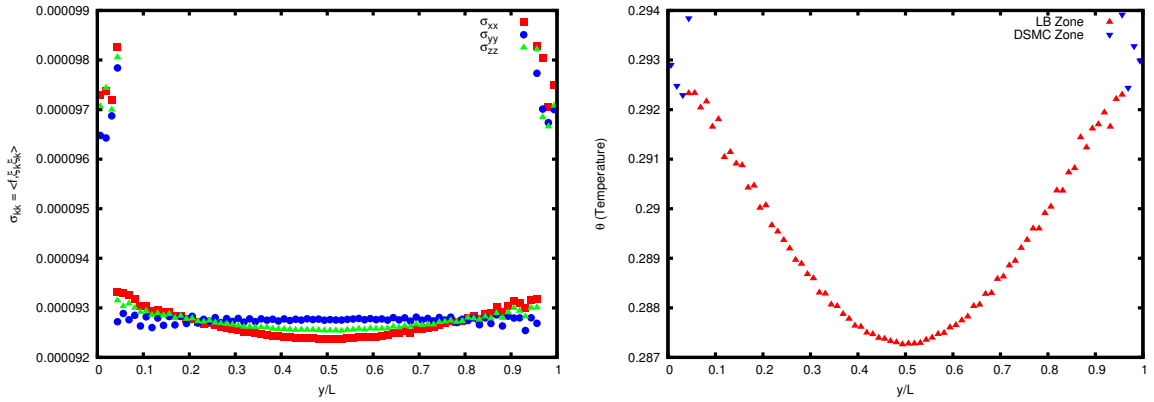

Figure 4.15: Steady state Stress Tensor components ($\sigma_{xx}, \sigma_{yy}, \sigma_{zz}$) and temperature ($\theta$) profiles for a poiseuille flow at Kn = 0.1

4.4.3 Turbulent plane Couette flow

At the fully developed state, the velocity profile in plane Couette flow is monotonic in the Laminar and Turbulent regimes. Linear Stability theory predicts that two-dimensional disturbances are damped for plane couette and pipe flows (Wasow, 1953; Gallagher & Mercer, 1962). Consequently their critical* Reynolds number is ∞. Although experiments have shown that for large enough initial disturbances, the flow transitions to turbulence subcritically (Davies & White, 1928; Kao & Park, 1970; Patel & Head, 1969).

The first known experiments on this setup were conducted by Reichardt(Reichardt, 1956). His further investigations (Reichardt, 1959) on turbulent plane Couette flow demonstrated the charateristic S-shape of the mean velocity profile. He also tried to determine the transition Reynolds number and arrived at Re = 750 (based on half the channel height and the average relative velocity of the walls). Later experiments and simulations have tried to obtain this threshold and the transitional Reynolds number was established to be around 360.

*The critical Reynolds number is defined to be the threshold beyond which infinitesimal perturbations to the flow grow exponentially, whereas the transition Reynolds number is the lowest Re for which transition is observed when a strong enough disturbance is applied

Experimental (Tillmark & Alfredsson, 1992) and numerical (Lundbladh & Johansson, 1991) studies performed to determine the transitional Re have shown excellent agreement, with the former predicting at 360 ± 10, while the latter at 375. Both studies have shown the transformation of an initial elliptical turbulent spot into a circular form. Also, the experimental studies have observed the turbulent state to be spatially intermittent, atleast for such low Reynolds numbers. Based on these, one can say that plane Couette flow transitions at Re $\approx$ 375 when the initial flow field is disturbed using a finite-amplitude perturbation.

Orszag & Kells (Orszag & Kells, 1978) performed simulations to determine the nature of two and three-dimensional finite amplitude disturbances that lead to transition at subcritical reynolds numbers in plane Poiseuille and Couette flows. They found that three-dimensional disturbances with an amplitude 5-10% of the mean velocity drives transition at reynolds numbers of 1000 in plane Couette flow. They also reported that two-dimensional perturbations do not seem strong even at Reynolds numbers an order of magnitude larger.

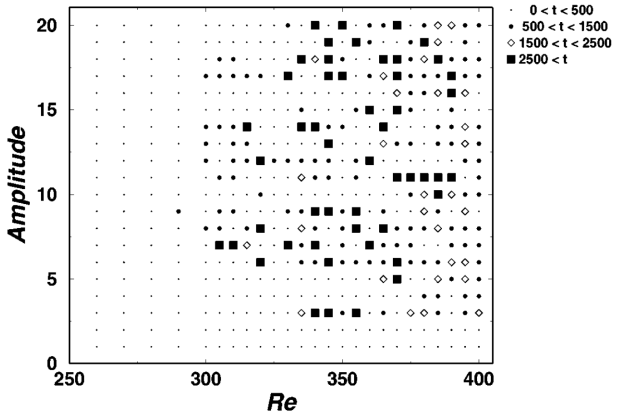

Figure 4.16: Perturbation lifetime as a function of Re and Amplitude (Schmiegel & Eckhardt, 1997)

Further investigations into the perturbation amplitude required for subcritical transition in plane Couette flow have shown that the distribution of the time taken for an initial amplitude to decay in the Amplitude vs Re plane has a fractal structure (Schmiegel & Eckhardt, 1997) (Fig.(4.16)). They shows that the lifetime of an initial perturbation in the Re vs Amplitude plane does not have a functional dependence. They have also shown that successive magnifications of the Re and Amplitude intervals (keeping either one constant) shows a self-similar distribution of lifetimes (Fig.(4.17)). This shows the sensitive dependence on the initial disturbance that can cause transition in such flows.

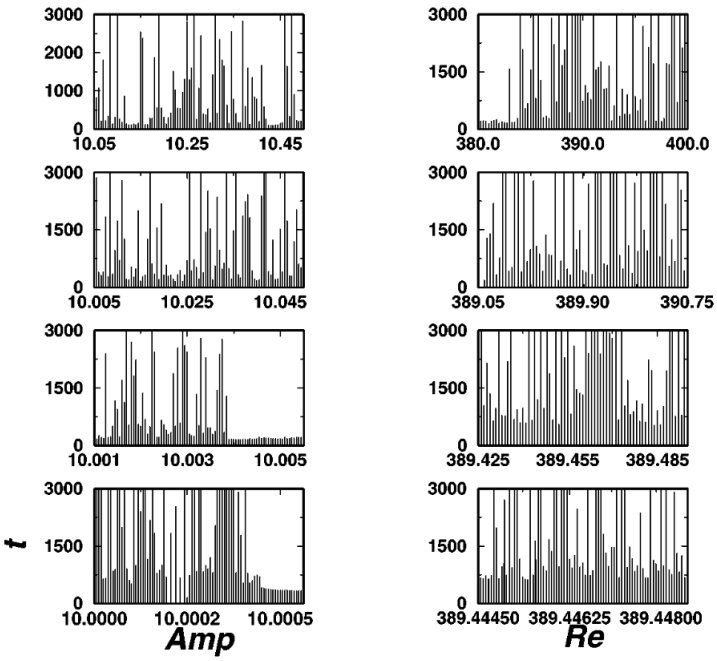

Figure 4.17: Successive magnification of the Re and Amplitude axis against the lifetime of disturbances (Schmiegel & Eckhardt, 1997). The sensitive dependence on the Amplitude and Reynolds number is exemplified here.

Over 5000 experimental and 2000 numerical studies performed on a pipe flow has shown that the turbulent state in such linearly stable shear flows belongs to a transient turbulent chaotic saddle (Hof *et al.*, 2006). The implication of this being that rather than a critical Re beyond which turbulence is believed to sustain indefinitely, there exists a time scale $\tau(\mathrm{Re})$ defining the lifetime of a turbulent state. This was also what was pointed by (Schmiegel & Eckhardt, 1997) for plane Couette flows.

Based on these observations for turbulent plane Couette flows, a few questions still need answers. The influence of the wall micro-roughness on the transition process is still unclear. Using a coupled solver with the DSMC layer providing a suitable and alternative boundary condition to the micro-roughness of the wall, one can attempt to probe such a problem. Similarly, the transition Reynolds number for highly compressible flows is something that can be looked into. Our attempt at benchmarking a case of the turbulent Couette flow is an initial step in this direction.

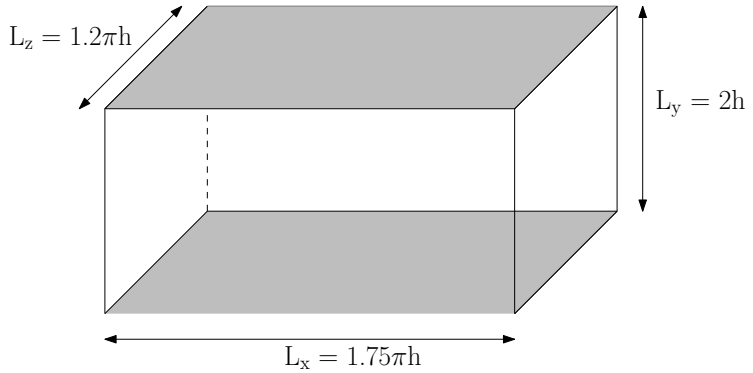

Figure 4.18: Minimal Couette flow geometry

We performed turbulent flow simulation of the minimal Couette flow setup (see Figure

4.18) at a Reynolds number of Re $\approx$ 1318 (based on channel full-width) and Ma = 0.2. The DSMC region consisted of $720 \times 8 \times 1200$ cells with $n_0 = 200$ particles per cell with a total of $\approx$ 1.4 Billion particles in the top and bottom DSMC layers combined. The memory requirement of the DSMC region is therefore $\approx$ 100 GB. The LB region consisted of $720 \times 200 \times 400$ nodes in the bulk. The solver took 120 hours using 2400 cores to reach 30 convection times (L/U_c). Beyond which averaging was done over another 10 convection times. Similar simulations using the coupled and a pure LB solver were performed, with the pure LB solver relaminarizing after a few convection times. The coupled solver in this case helps is capturing the slow regeneration cycles in turbulent MCF.

The domain was initially disturbed using a finite-amplitude perturbation generated using a curl-based Perlin noise (Bridson *et al.*, 2007). Perlin noise (Perlin, 1985) is a method for procedural noise generation that calculates the required noise which takes the gradient of a scalar/vector field as input.

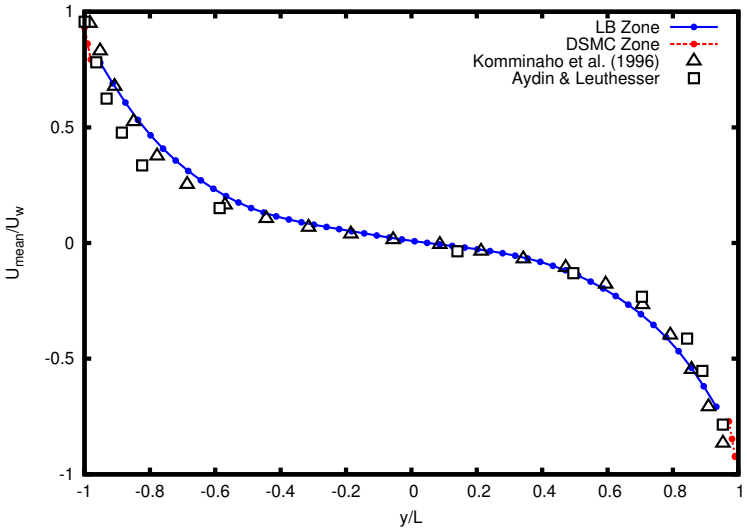

Figure 4.19: Turbulent steady state velocity profile from the coupled compared against the numerical profile from (Komminaho *et al.*, 1996) and experimental profile from (Aydin & Leutheusser, 1979)

Figure 4.19 compares the mean velocity obtained after spatial averaging along the span-wise and stream-wise directions as well as time averaged over 10 convection times. The numerical profile from (Komminaho *et al.*, 1996) have performed an averaging over 620 convection times. As there is constant decay and regeneration in the near wall region in turbulent Couette flow, an averaging over a timescale larger than the regeneration time scale would be required. We still observe a good agreement in the mean streamwise velocity profile between the DSMC-LB coupled solver and the profiles from (Komminaho *et al.*, 1996) and (Aydin & Leutheusser, 1979) near the channel mid-plane. The deviations near the wall result due to the noise from the DSMC solver in the region. We performed similar simulations using $n_0 = 30$ to 100 and observed the averaged velocity profile to deviate further as we lower the number of particles per cell.

Figure 4.20: Streamwise velocity (u_x) field for a DSMC-LB coupled Minimal Turbulent Couette flow (taken after 20 convection times) compared against an LB simulation with the same parameters

Figure 4.20 presents a visualization of the streamwise velocity (u_x) field taken after 20 convection times and compared against a corresponding LB simulation at the same parameters. One can see the presence of coherent structures in both simulations. The LB velocity profile is much smoother as compared to its DSMC-LB counterpart and has resulted in the LB simulation to relaminarize after $\sim$ 50 convection times. For linearly stable flows, finite lifetimes have been observed in similar studies on pipe flows (Hof *et al.*, 2006). A further perturbation to the relaminarized state leads to the cycles of regeneration and decay. As the DSMC layer injects thermal noise into the bulk LB, such a relaminarization could take longer to achieve.

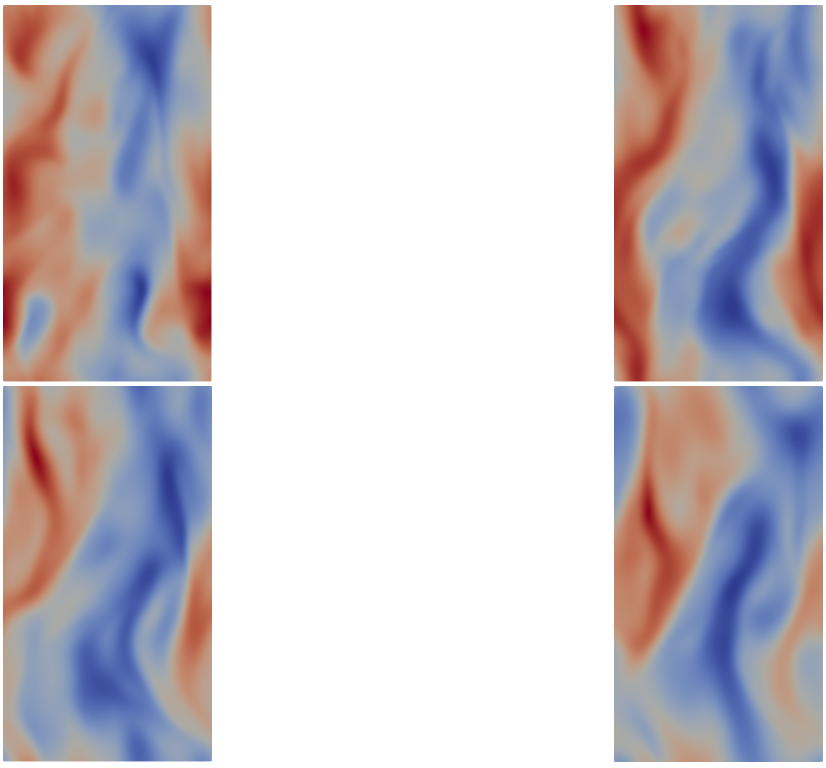

Figure 4.21: Cycle of Regeneration and decay of coherent structures in Minimal Couette flow at the channel mid-plane. Snapshots taken at 14, 20, 22 and 25 convection times respectively

Figure 4.21 shows the streamwise velocity profiles at the channel mid-plane taken at various convection times. We see long coherent structures that undergo cycles of regeneration and decay indicating the sustenance of turbulence using the coupled solver. As observed in similar studies

(Gallis *et al.*, 2018), we observe two vortices which occupy half the spanwise width with anti-parallel orientation along the streamwise direction. Where ref. (Gallis *et al.*, 2018) performed full-domain DSMC simulations using almost 600 Million CPU hours, our coupled solver has achieved a similar feat, yet on a smaller scale, using 300000 CPU hours. Although Pure LB simulations take much lower time, to study the effect of sustained turbulence, relaminarization in LB simulations are a hurdle. One needs to provide frequent disturbances to the LB simulation for sustaining turbulence. In such aspects, one can look to a coupled DSMC-LB solver to study the effects of turbulence in linearly stable shear flows, while also providing accuracy on par with fully microscopic simulations (Gallis *et al.*, 2018).

4.5 Outlook

In this chapter, we have shown how one could couple particle and mesoscopic solvers. We have demonstrated how the grad hermite expansion is used as a projection scheme in communicating between the two solvers and how the spatial and temporal averaging is used to reduce thermal noise from DSMC. The coupled solver was tested in various hydrodynamic regimes. The efficiency of the solver in the moderate Knudsen number regime against standard LBM solvers was shown. In a first of its kind, turbulent flow simulations were performed using a micro-meso solver using extremely low computational resources. Therefore, the coupled solver could serve as an alternative to conventional solvers that require a faithful reproduction of the kinetic equation through its particle wall layer.

Chapter 5
Summary and Conclusions

A hybrid DSMC - Lattice Boltzmann model was developed as an alternative to conventional Navier-Stokes solvers such as Finite volume, etc. and other mesoscopic solvers based on the discretized Boltzmann equation. Due to the large computational expense of particle simulations for fluid flows (with the cost scaling as Kn^{-3}), a full representation of the physical domain using such solvers is infeasible with modest computing resources. Similarly, mesoscopic solvers such as Lattice Boltzmann (LBM) known for their parallel efficiency and for their accurate representation of flows in the low to moderate Knudsen number regime are a preferred choice for such flows. Standard LBM models with their restricted velocity sets fail to provide a faithful representation of hydrodynamics in the rarefied regime. Meanwhile, particle methods due to their high order of isotropy provide a good description of rarefied flows. In this sense, one can therefore look to couple the two methods to mask the shortcomings of either methods while also to be able to use the positives to our advantage.

Our coupled DSMC - LBM solver uses a wall layer DSMC region extendable to accomodate the growing Knudsen layer for flows in the rarefied regime. This hybrid scheme utilizes the ability of the particle nature near the wall to accurately capture non-equilibrium effects near the wall while also retaining the parallel efficiency of the LBM solver occupying the bulk of the fluid domain. We have shown the communication routine between the two solvers which is based on the projection of the the macroscopic field variables into a Grad hermite polynomial expansion. The hybrid solver was then benchmarked on canonical plane Poiseuille and Couette flows in the continuum, rarefied and in the turbulent flow regimes. The solver has also demonstrated good scaling behavior and the computational gain in using a full-domain DSMC was drastic against using the coupled scheme.

In the future, we would like to probe the influence of the wall layer on the transition process for small disturbances. This would address the effect of wall micro-roughness on the transition of plane Channel flows. One can then determine the transition Reynolds number in such flows from a kinetic description. Similarly in the rarefied regime, the effect of a growing Knudsen layer can be probed through the use of a coupled solver. Weak compressibility effects, both in the laminar and turbulent regimes can be addressed with the presence of a DSMC layer. The Mach number dependence on the transitional Reynolds number could be characterized through such simulations.

References

Adhikari, R., Stratford, K., Cates, M. & Wagner, A. 2005 Fluctuating lattice boltzmann. *EPL (Europhysics Letters)* **71** (3), 473.

Agrawal, S., Bhattacharya, S. & Ansumali, S. 2018 Molecular dice: Random number generators á la boltzmann. *Physical Review E* **98** (6), 063315.

Al-Mohssen, H. A., Hadjiconstantinou, N. G. & Kevrekidis, I. G. 2007 Acceleration methods for coarse-grained numerical solution of the boltzmann equation .

Alder, B. J. & Wainwright, T. E. 1959 Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics* **31** (2), 459–466.

Alexander, F. J. & Garcia, A. L. 1997 The direct simulation monte carlo method. *Computers in Physics* **11** (6), 588–593.

Alexander, F. J., Garcia, A. L. & Alder, B. J. 1998 Cell size dependence of transport coefficients in stochastic particle algorithms. *Physics of Fluids* **10** (6), 1540–1542.

AN, K. 1941 Dissipation of energy in locally isotropic turbulence. In *CR Dokl Acad Sci USSR*, , vol. 32, pp. 16–18.

Ansumali, S., Karlin, I., Arcidiacono, S., Abbas, A. & Prasianakis, N. 2007 Hydrodynamics beyond navier-stokes: Exact solution to the lattice boltzmann hierarchy. *Physical review letters* **98** (12), 124502.

Atif, M., Namburi, M. & Ansumali, S. 2018 Higher-order lattice boltzmann model for thermohydrodynamics. *Physical Review E* **98** (5), 053311.

Aydin, M. & Leutheusser, H. J. 1979 Novel experimental facility for the study of plane couette flow. *Review of Scientific Instruments* **50** (11), 1362–1366.

Batchelor, G. K. 2000 *An Introduction to Fluid Dynamics*. Cambridge University Press.

Bird, G. 1963 Approach to translational equilibrium in a rigid sphere gas. *The Physics of Fluids* **6** (10), 1518–1519.

Bird, G. 1970 Direct simulation and the boltzmann equation. *The Physics of Fluids* **13** (11), 2676–2681.

Bird, G. A. & Brady, J. 1994 *Molecular gas dynamics and the direct simulation of gas flows*, , vol. 5. Clarendon press Oxford.

Bridson, R., Houriham, J. & Nordenstam, M. 2007 Curl-noise for procedural fluid flow. *ACM Transactions on Graphics (ToG)* **26** (3), 46–es.

Callen, H. B. 1998 Thermodynamics and an introduction to thermostatistics.

Cercignani, C. 1975 *Theory and application of the Boltzmann equation*. Scottish Academic Press.

Chapman, S. & Cowling, T. G. 1990 *The mathematical theory of non-uniform gases: an account of the kinetic theory of viscosity, thermal conduction and diffusion in gases*. Cambridge university press.

Chen, S. & Doolen, G. D. 1998 Lattice boltzmann method for fluid flows. *Annual review of fluid mechanics* **30** (1), 329–364.

Chikatamarla, S., Ansumali, S. & Karlin, I. 2006 Grad's approximation for missing data in lattice boltzmann simulations. *EPL (Europhysics Letters)* **74** (2), 215.

Davies, S. & White, C. 1928 An experimental study of the flow of water in pipes of rectangular section. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* **119** (781), 92–107.

Di Staso, G., Clercx, H., Succi, S. & Toschi, F. 2016 Dsmc–lbm mapping scheme for rarefied and non-rarefied gas flows. *Journal of Computational Science* **17**, 357–369.

Frapolli, N., Chikatamarla, S. S. & Karlin, I. V. 2015 Entropic lattice boltzmann model for compressible flows. *Physical Review E* **92** (6), 061301.

Gallagher, A. & Mercer, A. M. 1962 On the behaviour of small disturbances in plane couette flow. *Journal of Fluid Mechanics* **13** (1), 91–100.

Gallis, M., Bitter, N., Koehler, T., Torczynski, J., Plimpton, S. & Papadakis, G. 2017 Molecular-level simulations of turbulence and its decay. *Physical Review Letters* **118** (6), 064501.

Gallis, M., Torczynski, J., Krygier, M., Bitter, N. & Plimpton, S. 2021 Turbulence at the edge of continuum. *Physical Review Fluids* **6** (1), 013401.

Gallis, M. A., Torczynski, J. R., Bitter, N. P., Koehler, T. P., Plimpton, S. J. & Papadakis, G. 2018 Gas-kinetic simulation of sustained turbulence in minimal couette flow. *Physical Review Fluids* **3** (7), 071402.

Gallis, M. A., Torczynski, J. R., Plimpton, S. J., Rader, D. J. & Koehler, T. 2014 Direct simulation monte carlo: The quest for speed. In *AIP Conference Proceedings*, , vol. 1628, pp. 27–36. American Institute of Physics.

Garcia, A. L. & Alder, B. J. 1998 Generation of the chapman–enskog distribution. *Journal of computational physics* **140** (1), 66–70.

Garcia, A. L. & Wagner, W. 2000 Time step truncation error in direct simulation monte carlo. *Physics of Fluids* **12** (10), 2621–2633.

Grad, H. 1949 On the kinetic theory of rarefied gases. *Communications on pure and applied mathematics* **2** (4), 331–407.

He, X., Shan, X. & Doolen, G. D. 1998 Discrete boltzmann equation model for nonideal gases. *Physical Review E* **57** (1), R13.

Hof, B., Westerweel, J., Schneider, T. M. & Eckhardt, B. 2006 Finite lifetime of turbulence in shear flows. *Nature* **443** (7107), 59–62.

Kao, T. W. & Park, C. 1970 Experimental investigations of the stability of channel flows. part 1. flow of a single liquid in a rectangular channel. *Journal of Fluid Mechanics* **43** (1), 145–164.

Klothakis, A., Nikolos, I., Koehler, T., Gallis, M. A. & Plimpton, S. J. 2016 Validation simulations of the dsmc code sparta. In *AIP Conference Proceedings*, , vol. 1786, p. 050016. AIP Publishing LLC.

Kolluru, P. K., Atif, M., Namburi, M. & Ansumali, S. 2020 Lattice boltzmann model for weakly compressible flows. *Physical Review E* **101** (1), 013309.

Komminaho, J., Lundbladh, A. & Johansson, A. V. 1996 Very large structures in plane turbulent couette flow. *Journal of Fluid Mechanics* **320**, 259–285.

Leal, L. G. 2007 *Advanced transport phenomena: fluid mechanics and convective transport processes*, , vol. 7. Cambridge University Press.

Lubachevsky, B. D. 1991 How to simulate billiards and similar systems. *Journal of Computational Physics* **94** (2), 255–283.

Lundbladh, A. & Johansson, A. V. 1991 Direct simulation of turbulent spots in plane couette flow. *Journal of Fluid Mechanics* **229**, 499–516.

Montessori, A., Tiribocchi, A., Lauricella, M., Bonaccorso, F. & Succi, S. 2020 A multiresolution mesoscale approach for microscale hydrodynamics. *Advanced Theory and Simulations* **3** (4), 1900250.

Moore, G. E. *et al.* 1965 Cramming more components onto integrated circuits.

Namburi, M., Krithivasan, S. & Ansumali, S. 2016 Crystallographic lattice boltzmann method. *Scientific reports* **6**, 27172.

Nie, X., Doolen, G. D. & Chen, S. 2002 Lattice-boltzmann simulations of fluid flows in mems. *Journal of statistical physics* **107** (1-2), 279–289.

Orszag, S. A. & Kells, L. C. 1978 Transition to turbulence in plane poiseuille and plane couette flow. *Tech. Rep.*. Cambridge Hydrodynamics INC MA.

Papanastasiou, T., Georgiou, G. & Alexandrou, A. N. 2021 *Viscous fluid flow*. CRC press.

Patel, V. & Head, M. 1969 Some observations on skin friction and velocity profiles in fully developed pipe and channel flows. *Journal of Fluid Mechanics* **38** (1), 181–201.

Perlin, K. 1985 An image synthesizer. *ACM Siggraph Computer Graphics* **19** (3), 287–296.

Piekos, E. & Breuer, K. 1995 Dsmc modeling of micromechanical devices. In *30th Thermophysics Conference*, p. 2089.

Pope, S. B. 2001 Turbulent flows.

Pöschel, T. & Schwager, T. 2005 *Computational granular dynamics: models and algorithms*. Springer Science & Business Media.

Pradhan, S. & Kumaran, V. 2016 Transition and turbulence in a wall-bounded channel flow at high mach number. In *52nd AIAA/SAE/ASEE Joint Propulsion Conference*, p. 5050.

Rapaport, D. C. 2004 *The Art of Molecular Dynamics Simulation*, 2nd edn. Cambridge University Press.

Reichardt, H. 1956 Uber die geschwindigkeitsverteilung in einer geradlinigen turbulenten couettestormung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift for Angewandte Mathematik und Mechanik* **36**.

Reichardt, H. 1959 Gesetzmabigkeiten der geradlinigen turbulenten couette stromung. *Max-Planck-Institut fur Stromungsforschung* **22**.

Richardson, L. 1922 Weather prediction by numerical process (cambridge u. press).

Roy, P., Weisgraber, T. H. & Alder, B. J. 2017 The onset of turbulence in wall-bounded flows with surface roughness and fluctuations. In *Advances in the Computational Sciences: Symposium in Honor of Dr Berni Alder's 90th Birthday*, pp. 52–66. World Scientific.

Schmiegel, A. & Eckhardt, B. 1997 Fractal stability border in plane couette flow. *Physical review letters* **79** (26), 5250.

Shan, X. 1997 Simulation of rayleigh-bénard convection using a lattice boltzmann method. *Physical review E* **55** (3), 2780.

Shan, X. & He, X. 1998 Discretization of the velocity space in the solution of the boltzmann equation. *Physical Review Letters* **80** (1), 65.

Shet, A. G., Sorathiya, S. H., Krithivasan, S., Deshpande, A. M., Kaul, B., Sherlekar, S. D. & Ansumali, S. 2013 Data structure and movement for lattice-based simulations. *Physical Review E* **88** (1), 013314.

Succi, S. 2001 *The lattice Boltzmann equation: for fluid dynamics and beyond*. Oxford university press.

Tillmark, N. & Alfredsson, P. H. 1992 Experiments on transition in plane couette flow. *Journal of Fluid Mechanics* **235**, 89–102.

Wasow, W. 1953 On small disturbances of plane couette flow. *J. Res. Nat. Bur. Stand* **51** (4), 195–202.

Yudistiawan, W. P., Kwak, S. K., Patil, D. & Ansumali, S. 2010 Higher-order galilean-invariant lattice boltzmann model for microflows: Single-component gas. *Physical Review E* **82** (4), 046701.

Zhang, W.-M., Meng, G. & Wei, X. 2012 A review on slip models for gas microflows. *Microfluidics and nanofluidics* **13** (6), 845–882.