

Multi-Particle Collision Dynamics

For Fluid Flow Simulations

Submitted by

Ruchika Mahajan

(DI1504)

Work done under the Guidance of

Dr. Dhiraj V. Patil

Submitted in partial fulfilment of the

requirements for the award of

Master of Science in Physics



Indian Institute of Technology Mandi

Kamand, Mandi – 175005

Himachal Pradesh, INDIA.

Certificate

This is to certify that the project entitled

Multi-Particle Collision Dynamics

For Fluid Flow Simulations

is a bonafide work done by

Ruchika Mahajan (DI1504)

under the guidance of

Dr. Dhiraj V. Patil

in partial fulfilment of the requirements for the award of

Master of Science in Physics

by Indian Institute of Technology Mandi

Himachal Pradesh, INDIA.

Project Guide(s)

Faculty Advisor

Chairperson (SBS)

Course Coordinator

Contents

1	Fluid physics at meso-scales	1
1.1	Introduction	1
1.1.1	Computer Simulation	2
1.1.2	Multi-Scales in Fluid Flow	3
1.2	Top Down Approach	6
1.2.1	Navier-Stokes Method	6
1.2.2	Lattice Boltzmann	7
1.3	Bottom Up Approach	8
1.3.1	Molecular Dynamics (MD)	8
1.4	Meso-scale Methods	10
1.4.1	Dissipative Particle Dynamics (DPD)	10
1.4.2	Direct Simulation Monte Carlo (DSMC)	10
1.4.3	Multi-Particle Collision Dynamics (MPCD) or Stochastic Rotation Dynamics (SRD)	11
2	Multi-Particle Collision Dynamics	12
2.1	Algorithms	13
2.1.1	Position and Velocity Initialization	13
2.1.2	Streaming	15
2.1.3	Collision	15
3	Conservation laws	19

<i>CONTENTS</i>	iii
3.1 Energy and Momentum Conservation	19
3.1.1 Analytical approach to the momentum and energy conservation	19
3.1.2 Numerical Approach	21
3.2 Thermostat	21
3.2.1 Local simple scaling	22
4 Transport Coefficient	23
4.1 Viscosity	23
4.1.1 Analytic expression for kinetic Viscosity	23
4.1.2 Analytic expression for collisional Viscosity	26
5 Code Validations	29
5.1 Taylor Green vortex Flow	29
5.2 2D Poiseuille Flow (between two parallel plates)	31
5.2.1 Analytical Approach	31
5.2.2 Numerical Approach	33
6 Conclusion	35

Abstract

The meso-scale simulation method was introduced by (Malevanets and Kapral 1999), is referred as multi-particle collision (MPC) dynamics or stochastic rotation dynamics (SRD) is described [1]. The method consists of alternating streaming and collision steps in an ensemble of point particles. The simulations are performed by grouping point particles in collision cells, with local conservation of mass, momentum, and energy. Collisions are expressed in terms of a redistribution of momentum among them. This simulation technique captures both full hydrodynamic interactions and thermal fluctuations. To maintain a defined thermodynamic state, thermalization of the fluid is often required for certain MPC variants. In this simulation the local velocities are scaled by a factor, which is determined via a local simple scaling approach (LSS). Multi-particle collision dynamics (MPCD) is a particle based Navier-Stokes solver and in the last ten years it has been largely used to analyze meso-scopic systems where both hydrodynamics and thermal effects have to be taken into account, typical examples being colloidal suspensions, polymer solutions, Poiseuille flow. The shear viscosity associated with the stochastic rotation model. We use a kinetic theory approach to derive the expressions for the viscosity. The viscosity is divided into two contributions. The kinetic viscosity and the collisional viscosity.

Declaration

This thesis is a presentation of my original work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgment of collaborative research and discussions. The work was done under the guidance of Dr. Dhiraj V. Patil, at the Indian Institute of Technology, Mandi.

Candidate's name and signature

In my capacity as supervisor of the candidate's thesis, I certify that the above statements are true to the best of my knowledge.

Guide's name and signature

Acknowledgements

I would like to express deepest gratitude to my advisor **Dr. Dhiraj V. Patil** for his full support, guidance and encouragement throughout my study and research. Without his incredible patience and timely wisdom counsel, my thesis work would have been an overwhelming pursuit.

Author is grateful to **Prof. R. G. Winkler** and **Mr. S. Das** of Forschungszentrum Julich, Germany, for providing the FORTRAN code of MPCD, **Dr. Mauro Chinappi** of Universita degli studi di Roma Tor Vergata, Italy, **Dr. Apratim Chatterji** of IISER Pune, India, **Dr. Prasanth P Jose** and **Dr. Hari Verma** of IIT Mandi, for constant encouragement, technical discussions and help in MPCD code development.

I also thank **Mr. Amitkumar S. Gawas** and **Ms. Manju Bisht** for making me understand the fluid mechanics basics of my research work, without their help this research work is not possible.

I extend my thanks to **Ms. Swati Mudra**, **Mr. Saravanan Rajendran** and **Mr. Chinmoy Samanta** for their constant support during my stay in IIT Mandi. I would also like to thank all my colleagues at IIT Mandi for their encouragement and moral support.

Finally, I would like to thank my parents, brother, sister, grandparents and my uncle for their unconditional love and support during last two years; I would not have been able to complete this thesis without their continuous love and encouragement.

Chapter 1

Fluid physics at meso-scales

1.1 Introduction

The Focus of this chapter is to present and discuss methods from physics for the continuum, molecular-scales and meso-scales. The meso-scale region is defined as a scale in between the micro-scale and molecular scale. The meso-scale exists to cover the change in physics between the continuum approximated view and the discontinuous molecular description. The discussion begins with the computer simulations where we learn that why computer simulation is important in the areas of physics at different scales. Then, we focus on Knudsen number which determine whether statistical mechanics or the continuum mechanics formulation of fluid dynamics should be used to model a situation. After that, we will discuss top down approach, where simulate fluid as a continuous medium. Existing meso-scales methods will then be discussed, such as dissipative particle dynamics and the lattice Boltzmann method. The meso-scale methods commonly used present a top down approach. The benefit of which and applications will then be discussed, and the final section will discuss the bottom-up approach to looking at schemes for extracting bulk properties from molecular dynamics.

1.1.1 Computer Simulation

Fluid flow physics is one of the most important of all areas of physics. Fluids occur, and often dominate physical phenomena, on all macroscopic (non-quantum) length scales to the micro and even nano-scales e.g biological cell activity. This fluid physics traditionally analyzed using experimental analytical or computation method with the advent of high performance computing and accurate numerical models. It is now possible to accurately predict fluid flow physics using computer simulation. A computer simulation run on a single or a network of computers, to reproduce behavior of a system through appropriate algorithm describing physical phenomena. Simulation allow, physicists to explore numerous systems and geometries at relatively low cost and at various levels of detail. It represents discretization concepts in which the response of a continuum is projected onto particles carrying the mechanical information during deformations. Typical representatives are meshless methods, Smoothed Particles Hydrodynamics (SPH). On the other hand the notion expresses the computational representation of physical particles existing on different scales. Classical versions are Molecular Dynamics (MD) or the Discrete (Distinct) Element Method (DEM). The continuum description based on the Navier-Stokes equation is not sufficient, since, molecular-level details including thermal fluctuations play a central role in determining the dynamic behavior for example atomistic molecular dynamics simulations retain too many microscopic degrees of freedom, consequently requiring very small time steps in order to resolve the high frequency modes. This makes it impossible to study long timescale behavior such as self-assembly and other meso-scale phenomena. In order to overcome these difficulties, considerable effort has been devoted to the development of meso-scale simulation methods such as Dissipative Particle Dynamics, Lattice-Boltzmann, and Direct Simulation Monte Carlo. The common approach of all these methods is to average out irrelevant microscopic details in order to achieve high computational efficiency while keeping the essential features of the microscopic physics on the length scales of interest.

1.1.2 Multi-Scales in Fluid Flow

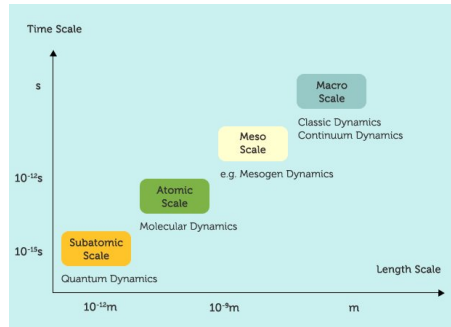


Figure 1.1: Different length and time scales that span various simulation methods.

The length and time scales of applicability of various simulation methods are shown in Fig.1.1. In order to describe the multi-scales in fluid flow we will consider a gas, it is a fluid. Consider a system which has gas molecules. In order to analyze the sys-

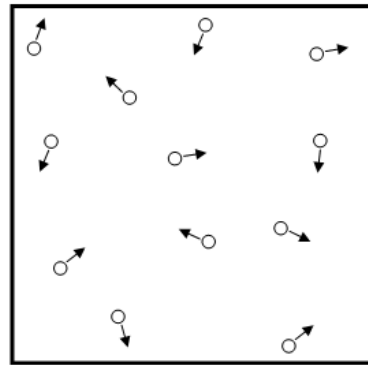


Figure 1.2: $6N$ degrees of freedom

tem. One probability is that we can write the equation of motion for each of these molecules. Each molecules may have three translational degrees of freedom and three rotational degrees of freedom that mean 6 independent equations for each molecule into the number of molecules. So, how many of unknown we have and we will have these number of matching equations of motion and we have to solve for that to get a physical picture of the molecular motion. Hence this is the fundamental way of analyzing the motion and is known as molecular dynamics. But one has to understand that it has a practical limitation that it cannot address a very large system. It can address only a small system with number of molecules not significantly large depending

upon the computational resources it may be 1000 or more. One alternative is that instead of addressing individual molecules we can make a statistically average of many molecules. So, instead of directly simulating the molecules, we statistically represent a group of molecules by statistical property, and that is what we commonly done in the kinetic theory of gases. So, in the kinetic theory of gases, we address the behavior of gas statistically. To assess the validity of the continuum or molecular model for a gas, it is necessary to obtain a measure of the rarefaction of the gas at the scale of interest. To do this, the concept of mean free path is introduced. The mean free path is the average distance traveled by a molecule before it collides/interacts with another molecule. For an ideal hard sphere gas, the mean free path, λ , is a function of pressure, P , and temperature, T , as follows:

$$\lambda = \frac{K_b T}{\sqrt{2} \pi P \sigma^3} \quad (1.1)$$

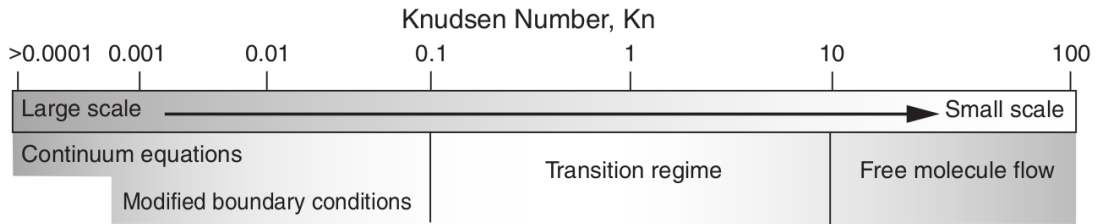


Figure 1.3: Range of Knudsen number for gas systems.

This gives a measure of the rarefaction of the gas. This can then be compared with the characteristic length of the flow field, L , which can either be taken as the characteristic dimension or the gradient of a bulk property, such as density

$$L = \left| \frac{\rho}{d\rho/dx} \right| \quad (1.2)$$

The ratio of the characteristic length to the mean free path of the gas is known as the Knudsen (K_n) number:

$$K_n = \frac{\lambda}{L} \quad (1.3)$$

the value of which is used as a measure of the rarefaction of a gas with respect to the

scale of the system, in order to test the validity of the continuum approximations. Typical values are shown in Fig.1.3 from the large scale continuum to molecular systems. A very small K_n number (< 0.001) describes a system that is well within the continuum laws, but as the K_n number increases the small scale effects of the fluid become more pronounced. The first stage of the breakdown of the continuum approximation occurs at a K_n number of greater than 0.001 where areas of high gradient, such as boundaries, cannot maintain the continuous distribution of macroscopic properties. This is a result of the deviation from thermodynamic equilibrium, where there are insufficient collisions in the system for the energy to propagate smoothly in areas of high gradient such as at the boundaries. The low number of molecular interactions with the boundary means that the velocity and temperature of the solid and fluid are no longer the same at the interface, causing a violation of the no-slip condition that is assumed in continuum mechanics but as the K_n number of the system increases above a value of 0.001, the slip effect becomes more pronounced. The amount of slip that is allowed depends on the roughness of the surface over which the fluid is flowing and the interaction rate between the fluid and solid molecules. For higher K_n numbers, the violation of the continuum laws becomes more serious, as the effect of finite numbers of molecules affects the propagation of macroscopic properties further away from boundaries and wider areas of high gradient. Also the localization of mass and energy at molecular sites starts to bring statistical variations into the fluid properties. This type of breakdown occurs in the transition region between the continuum and molecular regions. A transition region where continuum approximations cannot accurately predict the system behavior is between K_n numbers 0.1 to 10. In this region, the mean free path and characteristic length of the gas are comparable, indicating the importance of the underlying molecular physics of the system. In these systems, the continuum equations cannot be applied, even with boundary modifications. For K_n number greater than 10, the mean free path of the gas is more than 10 times greater than the characteristic length of the system, and the fluid is well within the limits of and can only be described by molecular physics.

1.2 Top Down Approach

Top down simulations typically start with differential equations for the behavior of fluids, and simulate the fluid as a continuous medium. If the domain under consideration is small enough that the fluid is no longer entirely continuous, it does not correctly model it. This type of method is appropriate for many types of fluid but it usually works best at a scale large enough for the fluid to behave deterministically. There are two popular approaches: the Navier-stokes solvers and lattice methods. In what follows, we will briefly review these methods.

1.2.1 Navier-Stokes Method

The Navier-Stokes equations guide the motion of the fluids. It is a vector equation obtained by applying Newton's Law of Motion to a fluid element and is also called the momentum equation. It is supplemented by the mass conservation equation (continuity equation) and the energy equation. The Navier-Stokes equations are as follows.

$$\rho \frac{Du}{Dt} = -\frac{\partial p}{\partial x} + \text{div}(\mu \text{ grad } u) + S_{Mx} \quad (1.4)$$

$$\rho \frac{Dv}{Dt} = -\frac{\partial p}{\partial y} + \text{div}(\mu \text{ grad } v) + S_{My} \quad (1.5)$$

$$\rho \frac{Dw}{Dt} = -\frac{\partial p}{\partial z} + \text{div}(\mu \text{ grad } w) + S_{Mz} \quad (1.6)$$

These three equations are the momentum equations in three dimensions (x, y and z direction). Where ρ is the density of the fluid, p is the pressure, μ is the viscosity, u, v, w are the component of velocity and S_{Mx}, S_{My} , and S_{Mz} are the source terms in x, y , and z direction [5].

$$\rho \frac{Di}{Dt} = -\rho \text{ div } u + \text{div} (K \text{ grad } T) + \phi + S_i \quad (1.7)$$

The energy equation is given by the above equation, where ρ is the density of the fluid, K is the kinetic energy and T is the temperature. The equations are always

coupled with the continuity equation.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \quad (1.8)$$

Solving these equations for a particular set of boundary conditions (such as inlets, outlets, and walls), estimate the fluid velocity and its pressure in a given geometry. Because of their complexity, these equations only admit a limited number of analytical solutions. It is relatively easy, for instance, to solve these equations for a flow between two parallel plates or for the flow in a circular pipe. For more complex geometries, however, the equations need to be solved numerically [4].

1.2.2 Lattice Boltzmann

Lattice Boltzmann methods (LBM) is a class of computational fluid dynamics (CFD) methods for fluid simulation. Instead of solving the Navier–Stokes equations, the discrete Boltzmann equation is solved to simulate the flow of a Newtonian fluid with collision models such as Bhatnagar–Gross–Krook (BGK). By simulating streaming and collision processes across a limited number of particles, the intrinsic particle interactions evince a microcosm of viscous flow behavior applicable across the greater mass. With proper choice of collision rules that conserve momentum and energy, accurate hydrodynamic behavior appear at the macroscopic scale. The lattice requires that both position and velocity are discretized. This technique solves the discrete Boltzmann equation on discrete lattice mesh. The change through lattice gas automata is that lattice Boltzmann utilization molecule likelihood densities as opposed to whole particles. This implies that position and speed nearly constant. In the computer algorithm, the collision and streaming step are defined as follows:

$$f_i^t(\vec{x}, t + \delta t) = f_i(\vec{x}, t) + \frac{1}{\tau}(f_i^{eq} - f_i) \quad (1.9)$$

$$f_i(\vec{x} + \vec{e}_i \delta t, t + \delta t) = f_i^t(\vec{x}, t + \delta t) \quad (1.10)$$

Here i are the direction of the momentum.

1.3 Bottom Up Approach

Alternatively, bottom-up methods can be used to model the behavior of fluids. In this case, the lowest-level physical rules are combined and applied, giving rise to the macroscopic behaviors of fluids. Below, we briefly describe some examples of bottom up techniques.

1.3.1 Molecular Dynamics (MD)

Molecular Dynamics is computational technique that considers spherical fluid molecules that interact through pair-wise Lennard-Jones potentials (interactions are all done as inter-particle potentials). But calculating the force between each pair of particles for each time step is a costly procedure. At times, more coarse-grained models of fluids will suffice. These potentials usually are electrostatic, though they can in theory be anything, such as Lennard-Jones, etc. [2].

In MD simulations we are integrating the Newtons equation of motions numerically with appropriate choice of potential using velocity-verlet algorithm. Velocity-verlet is a numerical method used to integrate Newton's equations of motion and frequently used to calculate trajectories of particles in **(MD)** simulations. The velocity-verlet method of advancing positions and velocities can be obtained as shown below.

$$X_i(t + \Delta t) = X_i(t) + \Delta t V_i + \frac{1}{2}(\Delta t)^2 \frac{F_i}{m_i} \quad (1.11)$$

$$V_i(t + \Delta(t)) = V_i(t) + \frac{1}{2}(\Delta(t)) + \frac{F_i(t) + F(t + \Delta(t))}{m_i} \quad (1.12)$$

In a dilute gas, the molecules are widely separated and the molecules move several (tens) of angstroms before “colliding” with another molecule. In a liquid, the packing is very similar to that of a solid except that the molecules are moving all the time. Consider N number of particles inside a cubical box. These particles or atoms behave approximately like hard spheres which attract one another with weak van der Waals

forces. The forces between two atoms can be approximated quite well by a Lennard-Jones potential energy function.

$$V(r) = 4\varepsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right], \quad (1.13)$$

where ' r ' is the distance between the centers of the two atoms, $\varepsilon = (1.65 \times 10)^{-21} J$ is the strength of the potential energy, and $\sigma = (3.4 \times 10)^{-10} m$ is the value of r at which the energy is zero. The $(1/r)^{12}$ term represents a repulsive hard core interaction between the atoms. The $(1/r)^6$ term represents an attractive dipole-dipole (van der Waals) interaction between the non-polar atoms. The potential has its minimum V . The shape of the potential and the strength of the Lennard-Jones force.

$$F(r) = -\frac{dV(r)}{dr} \quad (1.14)$$

$$F(r) = 24 \frac{\varepsilon}{\sigma} \left[2 \left(\frac{\sigma}{r} \right)^{13} - \left(\frac{\sigma}{r} \right)^7 \right] \quad (1.15)$$

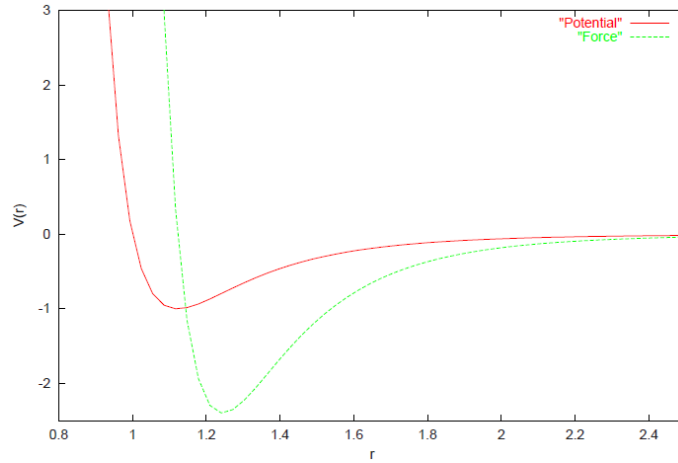


Figure 1.4: Lennard-Jones potential

1.4 Meso-scale Methods

Meso-scale models have some similar properties to molecular dynamics models, but they use particles, which represent a region of the fluid. They do not resolve the full microscopic details, as in molecular dynamics, but they still have fluctuations and produce correct hydrodynamic behavior. In what follows, we will give a few examples of such models.

1.4.1 Dissipative Particle Dynamics (DPD)

Dissipative particle dynamics (DPD) is a stochastic simulation technique for simulating the dynamic and rheological properties of simple and complex fluids. DPD takes after Molecular Dynamics in that the particles move as per Newton's laws, this takes into account the investigation of physical behavior on time scales. It is a particle-based model, with each particle representing a fluid region. The total non-bonded force acting on a DPD particle i is given by a sum over all particles j that lie within a fixed cut-off distance, of three pairwise additive forces [8].

$$f_i = \sum_{i \neq j} (F_{ij}^C + F_{ij}^D + F_{ij}^R), \quad (1.16)$$

where the first term in the above equation is a conservative force, the second a dissipative force and the third a random force. The conservative force acts to give beads a chemical identity, while the dissipative and random forces together form a thermostat that keeps the mean temperature of the system constant. A key property of all of the non-bonded forces is that they conserve momentum locally, so, that hydrodynamic modes of the fluid emerge even for small particle numbers.

1.4.2 Direct Simulation Monte Carlo (DSMC)

Direct Simulation Monte Carlo (DSMC) is the Meso-scale method which models fluids using aggregate representative particles. These particles, instead of modeling a

single atom or simple molecule, are representative of an entire region of the fluid. The model runs in discrete time steps with continuous position and velocity vectors. During each time step, the particles undergo a streaming and a collision step. In the streaming step, the particles advance their positions through time ballistically. In the collision step, the particles collide with each other via a set of collision rules. The system is divided into a set of boxes and only particles within a given box have a chance to collide. These collisions are calculated using probabilistic models, with collisions happening between individual particle pairs. Since, collisions are elastic (unless intentionally set otherwise), DSMC conserves energy and momentum.

1.4.3 Multi-Particle Collision Dynamics (MPCD) or Stochastic Rotation Dynamics (SRD)

Multi-particle collision dynamics (MPC), also known as stochastic rotation dynamics (SRD), is a particle-based meso-scale simulation technique for complex fluids which fully incorporates thermal fluctuations and hydrodynamic interactions [2]. This method is similar in principle to earlier model, however, it uses a different collision scheme. Instead of having randomly chosen particle pairs collide with each other, it collides groups of particles with each other simultaneously. These groups are picked by dividing the simulation domain into a set of boxes (square in 2D and cube in 3D). All the particles within a box collide with each other at once, using the simplest available collision rule that conserves momentum and energy. Since, particle pairs are not considered, the simulation scales much better than DSMC. This means that SRD can simulate larger systems more efficiently. There are several applications of MPCD some of these are simulation of complex system on meso-scale, fluid dynamics of moving fish in a two dimensional [17], conformation and diffusion behavior of ring polymers in solution [13], numerical study of the flow around a cylinder using MPCD [14] and multi-particle collision dynamics of visco-elastic fluids [11].

Chapter 2

Multi-Particle Collision Dynamics

The MPCD simulation is such a meso-scopic technique that can be used to model fluids and it was initially proposed by Malevanets and Kapral in 1999. In MPCD the solvent is modeled by a large number of point-like particles of mass m which move in continuous space with continuous velocities [7]. This is easy and fast to simulate, and has the advantage of making it easy to include other objects into the simulation. This is because, the interaction between a particle and an arbitrary object is comparatively simple to simulate. The mechanics of the SRD method is given below.

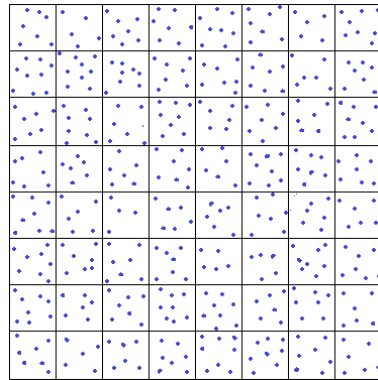


Figure 2.1: Example SRD grid with particles.

2.1 Algorithms

The algorithm consist of free streaming and multi-particle collision step, as the name refers to the fact that the collision consist of a random rotation of the relative velocities $\delta V_i = V_i - U$ of the particles in the collision cell, where U is the mean velocity of all the particles in a cell [9]. In MPCD collisions between fluid particles are replaced by multi-particle collision events that omit the molecular details and eliminate the need to calculate long-ranged forces between the fluid particles. This algorithm conserves the particle number, mass, and linear momentum. Together with a sufficient degree of isotropy, these properties lead to the reproduction of hydrodynamic behavior for a wide range of simulation parameters. The computational simplicity of the streaming and collision steps allows for highly parallel and efficient implementations, so that systems with 10^9 MPC particles can be simulated.

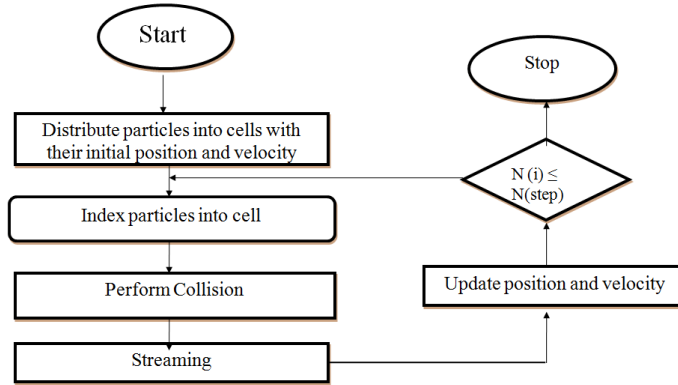


Figure 2.2: Schematic Flow Chart for the MPCD Algorithm

2.1.1 Position and Velocity Initialization

In multi-particle collision dynamics, we assign the random position and velocity to the particles in the x and y direction. Because we initialize the random velocity to the particles, it will follow some distribution that is nothing but the Maxwell- Boltzmann distribution. While assigning the random velocity of the particle we are taking pref-

actor, $v_{\text{const}} = \sqrt{12.0 * k_b T / m}$ because we are taking the velocities from the uniform random distribution but which is actually a Maxwell Boltzmann distribution. So, in order to make it compatible we have to normalize it and that prefactor comes from there. In Physics, the Maxwell–Boltzmann distribution is a probability distribution. It is used for describing particle speeds in idealized gases, where the particles move freely inside a stationary container without interacting with one another, except for very brief collisions in which they exchange energy and momentum with each other or with their thermal environment. Particle in this context refers to gaseous particles (atoms or molecules), and the system of particles is assumed to have reached thermodynamic equilibrium. A particle speed probability distribution indicates which speeds are more likely: a particle will have a speed selected randomly from the distribution, and is more likely to be within one range of speeds than another. The distribution depends on the temperature of the system and the mass of the particle. The Maxwell–Boltzmann distribution applies to the classical ideal gas, which is an idealization of real gases. In real gases, there are various effects (e.g. van der Waals interactions, vortical flow, relativistic speed limits, and quantum exchange interactions) that can make their speed distribution different from the Maxwell–Boltzmann form.

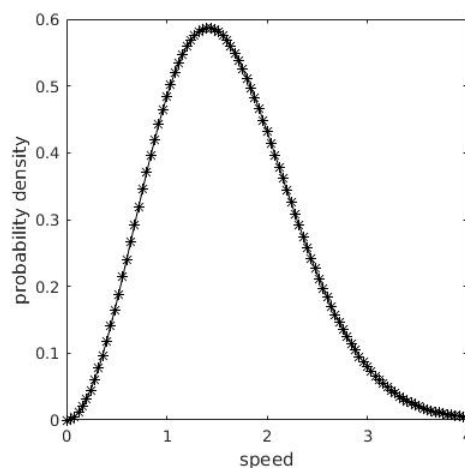


Figure 2.3: Comparison of the speed distribution(star) with the Maxwell-Boltzmann distribution function(solid line) for $K_b T=1$

2.1.2 Streaming

In the Streaming step the solvent molecules are point like particles having mass m , which moves in continuous space with a continuous distribution of velocity. In this step the particles move ballistically [7], and their positions are updated with discrete time interval Δt . During the streaming step additional forces can be applied to the particles for example a gravitational acceleration term could be applied to all the particles to create a gravity driven flow. In the streaming step the coordinates $\vec{r}_i(t)$ of all solvent particles at time t are simultaneously updated according to

$$\vec{r}_i(t + \Delta t) = \vec{r}_i(t) + \Delta t \vec{V}_i(t + \Delta t) \quad (2.1)$$

Streaming is done with the simple Euler integration. Where $V_i(t)$ is the velocity of particle i at time t and Δt is the value of the discretized time step [9]. The SRD simulation method was re-written in C++ from the FORTRAN code.

2.1.3 Collision

To efficiently simulate collisions between particles, both time and space are coarse-grained by using a square lattice (or cubic lattice) depending upon the dimension. In this step the momentum is transferred between fluid particles. To exchange the momentum the simulation domain is partitioned into d -dimensional square cells of size a . All the particles are assigned into whichever box they are in and total momentum is summed for each box [4]. To simulate collision, in each lattice cell all particles change their velocities according to

$$\vec{V}_i(t + \Delta t) = \vec{U}(t) + \mathbf{R} \delta \vec{V}_i(t) \quad (2.2)$$

All the particles in the cell are subject to the same rotation \mathbf{R} , but the rotations in different cells and at different times are statistically independent. The set of rotation used in MPCD dynamics can be chosen in various ways with appropriate choice of

rotation will determine the values of the transport properties of the system [9]. This is similar to molecular dynamics, as, the choice of intermolecular potential will determine the transport properties in a system. It is often convenient to use rotations about a randomly chosen direction \hat{n} by an angle α . The post collision velocity V_i of particle i arising from the rotation by an angle α is given by $\delta \vec{V}_i(t)$ is substituted as $(\vec{V}_i - \vec{U})$

$$\vec{V}_i(t + \Delta t) = \vec{U}(t) + \mathbf{R}(\vec{V}_i - \vec{U}) \quad (2.3)$$

Where \vec{U} is given by

$$\vec{U} = \frac{\sum_{i=1}^{N'} m_i V_i}{m_t} \quad (2.4)$$

$$m_t = \sum_{i=1}^{N'} m_i \quad (2.5)$$

N' is the number of particles in a box at any given time. On average $N' = N$. This \mathbf{R} is the rotation matrix or rotation tensor through an angle α [9]. The chosen axis is randomly generated for every collision operation. In three dimension, various strategies are possible for random rotation. The one involve in this work consist in picking a random axis n_μ (n_x, n_y, n_z) for each box, around which the relative velocities are rotated by a fixed angle θ . In this case the rotation matrix is

$$\mathbf{R} = \begin{pmatrix} n_x^2 + (1 - n_x^2)c & n_x n_y(1 - c) - n_z s & n_x n_z(1 - c) + n_y s \\ n_x n_y(1 - c) + n_z s & n_y^2 + (1 - n_y^2)c & n_y n_z(1 - c) - n_z s \\ n_x n_z(1 - c) - n_y s & n_y n_z(1 - c) + n_z s & n_z^2 + (1 - n_z^2)c \end{pmatrix} \quad (2.6)$$

where $c = \cos(\theta)$, $s = \sin(\theta)$ and the unit vector n_μ is defined as

$$n_x = \sqrt{1 - \rho^2} \cos \phi, n_y = \sqrt{1 - \rho^2} \sin \phi, n_z = \rho$$

Where ρ and ϕ are uncorrelated random numbers, which are taken from uniform distribution in the interval $[0, 2\pi]$ and $[-1, 1]$ respectively. We know from finite Rota-

tions,

$$\begin{aligned}
\mathbf{R}\delta\vec{V}_i(t) &= I \cdot \cos \theta \delta\vec{V}_i(t) + (1 - \cos \theta)(\hat{n}_\mu \cdot \delta\vec{V}_i(t))\hat{n}_\mu + \sin \theta(\hat{n}_\mu \times \delta\vec{V}_i(t)) \\
&= I \cdot \cos \theta \delta\vec{V}_i(t) + (\hat{n}_\mu \cdot \delta\vec{V}_i(t))\hat{n}_\mu - (\hat{n}_\mu \cdot \delta\vec{V}_i(t))\hat{n}_\mu \cos \theta + \sin \theta(\hat{n}_\mu \times \delta\vec{V}_i(t)) \\
&= \cos \theta \delta\vec{V}_i(t) \cdot (I - \hat{n}_\mu \cdot \hat{n}_\mu) + (\hat{n}_\mu \cdot \delta\vec{V}_i(t))\hat{n}_\mu + \sin \theta(\hat{n}_\mu \times \delta\vec{V}_i(t)) \\
&= (I - \hat{n}_\mu \cdot \hat{n}_\mu) \cdot (V_i - U(t)) \cos \theta + \hat{n}_\mu \hat{n}_\mu \cdot (V_i - U(t)) + \sin \theta(\hat{n}_\mu \times (V_i - U(t)))
\end{aligned} \tag{2.7}$$

$$\mathbf{R}\delta\vec{V}_i(t) = \hat{n}_\mu \hat{n}_\mu \cdot (V_i - U(t)) + (I - \hat{n}_\mu \hat{n}_\mu) \cdot (V_i - U(t)) \cos \theta + (\hat{n}_\mu \times (V_i - U(t))) \sin \theta \tag{2.8}$$

Substitute equation (2.8) in equation (2.2) we get,

$$\vec{V}_i(t + \Delta t) = \vec{U} + \hat{n} \hat{n} \cdot (\vec{V}_i - \vec{U}) + (I - \hat{n} \hat{n}) \cdot (\vec{V}_i - \vec{U}) \cos \theta - \hat{n} \times (\vec{V}_i - \vec{U}) \sin \theta \tag{2.9}$$

If the rotation axis is chosen from the set of orthogonal Cartesian axis ($\hat{x}\hat{y}\hat{z}$) then the collision operator is chosen randomly from $\hat{\Omega} \in (\hat{\Omega}_x, \hat{\Omega}_y, \hat{\Omega}_z)$, where

$$\hat{\Omega}_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha \\ 0 & -\sin \alpha & \cos \alpha \end{pmatrix} \tag{2.10}$$

$$\hat{\Omega}_y = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix} \tag{2.11}$$

$$\hat{\Omega}_z = \begin{pmatrix} \cos \alpha & \sin \alpha & 0 \\ -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.12}$$

Since, the rotation matrix \mathbf{R} is an orthogonal matrix, it satisfies

$$\mathbf{R}\mathbf{R}^{-1} = \mathbf{R}\mathbf{R}^T \quad (2.13)$$

The direction of rotation is randomly picked for each box. The collision process is demonstrated for two particles in Figures 2.4, 2.5 and 2.6.

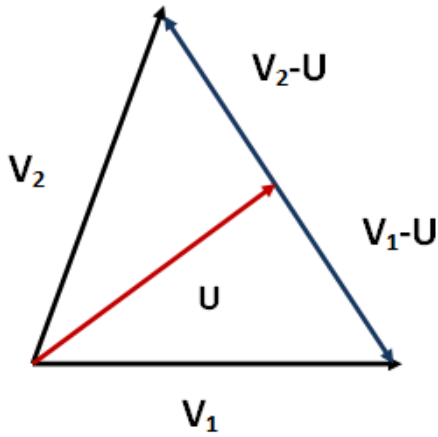


Figure 2.4: (a) Particles come in with velocities V_1 and V_2

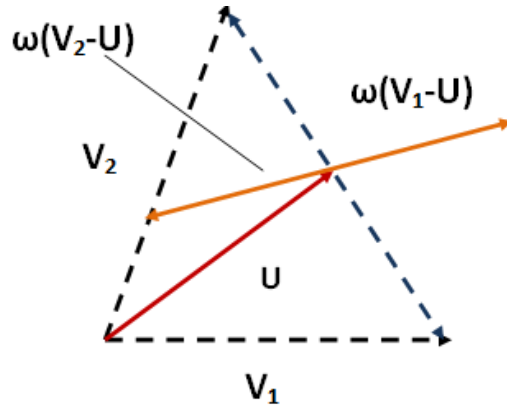


Figure 2.5: (b) Relative velocities are rotated by α

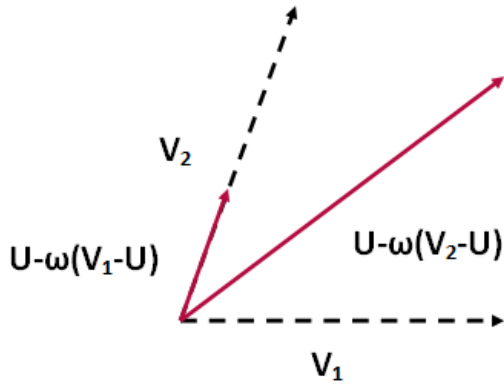


Figure 2.6: (c) Initial and final velocities

Figures 2.4, 2.5 and 2.6 are Example of an SRD collision with two equal mass particles. The collision angle $\theta = 130^\circ$, and U is the mean velocity of the two particles.

Chapter 3

Conservation laws

In this chapter MPCD conservation laws (mass, momentum and energy) are discussed, and because of that one can apply MPCD to any flow to analyze the flow behavior. The discussion begins with the analytical proof of the energy and momentum conservation, then we will compare the analytical results with the numerical results. The final section will discuss the thermostat. In MPCD, we used the thermostat to conserve the momentum of the system (or sometimes the temperature of the system).

3.1 Energy and Momentum Conservation

Since, the only operation being applied to the particles is a rotation in a static frame, kinetic energy and momentum are conserved within each box. This can be shown by calculating the momentum before and after the collision.

3.1.1 Analytical approach to the momentum and energy conservation

$$\sum_{i=1}^{N'} m_i \vec{V}_i(t + \Delta t) = \sum_{i=1}^{N'} m_i \mathbf{R}[(\vec{V}_i(t) - \vec{U}(t))] + m_t \vec{U}(t) \quad (3.1)$$

using the definition of \vec{U} from equation (2.4)

$$\begin{aligned}
 \sum_{i=1}^{N'} m_i \vec{V}_i(t + \Delta t) &= m_t \vec{U}(t) + \sum_{i=1}^{N'} \mathbf{R} m_i [\vec{V}_i(t) - \vec{U}(t)] \\
 &= m_t \vec{U}(t) + \sum_{i=1}^{N'} \mathbf{R} m_i [\vec{V}_i(t) - m_t \mathbf{R} \vec{U}(t)] \\
 &= m_t \vec{U}(t) + \sum_{i=1}^{N'} \mathbf{R} m_i [\vec{V}_i(t)] - \sum_{i=1}^{N'} m_i \mathbf{R} \vec{V}_i(t) \\
 &= \sum_{i=1}^{N'} m_i (\vec{V}_i(t)) + \sum_{i=1}^{N'} m_i [\mathbf{R}(\vec{V}_i(t) - \vec{V}_i(t))] \\
 &= \sum_{i=1}^{N'} m_i \vec{V}_i(t)
 \end{aligned}$$

Energy conservation can similarly be calculated by

$$\begin{aligned}
 \sum_{i=1}^N \frac{1}{2} m_i \vec{V}_i(t + \Delta t)^2 &= \sum_{i=1}^N \frac{1}{2} m_i [\mathbf{R}(\vec{V}_i(t) - \vec{U}(t)) + \vec{U}(t)]^2 \\
 \sum_{i=1}^{N'} \frac{1}{2} m_i \vec{V}_i(t + \Delta t)^2 &= \sum_{i=1}^{N'} \frac{1}{2} m_i [\vec{U}(t)^2 + 2\mathbf{R}(\vec{V}_i(t) - \vec{U}(t))\vec{U}(t) + \mathbf{R}\mathbf{R}^T(\vec{V}_i(t) - \vec{U}(t))^2]
 \end{aligned}$$

using equation (2.6) we get,

$$= \sum_{i=1}^{N'} \frac{1}{2} m_i [\vec{U}(t)^2 + 2\mathbf{R}(\vec{V}_i(t) - \vec{U}(t))\vec{U}(t) + (\vec{V}_i(t) - \vec{U}(t))^2]$$

and the expansion of the last term, by Equation (2.3) becomes

$$\begin{aligned}
 &= \frac{1}{2} m_t [2\vec{U}(t)^2 - 2\vec{U}(t)^2 + 2\mathbf{R}(\vec{U}(t) - \vec{U}(t))\vec{U}(t)] + \sum_{i=1}^{N'} \frac{1}{2} m_i \vec{V}_i(t)^2 \\
 &= \frac{1}{2} \sum_{i=1}^{N'} m_i \vec{V}_i(t)^2
 \end{aligned}$$

Since, momentum and energy are conserved within each individual box, they are also conserved in the sum of all boxes.

3.1.2 Numerical Approach

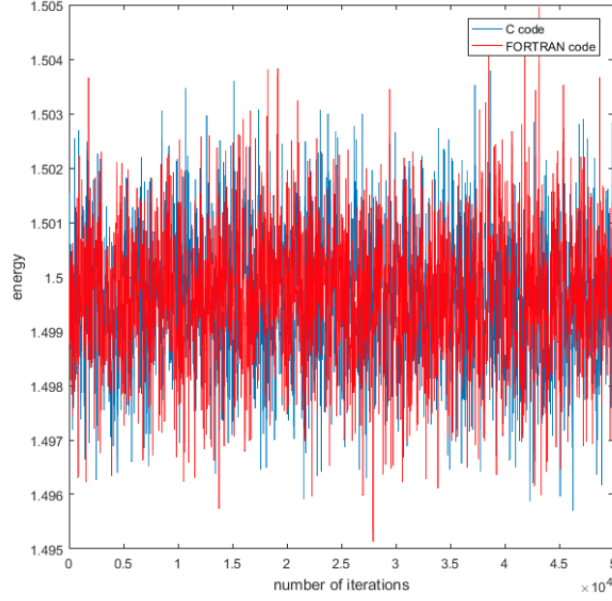


Figure 3.1: Per Particle energy conservation for each step

Fig. 3.1 shows the energy of the MPCD particles with time step using C and FORTRAN code, here we have done the simulation using C code. From the above Fig. 3.1 we can clearly see that the value of energy with respect to time step is fluctuating, but those fluctuations are very small, which means energy remains constant over the course of time. Similarly, momentum is also not changing over the course of time. The numerical result for momentum will be showed at the end of this section.

3.2 Thermostat

We perform simulations applying the thermostats, namely local simple scaling (LSS). In LSS the relative velocities $\delta v_i = V_i - U$ of the particles in a collision cell are scaled by a constant factor (scaling factor) let us say ξ , which typically differs for every cell and collision-time step. Hence, the relative velocities after collision $\delta v'_i$ are given by $\delta v'_i = \xi \delta v_i$. Since, the total relative momentum of a collision cell is zero, such a scaling leaves the total momentum of a cell unchanged [11].

3.2.1 Local simple scaling

In the simple scaling approach, the scale factor ξ is chosen as

$$\xi = \sqrt{3(N' - 1)K_b T / 3E_K} \quad (3.2)$$

with the kinetic energy of the collision cell

$$E_K = \frac{1}{2} \sum_{i=1}^{N'} m \delta v_i^2 \quad (3.3)$$

where N' is the number of instantaneous particles in a given cell. $3(N'-1)$ is the number of degrees of freedom of the fluid particles in the considered collision cell. The factor $3(N'-1)$ accounts for the fact that E_K is calculated in the center-of-mass reference frame of a collision cell [11]. LSS conserves the average kinetic energy rather than the temperature. This implies that the energy fluctuations are incompatible with that of an isothermal ensemble, and the distribution of velocities in a collision cell is not Maxwellian.

Numerical results for Energy and Momentum Conservation in 3D and 2D.

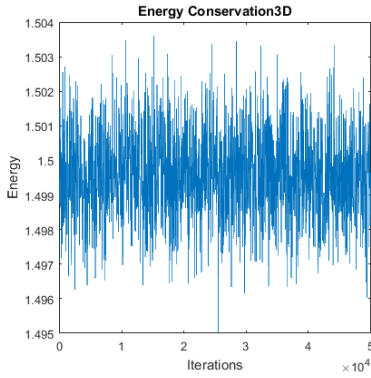


Figure 3.2: Energy Conservation in 3D (step = 25)

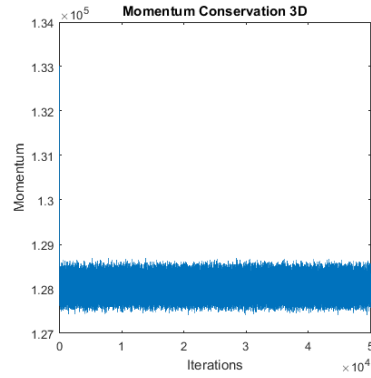


Figure 3.3: Momentum Conservation in 3D (step = 25)

Chapter 4

Transport Coefficient

4.1 Viscosity

Viscosity is one of the most important parameter to characterize the fluid. In this Section we investigate shear shear viscosity associated with the stochastic model. In MPCD the viscosity is divided into two contributions. The kinetic viscosity results from the momentum transferred during the free streaming step, and the collisional viscosity results from the momentum transferred during the velocity rotations.

4.1.1 Analytic expression for kinetic Viscosity

To calculate the kinetic viscosity we consider a system undergoing shear with rate $\dot{\gamma} = \frac{\partial u_x}{\partial y}$ in the x -direction and use a kinetic theory approach. Consider first two dimensions. On average the velocity profile is given by $v = (\dot{\gamma}y = 0)$. In the steady state there is a frictional force acting on any plane in the fluid perpendicular to y , which is given by the element of the stress tensor.

$$\sigma_{xy} = \eta \frac{\partial u_x}{\partial y} \quad (4.1)$$

where η is the shear viscosity which is equal to the flux of x -momentum crossing a plane of constant y . Our aim is to calculate the stress tensor during the streaming

step. For simplicity we take $y = y_0 = 0$. Consider a particle at position (x, y) with velocity (v_x, v_y) . Particles will only cross the plane during a time step if they are moving towards the plane and have a velocity such that $v_y \delta t$ is greater than the distance to the plane. Therefore, the stress tensor can be written

$$\begin{aligned} \sigma_{xy} = & -\frac{\rho}{\delta t} \int_{-\infty}^{\infty} dv_x \int_{-\infty}^0 dy \int_{-\frac{y}{\delta t}}^{\infty} dv_y v_x P(v_x - \dot{\gamma}y, v_y) \\ & + \frac{\rho}{\delta t} \int_{-\infty}^{\infty} dv_x \int_0^{\infty} dy \int_{-\frac{y}{\delta t}}^{\infty} dv_y v_x P(v_x - \dot{\gamma}y, v_y) \end{aligned} \quad (4.2)$$

where $P(v_x, v_y)$ is the velocity probability distribution of particles in the rest frame of the fluid and $\rho = \frac{m\gamma}{a^d}$ is the mass density. γ denotes the average number of particles per cell and $d = 2, 3$ is the dimensionality. By making the change of variable $v'_x = v_x - \dot{\gamma}y$ and changing the order of integration this reduces to

$$\sigma_{xy} = \frac{\dot{\gamma}\rho\delta t}{2} \langle v_y^2 \rangle - \rho \langle v_x v_y \rangle \quad (4.3)$$

where the averaging is over the probability distribution P . The first and second terms result from the shear profile and from induced correlations between v_x and v_y respectively. In the steady state the velocity distribution P deviates from the Maxwell Boltzmann form because of these correlations. To find the behavior of $\langle v_x v_y \rangle$ we calculate separately the effect of the streaming and collision operations. Consider the velocity distribution at the plane $y = y_0$. At time t , at y_0 , particles with positive velocity v_y come from $y_0 - v_y \delta t$ due to the streaming step, and correspondingly have a smaller x velocity. Conversely particles from $y > y_0$ tend to have a higher x velocity. This means that the velocity probability distribution is sheared by the streaming. That is

$$P^{after}(v_x, v_y) = P^{before}(v_x + \dot{\gamma}y\delta t, v_y) \quad (4.4)$$

Averaging $v_x v_y$ over this new distribution and changing variables gives

$$\begin{aligned}\langle v_x v_y \rangle^{after} &= \int_{-\infty}^{\infty} dv_x \int_{-\infty}^{\infty} dv_y v_x v_y P(v_x + \dot{\gamma} v_y \delta t, v_y) \\ &= \langle v_x v_y \rangle - \dot{\gamma} \delta t \langle v_y^2 \rangle\end{aligned}$$

Thus the streaming operation changes the value of $\langle v_x v_y \rangle$ by $-\dot{\gamma} \delta t \langle v_y^2 \rangle$ and tends to make v_x, v_y increasingly anti correlated. Next we consider the effect of the collision step on the velocity distribution. Collisions tend to reduce correlations. To see this we consider a cell of n particles and the two dimensional collision equation

$$v(t + \delta t) = \mathbf{R}[v(t) - v_{cm}] + v_{cm} \quad (4.5)$$

Where, \mathbf{R} is two dimensional rotation matrix as given above. Where, the sign corresponds to the direction of the rotation axis. The center of mass velocity v_{cm} can be divided into two contributions, $v_{cm} = (v + \hat{v}) / n$, the first from the test particle of velocity v and the second $\hat{v} = \sum_{i=1}^{n-1} v_i$ from the $(n-1)$ particles sharing the same cell. We assume molecular chaos, that is the velocities of different particles are uncorrelated. Then, $\langle v_x \hat{v}_y \rangle = \langle \hat{v}_x v_y \rangle = 0$ and $\langle \hat{v}_x v_y \rangle = (n-1) \langle v_x v_y \rangle$ and

$$\langle v_x(t + \delta t) v_y(t + \delta t) \rangle = \left[1 - \frac{n-1}{n} (1 - \cos 2\alpha) \right] \langle v_x(t) v_y(t) \rangle \quad (4.6)$$

Where, we have averaged over the direction of the rotation axis. In the simulations n is not constant. Therefore, we must consider density fluctuations. The probability of n particles being in a given cell is given by the Poisson distribution $\mathbf{P}(n) = e^{-\gamma} \gamma^n / n!$. The probability of a given particle being in a cell with a total of n particles is $\frac{n \mathbf{P}(n)}{\gamma}$ taking an average over this distribution.

$$\sum_{n=1}^{\infty} \frac{nP(n)}{\gamma} \langle v_x(t + \delta t) v_y(t + \delta t) \rangle = \left[1 - \frac{\gamma - 1 + e^{-\gamma}}{\gamma} (1 - \cos 2\alpha) \right] \langle v_x(t) v_y(t) \rangle \quad (4.7)$$

$$= f(\alpha, \gamma) \langle v_x(t) v_y(t) \rangle$$

During one time step, we find that $\langle \hat{v}_x v_y \rangle$ is first reduced by the streaming operation and then multiplied by a constant factor in the collision operation. In the steady state it will therefore oscillate between two values and we have the self-consistency condition $(\langle v_x(t) v_y(t) \rangle - \dot{\gamma} \delta t \langle v_y^2 \rangle) f = \langle v_x v_y \rangle$. Thus,

$$\langle v_x v_y \rangle = \frac{\dot{\gamma} \delta t f}{1 - f} \langle v_y^2 \rangle \quad (4.8)$$

Substitution into (4.3) gives

$$\sigma_{xy} = \rho \dot{\gamma} \delta t \langle v_y^2 \rangle \left(\frac{1}{2} + \frac{f}{1 - f} \right) \quad (4.9)$$

using equipartition energy, eqn.(4.1) and defination of viscosity

$$\eta_{kin}^{2D} = \frac{\gamma K_b t \delta t}{a^2} \left[\frac{\gamma}{(\gamma - 1 + e^{-\gamma})(1 - \cos 2\alpha)} - \frac{1}{2} \right] \quad (4.10)$$

$$\eta_{kin}^{3D} = \frac{\gamma K_b t \delta t}{a^3} \left[\frac{5\gamma}{(\gamma - 1 + e^{-\gamma})(4 - 2 \cos \alpha - 2 \cos 2\alpha)} - \frac{1}{2} \right] \quad (4.11)$$

4.1.2 Analytic expression for collisional Viscosity

The collisional viscosity η_{col} can also be calculated using a kinetic theory approach. Consider a cubic cell with side a . As in the case of the kinetic viscosity a shear rate $\dot{\gamma} = \frac{\partial u_x}{\partial y}$ is applied along x - axis. The system is divided into two subcells by a plane at $y = y_0$ where $0 \leq y_0 \leq a$. Let the subcell at $y_0 \leq y \leq a$ contain n_1 particles with an average collective flow velocity u_1 and that at $0 < y \leq y_0$ contain n_2 particles with an average velocity u_2 . We have $u_{1x} - u_{2x} = \frac{n_1 + n_2}{n_2} (u_{1x} - v_{cmx})$ where v_{cm} denotes the x -

component of the center of mass velocity of the cell. As the average distance between the subcell velocities is $\delta y = \frac{a}{2}$ the shear rate is

$$\dot{\gamma} = \frac{\partial u_x}{\partial y} = \frac{u_{1x} - u_{2x}}{\delta y} \dot{\gamma} = \frac{2n}{a(n - n_1)}(u_{1x} - v_{cmx}) \quad (4.12)$$

where $n = (n_1 + n_2)$ is the number of particles in the cell.

Our aim is to calculate the momentum crossing the plane at y_0 . We consider the n_1 particles at $y_0 < y \leq a$. The momentum transfer between the two subcells

$$\sigma_{xy} = -(\sum_i p_{ix}(t + \delta t) - \sum_i p_{ix}(t))/a^{d-1}\delta t \quad (4.13)$$

where i runs over the n_1 particles, is calculated using the collision operation (2.2).

Averaging over an isotropic distribution of the rotation axis gives

$$\sigma_{xy} = \frac{m}{a^{d-1}\delta t} \left[\frac{2}{d} n_1 (1 - \cos \alpha) (u_{1x} - v_{cmx}) \right] \quad (4.14)$$

where m is the mass of the solvent particle. Using eqns. (4.12), (4.14) and the definition of the viscosity (4.1) we obtain

$$\eta_{col} = \frac{mn_1(n - n_1)}{2n\delta t} (1 - \cos \alpha) \quad (4.15)$$

As $(n_1 + n_2)$ is in general small we must again consider fluctuations in the particle density. The numbers in the subcells, n_1 and n_2 are binomial distributed and averaging over them gives

$$\eta_{col} = \frac{m(1 - \cos \alpha)}{a^{d-2}d\delta t} (n - 1) \frac{y_0}{a} \frac{a - y_0}{a} \quad (4.16)$$

Here, we used $\sum n_1(n - n_1)P(n_1)_{bin} = n^2p - npq - n^2p^2$ where $p=1-\frac{y_0}{a}$ and $q=\frac{y_0}{a}$.

Using the Poisson distribution to average over n

$$\begin{aligned}\eta_{col} &= \frac{m(1 - \cos \alpha)}{a^{d-2}d\delta t} \frac{y_0}{a} \frac{a - y_0}{a} \sum_2^\infty (n-1) \frac{e^{-\gamma} \gamma^n}{n!} \\ &= \frac{m(1 - \cos \alpha)}{a^{d-2}d\delta t} (\gamma - a + e^{-\gamma}) \left(\frac{y_0}{a}\right) \left(1 - \frac{y_0}{a}\right)\end{aligned}\tag{4.17}$$

where the summation runs from 2 to ∞ because there is no momentum transfer unless $n_1 \geq 1$ and $n_2 \geq 1$. Finally, averaging over all planes $0 \leq y_0 \leq a$ gives

$$\eta_{col} = \frac{m(1 - \cos \alpha)}{6a^{d-2}d\delta t} (\gamma - 1 + e^{-\gamma})\tag{4.18}$$

In the large density limit the effect of the density fluctuations vanish and the theory agrees with the expression obtained by Ihle and Kroll [10] for the two dimensional case ($d = 2$). Also for case $\alpha = \frac{\pi}{2}$ and $d = 2$ the result presented here is identical to that given in by Malevanets and Kapral. When $d = 3$ we find that our expression differs from that given in for $\alpha = \frac{\pi}{2}$.

Chapter 5

Code Validations

5.1 Taylor Green vortex Flow

We have chosen the Taylor-Green Vortex simulation as a model problem, because it has a periodic domain. Initially the particles considered in the flow have velocities are given by

$$u(x, y, z, 0) = U \sin x \cos y \cos z$$

$$v(x, y, z, 0) = U \cos x \sin y \cos z$$

$$w(x, y, z, 0) = 0$$

We have used these relations to assign velocity corresponding to the position that is randomly assigned to the particles. We have updated average velocity of the cell using above relations. Fig.5.1, 5.2 shows the initial velocity field for Taylor-Green flow.

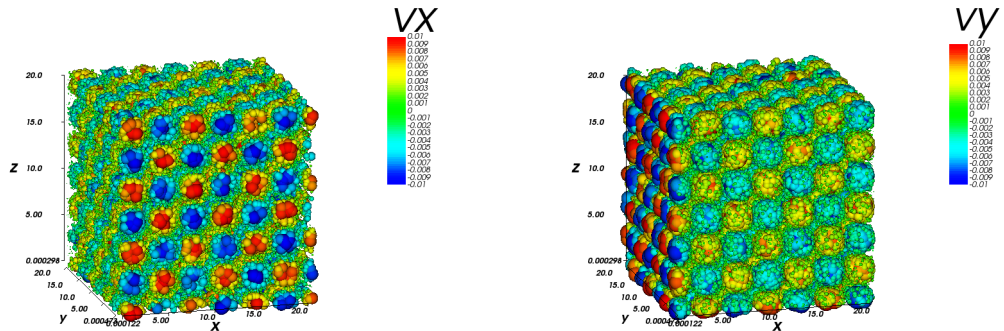


Figure 5.1: A stream-function plot corresponding to the initial conditions ($t = 0$) along y -axis for Taylor-Green vortex flow in 3D

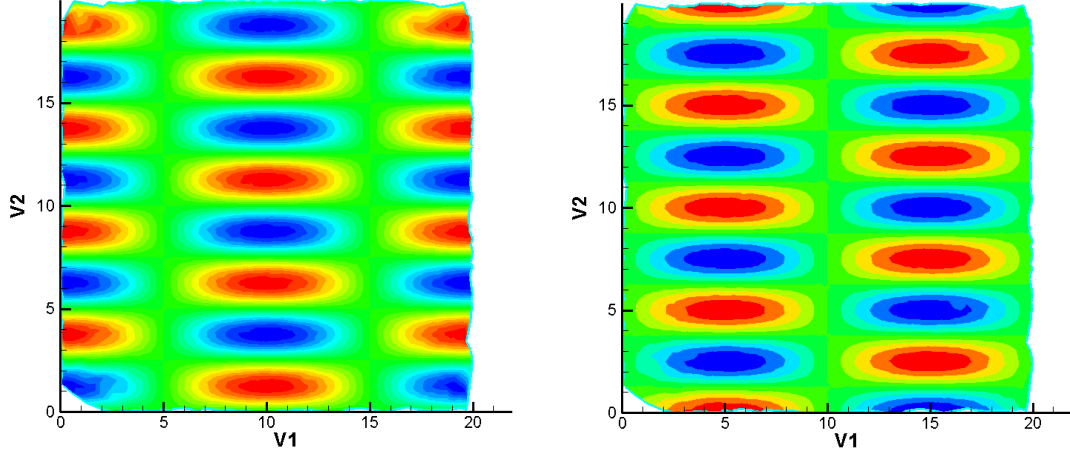


Figure 5.2: A stream-function plot corresponding to the initial conditions ($t = 0$) along x -axis and Y -axis for Taylor-Green vortex flow.

At any instant ($t > 0$), the initial magnitude of these velocities decay due to viscosity. Hence the energy will also decay. So, here we need the energy plot with time iterations. Generally, we perform the MPCD simulation with the thermostat. Earlier we have applied the thermostat LSS (local simple scaling), which we have described in chapter 3 section 3.1. Basically, LSS conserves the average kinetic energy rather than the temperature.

For non equilibrium or non homogeneous flow such as Taylor-Green, a thermostat is applied in order to enforce a homogeneous value of temperature θ . There are several thermostats strategies are possible. In order to introduce the decay of energy into the system, we used another thermostat. Here, in this thermostat initially we set the temperature ($\theta = \langle \frac{2mv^2}{3K_b} \rangle$) using equipartition theorem, where, K_b is Boltzmann constant. At each time step, after the collision phase, the particle velocities relative to the cell center δv_i are multiplied by the scaling factor $\frac{\theta}{\theta'}$ ($\theta' = \frac{1}{3(N_c-1)} \sum_{i=1}^{N_c} m \delta v_i^2$). Using this thermostat we got the energy dissipation, but the dissipation rate is very fast.

5.2 2D Poiseuille Flow (between two parallel plates)

5.2.1 Analytical Approach

Consider flow between two parallel plates whose spacing is $2h$ is assumed to be much smaller than other linear dimension of the channel Fig. 5.3. Irrespective of the inlet velocity profile, a velocity distribution is obtained at large values of x , which is invariant in x direction. The flow is then said to be fully developed. It is of interest to determine the shape of the velocity distribution in a parallel plates geometry. This

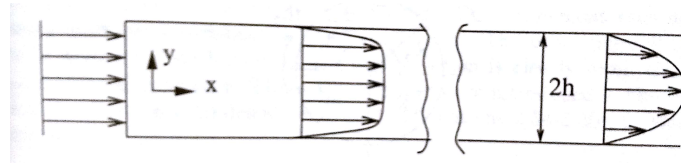


Figure 5.3: Flow between Parallel Plates.

configuration is often called Poiseuille flow.

In this situation we can assume two things.

- steady state (no change in time).
- Infinite channel ($\frac{\partial}{\partial x} = 0$)

There are two assumptions that simplify the problem. Let us write the Navier-Stokes equation for this situation.

$$X : \rho \left(\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} \right) = -\frac{\partial P}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \quad (5.1)$$

$$Y : \rho \left(\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} \right) = -\frac{\partial P}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (5.2)$$

$$Mass : \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (5.3)$$

Here we are solving for velocity vector, which has component form.

$$\vec{V} = \begin{bmatrix} u \\ v \end{bmatrix} \quad (5.4)$$

From the first assumption steady state (no change in time), all the derivative with respect to time is zero. and from second assumption (infinite channel) all the derivative with respect to x is zero. Therefore:

$$X : \rho \left(v \frac{\partial u}{\partial y} \right) = -\frac{\partial P}{\partial x} + \mu \left(\frac{\partial^2 u}{\partial y^2} \right) \quad (5.5)$$

$$Y : \rho \left(v \frac{\partial v}{\partial y} \right) = -\frac{\partial P}{\partial y} + \mu \left(\frac{\partial^2 v}{\partial y^2} \right) \quad (5.6)$$

$$Mass : \frac{\partial v}{\partial y} = 0 \quad (5.7)$$

from mass conservation v is constant everywhere and has the same constant. Hence vertical velocity cannot change according to conservation of mass. Vertical velocity at the wall has to be zero that implies $v = \text{constant} = 0$. Then we can say that there is infinite channel which consist of u velocity changing in x and no vertical flow bounded by two walls, therefore $\frac{\partial v}{\partial y} = 0$. We are left with

$$X : \frac{\partial P}{\partial x} = \mu \left(\frac{\partial^2 u}{\partial y^2} \right) \quad (5.8)$$

u is the only function of x .

$$Y : \frac{\partial P}{\partial y} = 0 \implies P = P(x) \quad (5.9)$$

$\frac{dP}{dx} = \text{constant} = -\frac{\Delta P}{L}$. total pressure across the channel.

$$\frac{d^2u}{dy^2} = -\frac{\Delta P}{L\mu} \quad (5.10)$$

$$u(y) = -\frac{\Delta P}{L\mu}y^2 + C_1y + C_2 \quad (5.11)$$

To determine the value of C_1 and C_2 , we have to apply the boundary condition.

$$u(h) = 0$$

$$u(-h) = 0$$

no slip condition (no fluid is slipped at the walls), after applying the no slip condition the value of C_1 is zero. and we are left with

$$u(y) = \frac{\Delta P}{2L\mu}(h^2 - y^2). \quad (5.12)$$

This is a parabolic profile for velocity which is symmetric about the axis $y = 0$.

5.2.2 Numerical Approach

Malevanets and Kapral presented two-dimensional Poiseuille flow results to demonstrate the general viability of the SRD algorithm [15]. They enforced the desired velocity profile using the inlet method, which resulted in the expected parabolic velocity profile with no slip at the walls. Because their simulation was in a regime where the mean free path was large, no bin shifting was required. The first three-dimensional SRD forced flow simulations were carried out by Allahyarov and Gompper [1]. They report having tried both the inlet flow and direct forcing methods to achieve Poiseuille flow and found the former to be problematic. They therefore used the direct forcing method coupled with a simple velocity rescaling thermostat. Here in our simulation we used the direct forcing method. The MPCD streaming step is modified to include an explicit constant force on each particle.

Chapter 6

Conclusion

We wrote an implementation of the SRD fluid simulation technique. This implementation was analytically proven to conserve momentum and energy, and demonstrated to do so in practice as well. In Multi-Particle Collision Dynamics since, particle pairs are not considered, the simulation scales much better. In this model no express molecule communication potential exists, which recognizes the model from other molecule based methods. In MPCD collisions between fluid particles are replaced by multi-particle collision events that omit the molecular details and eliminate the need to calculate long-ranged forces between the fluid particles. Therefore MPCD can simulate larger systems more efficiently. At the point when simulation is connected to the system, we can investigate the properties of the system.

Appendix

Algorithm

The SRD Simulation algorithm is written in C, the steps of the algorithm are given below.

Code Preamble

1. Define the variables to store position, velocity, mass, time step of integration.
2. Initialize the positions and velocities using random numbers to start with meaningful values also compute the relative velocities of the particles. (Set mass equal to unity, the integration step can be 0.1, K_b T is set to be unity).
3. Now set up a loop to perform the integration,
inside this loop do the following steps.
 - Calls the subroutines namely streaming, celllist, and collision.
 - Energy and momentum of the system.

Celllist

4. Divide the simulation domain in the number of cells.
5. Now set a loop to define the temporary position of the particles in the cells.
 - Inside the loop apply boundary conditions.

6. Check displacement (temporary position of particles) in x , y and z direction.
7. Again start a loop for integrating the number of particles in a cell number, let the index of this loop be i .

Collision

8. Compute center of mass velocity in each cell.
9. Do rotation.
10. Add average velocity back.
11. Apply thermostat.
12. Find the scaling factor with which velocities are to be multiplied such that temperature is adjusted to a preset value.

Streaming

13. Start a loop for integrating the position of particles in x , y and z direction.
14. Inside this loop apply boundary conditions (using if, else if statement).
15. Then compute energy and momentum of the system.

Bibliography

- [1] E Allahyarov and G Gompper. Mesoscopic solvent simulations: Multi-particle collision dynamics of three-dimensional flows. *Physical Review E*, 66(3): 036702, 2002.
- [2] P Allen. Introduction to molecular dynamics simulation. *Computational soft matter: from synthetic polymers to proteins*,. 23:1–28, 2004.
- [3] James Arvo.(1992.) Book: Fast random rotation matrices, *Citeseer*
- [4] Bjorck and Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971.
- [5] E De Angelis, M Chinappi, and G Graziani. Flow simulations with multi- particle collision dynamics. *Meccanica*, 47(8):2069–2077, 2012.
- [6] Dyson, Ransing, H Williams, and Williams. *Fluid properties at nano/meso scale: a numerical treatment*, volume 1. John Wiley and Sons, 2008.
- [7] Herbert Goldstein. Classical mechanics. Pearson Education India, 2011. *Classical mechanics*, Pearson Education India, 2011.
- [8] D Groot and Patrick B Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *The Journal of chemical physics*, 107(11):4423–4435, 1997.

- [9] Huang, Varghese, Gompper, and G Winkler. Thermostat for nonequilibrium multiparticle-collision-dynamics simulations. *Physical Review E*, 91(1):013310, 2015.
- [10] T Ihle and DM Kroll. Stochastic rotation dynamics: a Galilean-invariant mesoscopic model for fluid flow. *Physical Review E*, 63(2):020201, 2001.
- [11] Gerhard Kahl and David (2015)Book: “Multi-particle collision dynamics” simulation of viscoelastic fluids.
- [12] N Kikuchi, CM Pooley, JF Ryder, and JM Yeomans. Transport coefficients of a mesoscopic fluid dynamics model. *The Journal of Chemical Physics*, 119(12):6388–6395, 2003.
- [13] Kingsley. *A particle-based method for the simulation of complex fluids and polymer solutions*. PhD thesis, Worcester Polytechnic Institute, 2011.
- [14] A Lamura and G Gompper. Numerical study of the flow around a cylinder using multi-particle collision dynamics. *The European Physical Journal E: Soft Matter and Biological Physics*, 9(5):477–485, 2002.
- [15] Malevanets and Kapral. Mesoscopic model for solvent dynamics. *The Journal of Chemical Physics*, 110(17):8605–8613, 1999.
- [16] CM Pooley and JM Yeomans. Kinetic theory derivation of the transport coefficients of stochastic rotation dynamics. *The Journal of Physical Chemistry B*, 109(14):6505–6513, 2005.
- [17] AP Reid, H Hildenbrandt, JT Padding, and CK Hemelrijk. Fluid dynamics of moving fish in a two-dimensional multiparticle collision dynamics model. *Physical Review E*, 85(2):021901, 2012.
- [18] Tyler Shendruk. *Theoretical and Computational Studies of Hydrodynamics based Separation of Particles and Polymers in Microfluidic Channels*. PhD thesis, University of Ottawa, 2014.

- [19] Chakradhar Thantanapally, Dhiraj V Patil, Sauro Succi, and Santosh Ansumali. Universal mechanism for saturation of vorticity growth in fully developed fluid turbulence. *Journal of Fluid Mechanics*, 728:R4, 2013.
- [20] Wu-Ki Tung. Group Theory in Physics: An introduction to symmetry principles, group representations, and special functions in classical and quantum physics. *World Scientific Publishing*, 1985.
- [21] Versteeg and Malalasekera. *An introduction to computational fluid dynamics: The finite volume method*. Pearson Education, 2007.
- [22] J Wagner. A practical introduction to the lattice boltzmann method. *Adt. Notes for Statistical Mechanics*, 463:663, 2008.
- [23] Yamamoto, Hirota, and Fujiwara. A stochastic rotation dynamics model for dilute spheroidal colloid suspensions. *Journal of the Rheology Society of Japan*, 44(3):185–188, 2016
- [24] JM Yeomans. Mesoscale simulations: Lattice boltzmann and particle algorithms. *Physica A: Statistical Mechanics and its Applications*, 369(1):159–184, 2006.