Investigation of rheology and computation of viscosity of liquids at low temperatures

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Indrakanty Surya Shashank

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

December, 2024

Supervisor: Prof. Srikanth Sastry

Certificate

This is to certify that this dissertation entitled **Investigation of rheology and computation of viscosity of liquids at low temperatures** towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Indrakanty Surya Shashank at Jawaharlal Nehru Centre For Advanced Scientific Research, Bangalore under the supervision of Prof. Srikanth Sastry, Professor, Theoretical Sciences Unit , during the academic year 2024.

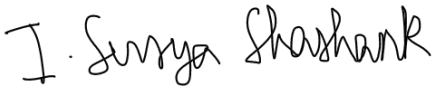
Prof. Srikanth Sastry

Committee:
Prof. Srikanth Sastry
Dr Apratim Chatterji

This thesis is dedicated to my parents and my sister.

Declaration

I hereby declare that the matter embodied in the report entitled **Investigation of rheology and computation of viscosity of liquids at low temperatures** are the results of the work carried out by me at the Theoretical Sciences Unit, Jawaharlal Nehru Centre For Advanced Scientific Research, Bengaluru, under the supervision of Prof. Srikanth Sastry and the same has not been submitted elsewhere for any other degree. Wherever others contribute, every effort is made to indicate this clearly, with due reference to the literature and acknowledgement of collaborative research and discussions.



Indrakanty Surya Shashank
20191031

Acknowledgments

First of all I would like to thank my supervisor Prof Srikanth Sastry for his constant support, enthusiasm and encouragement throughout the course of my project. His experience and mentorship has been invaluable to me and I am deeply grateful for the opportunity to work under his supervision. His constructive feedback and insightful advises has helped me not only grow academically but also personally and professionally.

I would thank Dr Anoop Krishnan and Abhijeet Gangan for their collaboration in the differentiable simulations to calculate the viscosity. It has been a great learning experience both in terms of physics as well as collaborative study.

I am also grateful to the Jawaharlal Nehru Centre For Advanced Scientific Research, Bangalore for providing me the opportunity to carry out my project. The resources and the environment there has been a constant source of motivation and greatly conducive to carry out my project.

I would also like to thank my lab-mates who have been a constant source of inspiration. The fun activities we had in the lab have definitely helped me to relax and focus better. I would like to thank Shiv, Raghav, Pushkar, Swarnendu, Krishna, Debargha, Jishnu, Prashanti, Om Prakash and Sounak who have made my stay at JNCASR and my project work memorable.

I would also like to thank Dr Ashwin SS who taught me the normalizing flows based on Neural ODEs as an attempt on calculation of the viscosity.

Finally, I would like to thank my parents and my sister. They have been a constant source of motivation and positivity that inspired me to push myself and work harder. Their love, care and support is a crucial part in my life that strengthens me.

Abstract

In this thesis we explore different methods of calculating the viscosity using molecular dynamics simulations. We initially investigated three methods, namely Green Kubo relation, Helfand moment method and the shear simulations using the SLLOD equations of motion. These simulations are performed in the canonical ensemble. Green Kubo and Helfand moment are equilibrium and strain rate independent methods while the SLLOD is the non-equilibrium method based on application of a shear flow and measuring the corresponding response in the form of steady state stress. The viscosity can then be estimated in the Newtonian regime as the ratio of the steady state stress to the applied strain rate. The standard Green Kubo formalism involves calculation of an autocorrelation function which is not very efficient. Green Kubo and Helfand moment methods are dependent on the stress relaxation time scales, which corresponds to the molecular relaxation time scales in dense gases and high temperature liquids. We then consider a new method for the computation of viscosity, namely through the use of differentiable simulations. We use the differential definition of the viscosity and differentiate the stress with respect to the strain rate to obtain the viscosity.

Contents

List of Algorithms

List of Figures

List of Tables

Chapter 1

Introduction

Viscosity is a fundamental property of fluids, that quantifies the resistance of fluids to externally applied stress. It plays an important role in diverse fields such as materials science, biophysics, and chemical engineering and influences processes from polymer processing to biological fluids [3]. Molecular Dynamics (MD) simulations have gained prominence as a robust computational technique to investigate viscosity.

MD simulations use classical mechanical equations of motion to extract various material properties. Using the microscopic time evolution of the system, one can extract the macroscopic properties such as viscosity and other transport properties [2]. Viscosity is defined to be the response of a fluid on the applied stress. In experiments, this is typically achieved by imposing deformation and measuring the corresponding response of the system. On the other hand, in computer simulations, one can measure viscosity as both equilibrium and non equilibrium response. Green Kubo [7, 12] and Helfand Moment [9] methods are equilibrium simulations, i.e. there is no applied deformation. Equilibrium methods to calculate the viscosity are based on linear response theory. On the other hand to get the strain rate dependent viscosity, the SLLOD equations of motion are employed that deform the fluid at a constant rate $\dot{\gamma}$. Then the corresponding steady state stress is calculated. The viscosity can be obtained as the ratio of steady state stress to the deformation rate $\dot{\gamma}$.

In this thesis we present the computational methods that are used to calculate the viscosity of fluids. In particular we focus on the the well studied Kob-Andersen BMLJ glass forming liquid. We apply the different methods of computing the viscosity on this fluid at different temperatures and calculate the viscosity. While it is feasible to calculate the viscosity at the

higher temperatures, it becomes harder to calculate the viscosity at the lower temperatures. This is particularly because the viscosity is a dynamic property that is dependent on the time scale of the dynamics of the system. As the system cools down, the time scales grow to a level that are not accessible by the molecular dynamics simulations. We present this case and show that indeed it is very difficult to obtain the viscosity at low temperatures.

We also describe an attempt of calculating the viscosity using the differentiable simulations. We here define viscosity to be the derivative of the stress with respect to the shear rate. The SLLOD equations of motion are then implemented in the JAX MD [18] package implemented in Python. This allows the usage Automatic Differentiation to get the derivative of the stress with respect to the strain rate giving the differential viscosity. This definition of viscosity is particularly more important when looking at the viscosity at the vanishing strain rates.

This thesis is organized into following chapters:

In **chapter 2** we discuss the basic idea of molecular dynamics simulations. The algorithm to implement MD is provided. Then the integration scheme is derived which is used to update the positions and velocities of the particles in the simulation. We also discuss three common ensembles of MD simulations. Firstly the microcanonical ensemble is presented with the integration scheme. Then Canonical Ensemble is introduced with Nosé Hoover thermostat and the corresponding implementation algorithm is presented. Finally Isothermal-Isobaric ensemble is presented with the integration scheme.

In **chapter 3** we discuss the methods of viscosity calculations that are used along with the MD simulations. We first introduce the Green Kubo method for calculating the equilibrium viscosity. Then another equilibrium method of Helfand moment is presented. Finally non equilibrium shear simulation are presented and the SLLOD equations of motion are derived along with the implementation algorithm. Some preliminary results are presented for which the code was written in FORTRAN.

In **chapter 4** We describe the differentiable simulations to calculate the viscosity. We discuss the importance of using the differential viscosity compared to dynamic viscosity. We then present a brief overview of working of the Automatic Differentiation and discuss how

this can be used as an attempt to estimate viscosity.

In **chapter 5** main results of this thesis are presented. Here, the viscosity of Kob Andersen system is calculated across temperature ranges and strain rates are presented. Also, a comparison of these different methods are presented.

In **chapter 6** Summary and concluding remarks are presented along with future directions for the calculation of the viscosity is discussed.

Chapter 2

Methods: Molecular Dynamics

Molecular Dynamics (MD) is a computational simulation method to study the time dependent and statistical properties at the the atomic and molecular scales. It is a strong technique that allows to study thermodynamic, transport, elastic and other properties of materials without the need for experimentation. MD was initially developed in the 1950s for the study of the behavior of simple systems like hard spheres and liquids [1]. MD now is an indispensable tool to calculate properties of materials from industrial importance to medical research and drug discovery.

At its core, MD simulations involve solving Newton's equations of motion for a system of interacting particles, where the forces acting on each atom are derived from potential energy functions. These forces can be empirically derived or calculated from first principles through quantum mechanical calculations. The Newton's equations of motion update the position and velocity of each particle over small time increments, typically on the order of femtoseconds (10^{-15} seconds), which allows for a detailed analysis of molecular motion over time [6].

Although these simulations are very versatile and allow probing the microscopic length scales, these simulations are severely limited by the computational costs. The time steps in MD are limited by the fastest atomic motions, such as bond vibrations, restricting simulations to relatively short time scales (typically nanoseconds to microseconds). Since, most classical MD simulations involve empirical force fields, the simulated properties are limited to the models that are accurately described by these potentials. [13].

2.1 The Basic Idea of MD

Classical MD simulation consists of defining the system environment. The simualtion could be performed in microcanonical, canonical or isothermal-isobaric ensembles. The ensemble defines what are the quantities or parameters that fix the environment under which the simulation is performed. After choosing the right ensemble and parameters governing them, the system is initialized by giving the initial positions and the velocities along with specifying the boundary conditions. For studying the bulk properties, typically the periodic boundary conditions are used. Then the forces are calculated and the equations of motion are integrated to move one time step. This procedure is repeated and quantities of interest are calculated. The discussion presented here is followed from [6].The basic MD algorithm can be summed up as follows:

Algorithm 1 MD Program

Initialize **positions** and **velocities**
$t = 0$
while $t \leq t_{\max}$ **do**
  calculate forces
  call **integrator**
  calculate required quantities
  $t = t + \Delta t$
end while loop
call **analysis**
stop

2.2 Microcanonical Ensemble (NVE)

In the microcanonical ensemble, we sample the trajectory where the total energy of the system is held constant. The Hamiltonian of the system is:

$$\mathcal{H} = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + \mathcal{U}\left(\{\mathbf{r}_i\}\right) \tag{2.1}$$

The Hamilton's equations of motion give:

$$\begin{aligned} \dot{\mathbf{p}} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{r}} \\ \dot{\mathbf{r}} &= \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \end{aligned} \tag{2.2}$$

Consider a function $f = f(\mathbf{r}, \mathbf{p})$ (we use shorthand notation $\mathbf{r}$ and $\mathbf{p}$ for all particles), which depends only on position and momenta of the particles and has no explicit time dependence. The time dependence occurs via the coordinates and the momenta. The time derivative of f is then:

$$\begin{aligned} \frac{df}{dt} &= \dot{\mathbf{p}} \cdot \frac{\partial f}{\partial \mathbf{p}} + \dot{\mathbf{r}} \cdot \frac{\partial f}{\partial \mathbf{r}} \\ &= \left[\dot{\mathbf{p}} \cdot \frac{\partial}{\partial \mathbf{p}} + \dot{\mathbf{r}} \cdot \frac{\partial}{\partial \mathbf{r}} \right] f \\ &= iLf \end{aligned} \tag{2.3}$$

Here, we identify the operator in the square brackets as iL the Liouville operator. Then we can write the solution of Equation (2.3) as:

$$f[\mathbf{r}(t), \mathbf{p}(t)] = \exp(iLt) f[\mathbf{r}(0), \mathbf{p}(0)] \tag{2.4}$$

Now consider the following:

$$\begin{aligned} iL_r &= \dot{\mathbf{r}} \frac{\partial}{\partial \mathbf{r}} \\ iL_p &= \dot{\mathbf{p}} \frac{\partial}{\partial \mathbf{p}} \end{aligned}$$

Then, we can write:

$$iL = iL_r + iL_p$$

So that that Equation (2.4) can now be written as:

$$f[\mathbf{r}(t), \mathbf{p}(t)] = \exp[(iL_r + iL_p)t] f[\mathbf{r}(0), \mathbf{p}(0)] \tag{2.5}$$

Let us now consider the action of $\exp(iL_r t)$ on the f.

$$\begin{aligned}
\exp\left(iL_r t\right) f\left[\mathbf{p}^N\left(0\right), \mathbf{r}^N\left(0\right)\right] &= \sum_{n=0}^{\infty} \frac{\left(iL_r\right)^n}{n!} f\left[\mathbf{p}^N\left(0\right), \mathbf{r}^N\left(0\right)\right] \\
&= \sum_{n=0}^{\infty} \frac{1}{n!} \left(\dot{\mathbf{r}}\left(0\right) \frac{\partial}{\partial \mathbf{r}}\right)^n f\left[\mathbf{p}^N\left(0\right), \mathbf{r}^N\left(0\right)\right] \\
&= \sum_{n=0}^{\infty} \frac{\left(\dot{\mathbf{r}}\left(0\right)\right)^n}{n!} \frac{\partial^n}{\partial \mathbf{r}^n} f\left[\mathbf{p}^N\left(0\right), \mathbf{r}^N\left(0\right)\right] \\
&= f\left[\mathbf{p}^N\left(0\right), \left(\mathbf{r} + \dot{\mathbf{r}}\left(0\right) t\right)^N\right]
\end{aligned}$$

Similarly the action of $\exp\left(iL_p t\right)$ on f would be:

$$\exp\left(iL_p t\right) f\left[\mathbf{p}^N\left(0\right), \mathbf{r}^N\left(0\right)\right] = f\left[\left(\mathbf{p} + \dot{\mathbf{p}}\left(0\right) t\right)^N, \mathbf{r}^N\left(0\right)\right]$$

So that the action of operator $\exp\left(iL_r t\right)$ is to shift the position and the action of $\exp\left(iL_p t\right)$ is to shift the momentum.

Thus, in general we can conclude:

$$\exp\left(\alpha \frac{\partial}{\partial x}\right) f\left(x\right) = f\left(x + \alpha\right)$$

It is important to note that for any two operators A and B, that need not commute, $[A, B] = AB - BA \neq 0$, we have:

$$\exp\left(A + B\right) \neq \exp\left(A\right) \exp\left(B\right)$$

We may employ the Trotter Factorization:

$$\exp\left(A + B\right) = \lim_{P \to \infty} \left(\exp\left(\frac{A}{2P}\right) \exp\left(\frac{B}{P}\right) \exp\left(\frac{A}{2P}\right)\right)^P$$

For finite P, we have:

$$\exp\left(A + B\right) \approx \left(\exp\left(\frac{A}{2P}\right) \exp\left(\frac{B}{P}\right) \exp\left(\frac{A}{2P}\right)\right)^P \exp\left(\mathcal{O}\left(\frac{1}{P^2}\right)\right)$$

We now show that iL_r and iL_p do not commute. We consider 1-D case, with the following

Hamiltonian:

$$\mathcal{H} = \frac{p^2}{2m} + V(x)$$

In this case we have:

$$\begin{aligned} iL_r &= \dot{x}\frac{\partial}{\partial x} \\ &= \frac{p}{m}\frac{\partial}{\partial x} \end{aligned}$$

And similarly

$$iL_p = F\frac{\partial}{\partial p}$$

where we used the fact that $\dot{p} = F$. Now, we calculate the commutator of these acting on a function $\phi(x,p)$. We have:

$$\begin{aligned} iL_r iL_p \phi(x,p) &= \frac{p}{m}\frac{\partial}{\partial x}\left[F(x)\frac{\partial}{\partial p}\phi(x,p)\right] \\ &= \frac{p}{m}\left[\frac{\partial F}{\partial x}\frac{\partial \phi}{\partial p} + F(x)\frac{\partial^2}{\partial x \partial p}\right] \end{aligned}$$

And

$$\begin{aligned} iL_p iL_r &= F(x)\frac{\partial}{\partial p}\frac{p}{m}\frac{\partial}{\partial x}\phi(x,p) \\ &= \frac{F(x)}{m}\left[p\frac{\partial^2 \phi}{\partial p \partial x} + \frac{\partial \phi}{\partial x}\right] \end{aligned}$$

Thus the commutator is:

$$[iL_r, iL_p] = \frac{p}{m}\frac{dF}{dx}\frac{\partial}{\partial p} - \frac{F(x)}{m}\frac{\partial}{\partial x} \neq 0$$

The commutator is not identically zero.

Now we identify:

$$\frac{A}{P} = \frac{iL_p t}{P} \equiv \Delta t \dot{\mathbf{p}}\frac{\partial}{\partial \mathbf{p}}$$

$$\frac{B}{P} = \frac{iL_r t}{P} \equiv \Delta t \dot{\mathbf{r}}\frac{\partial}{\partial \mathbf{r}}$$

where, $\Delta t = t/P$. The Trotter expansion for our case can be written as:

$$e^{iL\Delta t} = e^{iL_p\Delta t/2} e^{iL_r\Delta t} e^{iL_p\Delta t/2}$$

The action of these operators is:

$$\begin{aligned} \text{Apply} \exp\left(iL_r\Delta t\right) &\implies \mathbf{r}_i \to \mathbf{r}_i + \mathbf{v}_i\Delta t \\ \text{Apply} \exp\left(iL_p\Delta t\right) &\implies \mathbf{v}_i \to \mathbf{v}_i + \frac{\mathbf{f}_i}{m_i}\Delta t \end{aligned}$$

Translating the Trotter factorization into an algorithm we get the velocity verlet algorithm.

Algorithm 2 Velocity Verlet Algorithm

for i = 1, N **do**
  $\mathbf{v}_i \to \mathbf{v}_i + \frac{\mathbf{F}_i}{m_i}\frac{\Delta t}{2}$
  $\mathbf{r}_i \to \mathbf{r}_i + \mathbf{v}_i\Delta t$
  Calculate Forces
  $\mathbf{v}_i \to \mathbf{v}_i + \frac{\mathbf{F}_i}{m_i}\frac{\Delta t}{2}$
end for

2.3 Canonical Ensemble (NVT)

In the canonical ensemble, the temperature of the system is the control parameter instead of the total energy as in the case of the microcanonical ensemble. This is achieved by using a thermostat. The most common thermostat is the Nosé Hoover thermostat [17, 10]. The thermostatting is achieved by introducing an extra coordinate in the Lagrangian of the $N-$body system:

$$\mathcal{L}_{\text{Nosé}} = \sum_{i=1}^{N} \frac{m_i}{2} s^2 \mathbf{r}_i^2 - \mathcal{U}\left(\mathbf{r}^N\right) + \frac{1}{2} Q\dot{s}^2 - \frac{L}{\beta} \ln\left(s\right)$$

Here, $\beta = 1/k_B T$ is the inverse temperature, Q is the effective mass associated with s that acts as the damping parameter and decides how rapidly does the temperature converges. Canonical ensemble is achieved by setting $L = 3N - 3$ [6]. The following equations of motion are obtained as derived in [6].

$$
\begin{aligned}
\dot{r_i} &= v_i \\
\dot{v_i} &= \frac{f_i}{m_i} - v_\xi v_i \\
\dot{\xi} &= v_\xi \\
\dot{v_\xi} &= G
\end{aligned}
\tag{2.6}
$$

where,

$$
G = \frac{1}{Q}\left(\sum_{i=1}^{N}\frac{p_i^2}{m_i} - \frac{L}{\beta}\right) = \frac{1}{Q}\left(\sum_{i=1}^{N} m_i v_i^2 - \frac{L}{\beta}\right)
$$

Here, $\xi = \ln s$. These equations of motion have a conserved quantity $\mathcal{H}_{\text{Nosé}}$.

$$
\mathcal{H}_{\text{Nosé}} = \sum_{i=1}^{N}\frac{1}{2}mv_i^2 + \mathcal{U}\left(\mathbf{r}^N\right) + \frac{1}{2}Qv_\xi^2 + \frac{L}{\beta}\xi \tag{2.7}
$$

The Liouville operator is:

$$
\begin{aligned}
iL &= \dot{r_i}\frac{\partial}{\partial r_i} + \dot{v_i}\frac{\partial}{\partial v_i} + \dot{\xi}\frac{\partial}{\partial \xi} + \dot{v_\xi}\frac{\partial}{\partial v_\xi} \\
&= v_i\frac{\partial}{\partial r_i} + \frac{f_i}{m_i}\frac{\partial}{\partial v_i} - v_\xi v_i\frac{\partial}{\partial v_i} + v_\xi\frac{\partial}{\partial \xi} + G\frac{\partial}{\partial v_\xi} \\
iL &= iL_T + iL_r + iL_v
\end{aligned}
$$

with

$$
\begin{aligned}
iL_r &= v_i\frac{\partial}{\partial r_i} \\
iL_v &= \frac{f_i}{m_i}\frac{\partial}{\partial v_i}
\end{aligned}
$$

$$
\begin{aligned}
iL_T &= v_\xi\frac{\partial}{\partial \xi} - v_\xi v_i\frac{\partial}{\partial v_i} + G\frac{\partial}{\partial v_\xi} \\
&= iL_\xi + iL_{C_v} + iL_G
\end{aligned}
$$

Following the Trotter Factorization, one can obtain the update equations for the Nosé Hoover thermostat as shown in [16, 22]. The Trotter factorizations for the iL and iL_{T} gives:

$$
\exp\left(iL\Delta t\right) = \exp\left(iL_T\frac{\Delta t}{2}\right)\exp\left(iL_v\frac{\Delta t}{2}\right)\exp\left(iL_r\Delta t\right)\exp\left(iL_v\frac{\Delta t}{2}\right)\exp\left(iL_T\frac{\Delta t}{2}\right)
$$

$$\exp\left(iL_T\frac{\Delta t}{2}\right) = \exp\left(iL_{C_v}\frac{\Delta t}{4}\right)\exp\left(iL_G\frac{\Delta t}{4}\right)\exp\left(iL_\xi\frac{\Delta t}{2}\right)\exp\left(iL_G\frac{\Delta t}{4}\right)\exp\left(iL_{C_v}\frac{\Delta t}{4}\right)$$

Using

$$\exp\left(\alpha x\frac{\partial}{\partial x}\right)f(x) = f\left(xe^{\alpha}\right)$$

for $e^{iL_{C_v}\Delta t/4} \implies v_i \to v_i \exp\left(-v_\xi\frac{\Delta t}{4}\right)$ the algorithm for the thermostatting part is obtained Algorithm 3

Algorithm 3 Thermostat

for i = 1, N **do**
  $v_i \to v_i \exp\left(-v_\xi\frac{\Delta t}{4}\right)$
end for
Calculate G
$v_\xi \to v_\xi + G\frac{\Delta t}{4}$
$\xi \to \xi + v_\xi\frac{\Delta t}{2}$
$v_\xi \to v_\xi + G\frac{\Delta t}{4}$
for i = 1, N **do**
  $v_i \to v_i \exp\left(-v_\xi\frac{\Delta t}{4}\right)$
end for

The full Nosé Hoover thermostat algorithm is given by Algorithm 4

Algorithm 4 Nosé Hoover Algorithm

Apply Thermostat
Apply Velocity Verlet Algorithm
Apply Thermostat

We now show that the Nosé Hoover thermostat samples from the canonical ensemble. The following discussion has been adopted from the [6]. Unlike velocity rescaling schemes, where, the fluctuations in the temperature are suppressed, the Nosé Hoover thermostat allows for the fluctuations. These fluctuations scale inversely with the system size. The Maxwell Boltzmann Velocity distribution is given by:

$$\mathcal{P}(p) = \left(\frac{\beta}{2\pi m}\right)^{\frac{3}{2}}\exp\left(-\beta\frac{p^2}{2m}\right)$$

where, p is the momentum of the particle. We first calculate the relative variance in the kinetic energy of any single particle. Since Kinetic Energy is proportional to p^2, the relative

variance is given by (Kinetic Energy of the chosen particle is denoted as T):

$$\begin{aligned}\frac{\sigma_T^2}{\langle T\rangle^2} &= \frac{\langle T^2\rangle - \langle T\rangle^2}{\langle T\rangle^2} \\ &= \frac{\langle p^4\rangle - \langle p^2\rangle^2}{\langle p^2\rangle^2}\end{aligned}$$

Now, we need

$$\begin{aligned}\langle p^2\rangle &= \int d\vec{p}\; p^2\; \mathcal{P}\left(p\right) \\ &= \int_{p=0}^{\infty}\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} p^2 \sin\left(\theta\right) dp\; d\theta\; d\phi\; p^2\mathcal{P}\left(p\right) \\ &= 4\pi\int_{p=0}^{\infty} p^4\mathcal{P}\left(p\right)\; dp \\ &= 3\left(\frac{m}{\beta}\right)\end{aligned}$$

and,

$$\begin{aligned}\langle p^4\rangle &= \int d\vec{p}\; p^4\; \mathcal{P}\left(p\right) \\ &= \int_{p=0}^{\infty}\int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} p^2 \sin\left(\theta\right) dp\; d\theta\; d\phi\; p^4\mathcal{P}\left(p\right) \\ &= 4\pi\int_{p=0}^{\infty} p^6\mathcal{P}\left(p\right)\; dp \\ &= 15\left(\frac{m}{\beta}\right)^2\end{aligned}$$

In evaluating the above integrals, we have used the following formulas:

$$\begin{aligned}\int_0^{\infty} x^4 \exp\left(-ax^2\right) &= \frac{3\sqrt{\pi}}{8a^{\frac{5}{2}}} \\ \int_0^{\infty} x^6 \exp\left(-ax^2\right) &= \frac{15\sqrt{\pi}}{16a^{\frac{7}{2}}}\end{aligned}$$

Thus, relative variance of the kinetic energy of the chosen particle is:

$$\begin{aligned}
\frac{\sigma_T^2}{\langle T\rangle^2} &= \frac{\langle T^2\rangle - \langle T\rangle^2}{\langle T\rangle^2} \\
&= \frac{\langle p^4\rangle - \langle p^2\rangle^2}{\langle p^2\rangle^2} \\
&= \frac{15\left(\frac{m}{\beta}\right)^2 - 9\left(\frac{m}{\beta}\right)^2}{9\left(\frac{m}{\beta}\right)^2} \\
&= \frac{2}{3}
\end{aligned}$$

Now, we may take kinetic energy per particle $\mathcal{T}$ as a measure of the Temperature. So we have:

$$\begin{aligned}
\langle \mathcal{T}\rangle &= \langle \sum_{i=1}^{N} p_i^2\rangle \\
&= \sum_{i=1}^{N} \langle p_i^2\rangle \\
&= N\langle p^2\rangle
\end{aligned}$$

And,

$$\begin{aligned}
\langle \mathcal{T}^2\rangle &= \langle \sum_{i=1}^{N}\sum_{j=1}^{N} p_i^2 p_j^2\rangle \\
&= \langle \sum_{i=1}^{N} p_i^4 + \sum_{i\neq j} p_i^2 p_j^2\rangle
\end{aligned}$$

$$\begin{aligned}
\langle \mathcal{T}^2\rangle &= \sum_{i=1}^{N}\langle p^4\rangle + \left(\sum_{i}{}' \langle p_i^2\rangle\right)\left(\sum_{j\neq i}{}' \langle p_j^2\rangle\right) \\
&= N\langle p^4\rangle + N\langle p^2\rangle\,(N-1)\,\langle p^2\rangle \\
&= N\langle p^4\rangle + N\,(N-1)\,\langle p^2\rangle^2
\end{aligned}$$

Thus the relative variance in Temperature is:

$$\begin{aligned}\frac{\sigma_{\mathcal{T}}^2}{\langle \mathcal{T} \rangle^2} &= \frac{\langle \mathcal{T}^2 \rangle - \langle \mathcal{T} \rangle^2}{\langle \mathcal{T} \rangle^2} \\ &= \frac{N\langle p^4 \rangle + N\left(N-1\right)\langle p^2 \rangle^2 - N^2 \langle p^2 \rangle^2}{N^2 \langle p^2 \rangle^2} \\ &= \frac{2}{3N}\end{aligned}$$

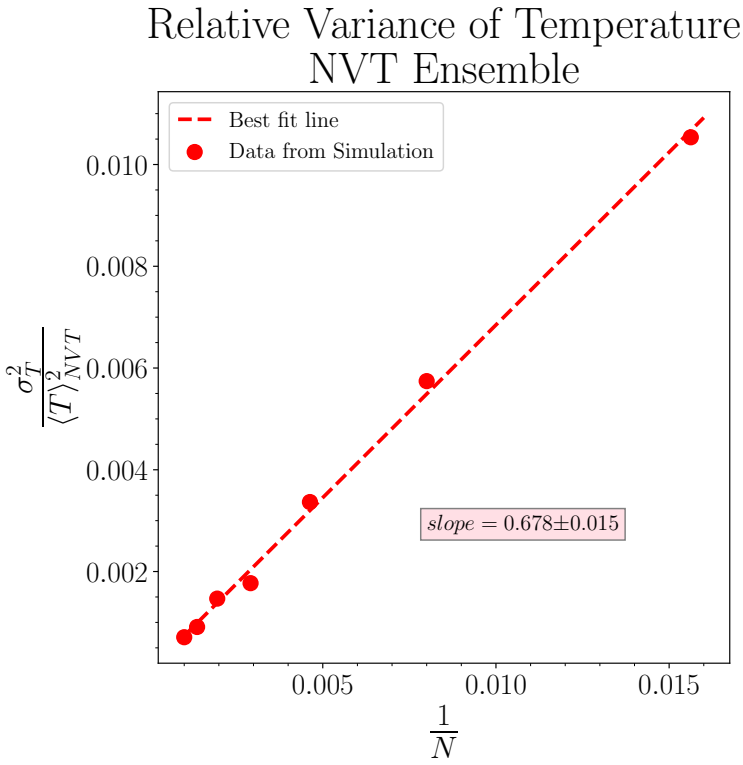


Figure 2.1: Scaling of temperature fluctuations with system size in canonical ensemble.

The slope obtained in the Figure 2.1 is close to theoretical value of 2/3 validating that the Nosé Hoover thermostat generates canonical ensemble.

2.4 Isothermal-Isobaric Ensemble (NPT)

The NPT involves fixing the pressure and temperature and letting the Volume fluctuate. In this ensemble, the system can exchange both energy (with thermostat) and volume (with barostat). This setup models systems under conditions where pressure is controlled, such

as chemical reactions in open containers. To study the effect of composition of solvent on system properties, then Volume has to be adjusted to ensure the constant Pressure [6]. This is achieved by introducing a barostat [15] and the corresponding equations of motion are [6]:

$$\begin{aligned}
\dot{r}_i &= \frac{p_i}{m_i} + \frac{p_\epsilon}{W} r_i \\
\dot{p}_i &= f_i - \left(1 + \frac{1}{N}\right) \frac{p_\epsilon}{W} p_i - \frac{p_{\xi_1}}{Q_1} p_i \\
\dot{V} &= \frac{3 V p_\epsilon}{W} \\
\dot{p}_\epsilon &= 3V \left(P_{\text{int}} - P_{\text{ext}}\right) + \frac{2K}{N} - \frac{p_{\xi_1}}{Q_1} p_\epsilon \\
\dot{\xi}_1 &= \frac{p_{\xi_1}}{Q_1} \\
\dot{\xi}_2 &= \frac{p_{\xi_2}}{Q_2} \\
\dot{p}_{\xi_1} &= 2K + \frac{p_\epsilon^2}{W} - (3N+1)\, k_B T - \frac{p_{\xi_2}}{Q_2} p_{\xi_1} \\
\dot{p}_{\xi_2} &= \frac{p_{\xi_1}^2}{Q_1} - k_B T
\end{aligned} \tag{2.8}$$

Here, W is the mass parameter of the barostat, ϵ and p_ϵ are the coordinate and momenta of the barostat. In this particular case Equation (2.8) a chain of thermostats of length two is used and the corresponding coordinate variables are ξ_1 and ξ_2 and the corresponding momenta are p_{ξ_1} and p_{ξ_2}. Q_1 and Q_2 are the mass parameters of the two thermostats. The variable ϵ is related to the volume of the simulation box:

$$\epsilon = \frac{1}{3} \ln V$$

The forces on the thermostats G_1 and G_2 and the force on barostat G_ϵ are given by:

$$\begin{aligned}
G_1 &= \frac{1}{Q_1} \left[\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m} + W v_\epsilon^2 - (3N+1)\, k_B T \right] \\
G_2 &= \frac{1}{Q_2} \left[Q_1 v_{\xi_1}^2 - k_B T \right] \\
G_\epsilon &= \frac{1}{W} \left[\left(1 + \frac{1}{N}\right) \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m} + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} r_{ij} f_{ij} m - 3 P_{\text{ext}} V - 3V \frac{\partial U}{\partial V} \right]
\end{aligned}$$

The Liouvillean is:

$$iL_{\mathrm{NPT}} = iL_r + iL_v + iL_{\mathrm{CP}}$$

where,

$$\begin{aligned}
iL_r &= (\mathbf{v} + v_\epsilon \mathbf{r}) \frac{\partial}{\partial \mathbf{r}} + v_\epsilon \frac{\partial}{\partial \epsilon} \\
iL_v &= \frac{\mathbf{F}}{m} \frac{\partial}{\partial \mathbf{v}} \\
iL_{\mathrm{CP}} &= -v_{\xi_1} \mathbf{v} \frac{\partial}{\partial \mathbf{v}} + v_{\xi_1} \frac{\partial}{\partial \xi_1} + v_{\xi_2} \frac{\partial}{\partial \xi_2} \\
&\quad + (G_1 - v_{\xi_1} v_{\xi_2}) \frac{\partial}{\partial v_{\xi_1}} + G_2 \frac{\partial}{\partial v_{\xi_2}} - \left(1 + \frac{1}{N}\right) v_\epsilon \mathbf{v} \frac{\partial}{\partial \mathbf{v}} \\
&\quad + (G_\epsilon - v_\epsilon v_{\xi_1}) \frac{\partial}{\partial v_\epsilon}
\end{aligned}$$

The operator iL_{CP} can be split as follows:

$$iL_{\mathrm{CP}} = iL_\xi + iL_{C_v} + iL_{G_\epsilon} + iL_{v_\epsilon} + iL_{G_1} + iL_{G_2} + iL_{v_{\xi_2}}$$

$$\begin{aligned}
iL_\xi &= v_{\xi_1} \frac{\partial}{\partial \xi_1} + v_{\xi_2} \frac{\partial}{\partial \xi_2} \\
iL_{C_v} &= -\left[v_{\xi_1} + \left(1 + \frac{1}{N}\right) \right] v_i \frac{\partial}{\partial v_i} \\
iL_{G_\epsilon} &= G_\epsilon \frac{\partial}{\partial v_\epsilon} \\
iL_{v_\epsilon} &= -v_{\xi_1} v_\epsilon \frac{\partial}{\partial v_\epsilon} \\
iL_{G_1} &= G_1 \frac{\partial}{\partial v_{\xi_1}} \\
iL_{v_{\xi_1}} &= -v_{\xi_1} v_{\xi_2} \frac{\partial}{\partial v_{\xi_1}} \\
iL_{G_2} &= G_2 \frac{\partial}{\partial v_{\xi_2}}
\end{aligned}$$

We can write the Trotter expansion of $e^{iL_{\mathrm{CP}}}$ as:

$$\begin{aligned} e^{iL_{\mathrm{CP}}\Delta t/2} = & e^{iL_{G_2}\Delta t/4}\left[e^{iL_{v_{\xi_1}}\Delta t/8}e^{iL_{G_1}\Delta t/4}e^{iL_{v_{\xi_1}}\Delta t/8}\right] \\ & \times \left[e^{iL_{v_\epsilon}\Delta t/8}e^{iL_{G_\epsilon}\Delta t/4}e^{iL_{v_\epsilon}\Delta t/8}\right] \\ & \times e^{iL_\xi \Delta t/2} \times e^{iL_{C_v}\Delta t/2} \\ & \times \left[e^{iL_{v_\epsilon}\Delta t/8}e^{iL_{G_\epsilon}\Delta t/4}e^{iL_{v_\epsilon}\Delta t/8}\right] \\ & \times \left[e^{iL_{v_{\xi_1}}\Delta t/8}e^{iL_{G_1}\Delta t/4}e^{iL_{v_{\xi_1}}\Delta t/8}\right] e^{iL_{G_2}\Delta t/4} \end{aligned}$$

The update rules can be derived and translated into an algorithm Algorithm 6. Simillarly the Trotter Factorization is applied to the $iL_r + iL_v$ operators. The position update is obtained analytically by solving:

$$\dot{r}_i = v_i + v_\epsilon r_i$$

The solution to this is:

$$r_i(t) = r_i(0)\, e^{v_\epsilon t} + v_i t \frac{\sinh\left(v_\epsilon \frac{t}{2}\right)}{\left(v_\epsilon \frac{t}{2}\right)} e^{v_\epsilon \frac{t}{2}}$$

So that update equations becomes:

$$r_i \to r_i e^{v_\epsilon \Delta t} + v_i \Delta t \frac{\sinh\left(v_\epsilon \frac{\Delta t}{2}\right)}{\left(v_\epsilon \frac{\Delta t}{2}\right)} e^{v_\epsilon \frac{\Delta t}{2}}$$

$$\epsilon \to \epsilon + v_\epsilon \Delta t$$

This gives the Algorithm 5. The full NPT algorithm is then given by Algorithm 7.

Algorithm 5 NPT Modified Velocity Verlet

for i = 1 to N **do**
$v_i \to v_i + \frac{f_i}{m}\frac{\Delta t}{2}$
$r_i \to r_i e^{v_\epsilon \Delta t} + v_i \Delta t \frac{\sinh\left(v_\epsilon \frac{\Delta t}{2}\right)}{\left(v_\epsilon \frac{\Delta t}{2}\right)} e^{v_\epsilon \frac{\Delta t}{2}}$
end for
$\epsilon \to \epsilon + v_\epsilon \Delta t$
Calculate Forces
for i = 1 to N **do**
$v_i \to v_i + \frac{f_i}{m}\frac{\Delta t}{2}$
end for

Algorithm 6 Thermo Baro

Calculate G_2
$v_{\xi_2} \to v_{\xi_2} + G_2 \frac{\Delta t}{4}$
Calculate G_1
$v_{\xi_1} \to v_{\xi_1} \exp\left(-v_{\xi_2} \frac{\Delta t}{8}\right)$
$v_{\xi_1} \to v_{\xi_1} + G_1 \frac{\Delta t}{4}$
$v_{\xi_1} \to v_{\xi_1} \exp\left(-v_{\xi_2} \frac{\Delta t}{8}\right)$
Calculate G_ϵ
$v_\epsilon \to v_\epsilon \exp\left(-v_{\xi_1} \frac{\Delta t}{8}\right)$
$v_\epsilon \to v_\epsilon + G_\epsilon \frac{\Delta t}{4}$
$v_\epsilon \to v_\epsilon \exp\left(-v_{\xi_1} \frac{\Delta t}{8}\right)$
for i = 1 to N **do**
$v_i \to v_i \exp\left[-v_{\xi_1} \frac{\Delta t}{2} - \left(1 + \frac{1}{N}\right) v_\epsilon \frac{\Delta t}{2}\right]$
end for
$\xi_1 \to \xi_1 + v_{\xi_1} \frac{\Delta t}{2}$
$\xi_2 \to \xi_2 + v_{\xi_2} \frac{\Delta t}{2}$
Calculate G_ϵ
$v_\epsilon \to v_\epsilon \exp\left(-v_{\xi_1} \frac{\Delta t}{8}\right)$
$v_\epsilon \to v_\epsilon + G_\epsilon \frac{\Delta t}{4}$
$v_\epsilon \to v_\epsilon \exp\left(-v_{\xi_1} \frac{\Delta t}{8}\right)$
Calculate G_1
$v_{\xi_1} \to v_{\xi_1} \exp\left(-v_{\xi_2} \frac{\Delta t}{8}\right)$
$v_{\xi_1} \to v_{\xi_1} + G_1 \frac{\Delta t}{4}$
$v_{\xi_1} \to v_{\xi_1} \exp\left(-v_{\xi_2} \frac{\Delta t}{8}\right)$
Calculate G_2
$v_{\xi_2} \to v_{\xi_2} + G_2 \frac{\Delta t}{4}$

Algorithm 7 NPT Algorithm

Apply Thermo Baro
Apply NPT modified Velocity Verlet
Apply Thermo Baro

In this work, the Nosé Hoover thermostat is implemented in the FORTRAN language which is needed to simulate the Green Kubo and Helfand moment calculations. NPT simulations are also implemented to equilibrate the system at the constant pressure before the production run.

Chapter 3

Methods: Calculation of Viscosity

In this chapter methods to calculate viscosity are discussed.

3.1 Green Kubo Method

Green Kubo methods are used to calculate transport coefficients like, diffusion constant, conductivity and viscosity from equilibrium MD simulations [7, 12]. Consider a system of N particles interacting via pair potentials. For such a system the stress tensor is defined as [6]:

$$\sigma_{\alpha\beta} = \frac{1}{V}\left(\sum_{i=1}^{N} m v_i^{\alpha} v_i^{\beta} + \sum_{i=1}^{N}\sum_{j>i} r_{ij}^{\alpha} f_{ij}^{\beta}\right) \tag{3.1}$$

where, α, $\beta \in (x, y, z)$ are the cartesian components. The r_{ij} is the nearest image distance between the particles labelled i and j and f_{ij} is the pair-wise force between the particles i and j. The Green-Kubo relations relate the transport properties to the autocorrelation functions. The shear viscosity η is given by the integral of the stress autocorrelation function.

$$\eta = \frac{V}{k_B T}\int_0^{\infty} dt \ \left\langle \sigma_{\alpha\beta}(t)\, \sigma_{\alpha\beta}(0)\right\rangle \tag{3.2}$$

For improved numerical accuracy, we use the component averaged quantity:

$$\begin{aligned} \eta &= \frac{V}{k_B T}\int_0^{\infty} dt \ \left\langle \sigma_{\alpha\beta}(t)\, \sigma_{\alpha\beta}(0)\right\rangle_{\text{avg}} \\ \left\langle \sigma_{\alpha\beta}(0)\, \sigma_{\alpha\beta}(t)\right\rangle_{\text{avg}} &= \frac{\left\langle \sigma_{xy}(0)\, \sigma_{xy}(t)\right\rangle + \left\langle \sigma_{yz}(0)\, \sigma_{yz}(t)\right\rangle + \left\langle \sigma_{zx}(0)\, \sigma_{zx}(t)\right\rangle}{3} \end{aligned} \tag{3.3}$$

The integral in Equation (3.2) is calculated by taking the cumulative sum of the averaged stress autocorrelation function. As the autocorrelation function decays to zero, this integral reaches a constant value in principle. A plateau region is observed in the integral around the time when the autocorrelation first reaches zero. This value is taken to be the estimate of the shear viscosity η. A test simulation is carried out with $N = 2048$ particles for the WCA potential Equation (3.4), at the reduced density of $\rho = 0.8442$ and reduced temperature $T = 0.722$.

$$\phi_{\mathrm{WCA}}(r) = \begin{cases} 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] + \epsilon, & \text{for } r \leq 2^{\frac{1}{6}}\sigma \\ 0, & \text{for } r > 2^{\frac{1}{6}}\sigma \end{cases} \tag{3.4}$$

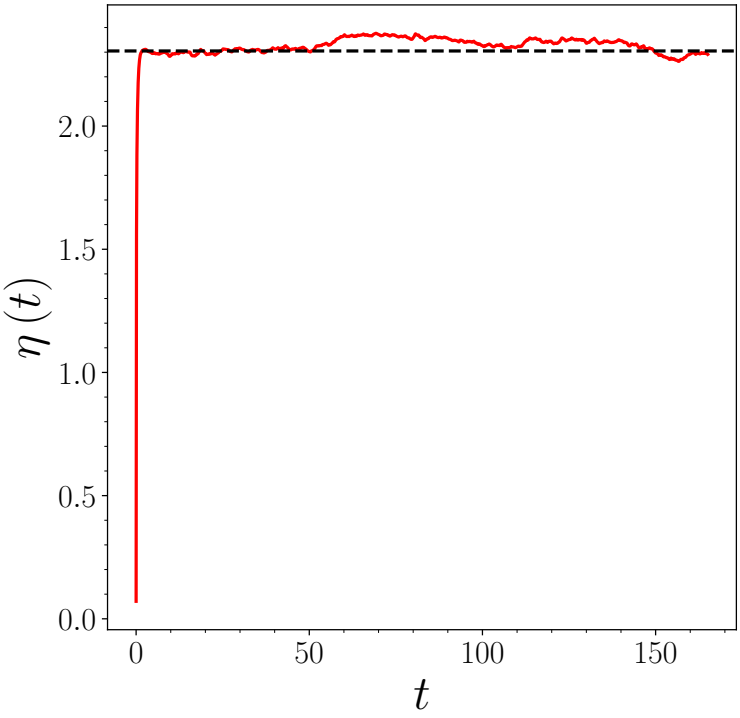


Figure 3.1: Green Kubo calculation for $\rho = 0.8442$ and reduced temperature $T = 0.722$. The horizontal line represents the value at the time autocorrelation crosses zero value. A plateau region is observed around this region, which is taken to be the estimate of the viscosity.

The value of the viscosity obtained from Green Kubo method is $\eta = 2.29 \pm 0.12$. This can be compared with the work of Todd et al [8] which reported $\eta = 2.32 \pm 0.01$

3.2 Helfand Moment Method

Helfand Moment method is based on the Einstein Relations [9]. Just like the diffusion constant is related to the mean squared displacement by the Einstein-Helfand formula:

$$D = \lim_{t\to\infty} \frac{\left\langle [x(t) - x(0)]^2 \right\rangle}{2t} \tag{3.5}$$

other transport quantities can also be expressed as mean squared displacements of the corresponding Helfand Moments. For the case of viscosity, we define the Helfand moment $A_{\alpha\beta}(t)$ as:

$$\begin{aligned} \frac{dA_{\alpha\beta}(t)}{dt} &= \sigma_{\alpha\beta}(t) V \\ A_{\alpha\beta}(t) - A_{\alpha\beta}(0) &= V \int_0^t dt' \sigma_{\alpha\beta}(t) \end{aligned} \tag{3.6}$$

where, $\alpha, \beta \in (x, y, z)$ are the spatial coordinates. The viscosity is then given by the mean squared displacement of the Helfand Moment:

$$\eta = \frac{1}{Vk_BT} \lim_{t\to\infty} \frac{\left\langle [A_{\alpha\beta}(t) - A_{\alpha\beta}(0)]^2 \right\rangle}{2t} \tag{3.7}$$

For better statistics, the Helfand Moment in Equation (3.7) is component averaged.

$$\eta = \frac{1}{Vk_BT} \lim_{t\to\infty} \frac{\left\langle [A_{\alpha\beta}(t) - A_{\alpha\beta}(0)]^2 \right\rangle_{\text{avg}}}{2t} \tag{3.8}$$

where, the component averaged Helfand Moment is given by:

$$\left\langle [A_{\alpha\beta}(t) - A_{\alpha\beta}(0)]^2 \right\rangle_{\text{avg}} = \frac{1}{3} \left[\left\langle [A_{xy}(t) - A_{xy}(0)]^2 \right\rangle + \left\langle [A_{yz}(t) - A_{yz}(0)]^2 \right\rangle + \left\langle [A_{zx}(t) - A_{zx}(0)]^2 \right\rangle \right]$$

The Equation (3.7) can be rewritten as:

$$\eta(t)\, t = \frac{1}{Vk_BT} \lim_{t\to\infty} \frac{\left\langle [A_{\alpha\beta}(t) - A_{\alpha\beta}(0)]^2 \right\rangle}{2} \tag{3.9}$$

Viscosity is obtained from the plot of Equation (3.9) as a function of time. The observed behavior is $\eta(t)\, t \propto t^2$ for short times and $\eta(t)\, t \propto t$ at longer times. The slope of the linear regime is the viscosity estimate. A test simulation was performed for the case of WCA potential at reduced density $\rho = 0.8442$ and reduced temperature $T = 0.722$. The viscosity

thus obtained is $\eta = 2.322 \pm 0.002$. The error is obtained from the uncertainty in the least squared fit. In the Figure 3.2, the log-log plot gives the exponents at the short time as 1.960 ± 0.006 and at the long times 1.140 ± 0.001. These are in agreement with the quadratic behavior in the short times and the linear behavior in the long times.

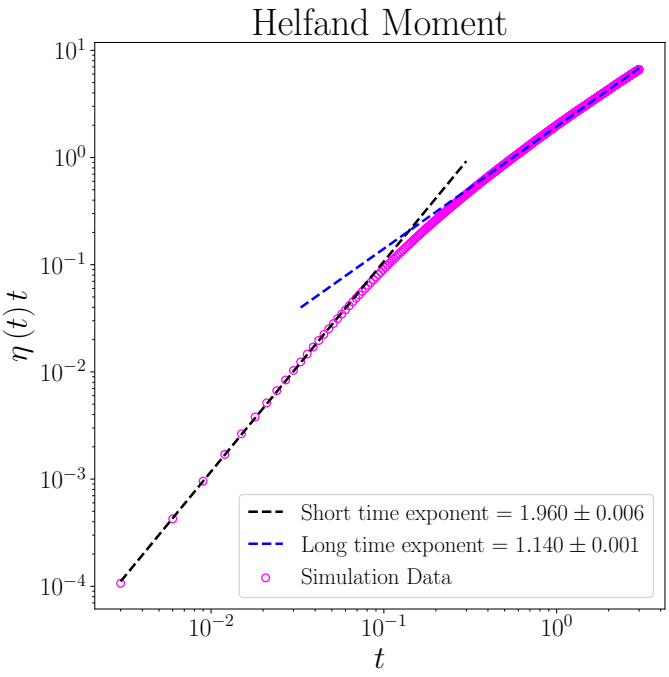


Figure 3.2: Helfand moment simulation for $\rho = 0.8442$ and reduced temperature $T = 0.722$. It is observed that the exponent changes from 2 to 1, implying quadratic to linear behavior in the long time.

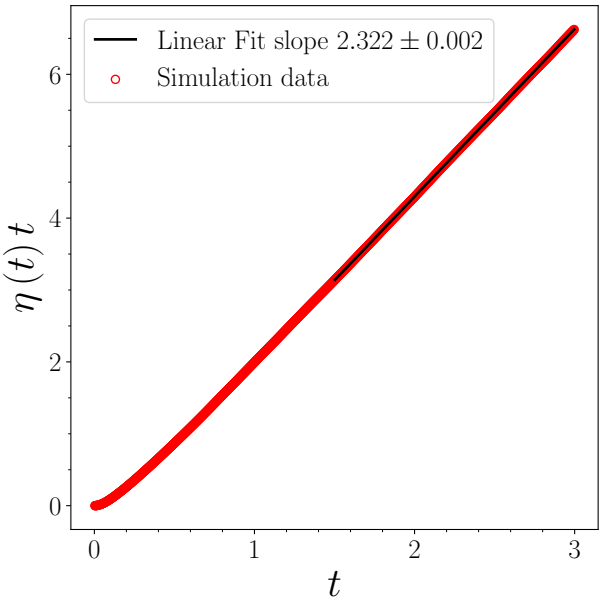


Figure 3.3: Helfand moment fit for the WCA potential t $T = 0.722$ and $\rho = 0.8442$. The black line is the linear fit in the long time limit. The slope of this black line is the viscosity.

3.3 Finite shear rate simulations through SLLOD

The methods discussed in previous two sections are equilibrium viscosity methods, where the viscosity is calculated without any applied strain rate. Although transport properties can be calculated from the equilibrium simulations, in principle, the transport properties are experimentally defined as a response to external perturbation. For the case of viscosity, it is experimentally measured by inducing a shear flow as an external driving force. The rate at which the external perturbation is applied defines the shearing rate or strain rate. Here, we consider the case of planar flow, where, the strain rate is defined as the gradient of the velocity field. Suppose that the system is sheared along x direction, then a velocity profile is created along the y direction. Then the strain rate is defined as:

$$\dot{\gamma}_{xy} = \frac{dv_x}{dy} \tag{3.10}$$

where, v_x is the streaming velocity along the x axis. The viscosity can be calculated in the linear regime for the Newtonian fluids as:

$$\eta = \frac{\sigma_{xy}}{\dot{\gamma}_{xy}} \tag{3.11}$$

where, σ_{xy} is the strain developed due to the external shear flow. In practice, to calculate viscosity using Equation (3.11), the stress is averaged over the steady state. Hence, in simulations the steady state stress is calculated to give $\langle \sigma_{xy}(t) \rangle_{\text{neq}}$, where the angular brackets denote average over non-equilibrium steady state. Hence, in nonequilibrium simulations, the stress is given by:

$$\eta = \frac{\langle \sigma_{xy}(t) \rangle_{\text{neq}}}{\dot{\gamma}_{xy}} \tag{3.12}$$

3.3.1 Derivation of SLLOD equations of motion

We derive the SLLOD equations of motion. The derivation has been adapted from [21]. We have the momentum continuity equation:

$$\frac{\partial \mathbf{J}(\mathbf{r},t)}{\partial t} = -\boldsymbol{\nabla} \cdot [\rho(\mathbf{r},t)\,\mathbf{v}(\mathbf{r},t)\,\mathbf{v}(\mathbf{r},t) + \boldsymbol{P}] + \mathbf{f}(\mathbf{r},t) \tag{3.13}$$

where,

$$\rho(\mathbf{r},t)\,\mathbf{v}(\mathbf{r},t) = \mathbf{J}(\mathbf{r},t)$$

Here, $\mathbf{f}(\mathbf{r},t) = n(\mathbf{r},t)\,\mathbf{F}_e$ is the external force per unit volume. We can write the density of the fluid system as:

$$n(\mathbf{r},t) = \sum_i \delta(\mathbf{r} - \mathbf{r}_i)$$

This gives us:

$$\mathbf{f}(\mathbf{r},t) = \sum_i \mathbf{F}_i^e \delta(\mathbf{r} - \mathbf{r}_i) \tag{3.14}$$

Now, we take the Fourier transform of Equation (3.13) with respect to position:

$$\frac{\partial}{\partial t}\widetilde{\mathbf{J}}(\mathbf{k},t) = i\mathbf{k} \cdot \widetilde{\boldsymbol{P}}(\mathbf{k},t) + i\mathbf{k} \cdot [\widetilde{\rho \mathbf{v}\mathbf{v}}(\mathbf{k},t)] + \widetilde{\mathbf{G}}(\mathbf{k},t) \tag{3.15}$$

This tells us that the long wavelength (or zero wavevector) contribution comes from only the external forces. We are considering $\mathbf{k} = \mathbf{0}$ because the average behaviour over entire volume is described by this quantity. Thus, the total momentum change for this case can be written

as:

$$\frac{d}{dt}\sum_i m_i \mathbf{v}_i = \sum_i \mathbf{F}_i^e \tag{3.16}$$

Now, we split the total velocity into two parts:

- Thermal velocity denoted as $\mathbf{c}_i$
- Flow velocity due to external force denoted as $\mathbf{v}(\mathbf{r}_i)$

Assume that the system is initiall in equilibrium at $t = 0$. An external force is applied at the instant $t = 0$ that starts the flow. So that we can write:

$$\mathbf{v}_i = \mathbf{c}_i + \mathbf{v}(\mathbf{r}_i)\,\theta(t) \tag{3.17}$$

Here $\theta(t)$ is Heaviside step function defined as:

$$\theta(t) = \begin{cases} 0, & \text{if } t < 0 \\ 1, & \text{if } t \geq 1 \end{cases}$$

This means that till the time $t = 0$, there is only the thermal velocity for each particle in the fluid. At the instant $t = 0$, there is an additional term independent of the thermal velocity due to the external field. For homogeneous flows, Equation (3.17) can be rewritten as:

$$\mathbf{v}_i = \mathbf{c}_i + \mathbf{r}_i \cdot \boldsymbol{\nabla}\mathbf{v}\theta(t) \tag{3.18}$$

Here, $\boldsymbol{\nabla}\mathbf{v}$ is the gradient of a vector. Hence, it is a second order tensor. It is defined as:

$$\boldsymbol{\nabla}\mathbf{v} = \begin{pmatrix} \frac{\partial v_x}{\partial x} & \frac{\partial v_y}{\partial x} & \frac{\partial v_z}{\partial x} \\ \frac{\partial v_x}{\partial y} & \frac{\partial v_y}{\partial y} & \frac{\partial v_z}{\partial y} \\ \frac{\partial v_x}{\partial z} & \frac{\partial v_y}{\partial z} & \frac{\partial v_z}{\partial z} \end{pmatrix}$$

Since, $\mathbf{c}_i$ is the thermal momentum, that is independent of the flow velocity, and that it is same as that of a system at equilibrium, the total thermal momentum and thus its derivative

both are zero.

$$\begin{aligned} \sum_i m_i \mathbf{c}_i &= \mathbf{0} \\ \frac{d}{dt} \sum_i m_i \mathbf{c}_i &= \mathbf{0} \end{aligned} \tag{3.19}$$

Multiply Equation (3.18) by m_i, take the sum over all particles and and then differentiate with respect to time to obtain:

$$\frac{d}{dt} \sum_i m_i \mathbf{v}_i = \frac{d}{dt} \sum_i m_i \left[\mathbf{c}_i + \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \theta \left(t \right) \right]$$

Now, using Equation (3.19), we obtain

$$\sum_i \mathbf{F}_i^e = \frac{d}{dt} \sum_i m_i \mathbf{v}_i = \frac{d}{dt} \sum_i m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \theta \left(t \right) \tag{3.20}$$

This gives us:

$$\sum_i \mathbf{F}_i^e = \sum_i m_i \mathbf{v}_i \cdot \boldsymbol{\nabla} \mathbf{v} \theta \left(t \right) + \sum_i m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \frac{d}{dt} \theta \left(t \right)$$

Here, the velocity gradient profile is constant in time. Now we use:

$$\frac{d}{dt} \theta \left(t \right) = \delta \left(t \right)$$

Substituting the Equation (3.18) for the $\mathbf{v}_i$ in RHS we get:

$$\sum_i \mathbf{F}_i^e = \sum_i \left[m_i \left\{ \mathbf{c}_i + \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \theta \left(t \right) \right\} \cdot \boldsymbol{\nabla} \mathbf{v} \theta \left(t \right) + m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \delta \left(t \right) \right]$$

Now using Equation (3.19) again and the fact that $\theta \left(t \right) \times \theta \left(t \right) = \theta \left(t \right)$, we get:

$$\sum_i \mathbf{F}_i^e = \sum_i \left[m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \cdot \boldsymbol{\nabla} \mathbf{v} \cdot \theta \left(t \right) + m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \cdot \delta \left(t \right) \right] \tag{3.21}$$

Now we impose the condition that the external force has same functional form for all the particles to obtain:

$$\mathbf{F}_i^e = m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \cdot \boldsymbol{\nabla} \mathbf{v} \cdot \theta \left(t \right) + m_i \mathbf{r}_i \cdot \boldsymbol{\nabla} \mathbf{v} \cdot \delta \left(t \right) \tag{3.22}$$

The external field is applied at the instant $t = 0$, so that, the first term is zero for $t < 0$. From Newton's second law of motion we have:

$$\mathbf{F}_i = m_i \frac{d\mathbf{v}_i}{dt} \tag{3.23}$$

and

$$\mathbf{F}_i = \mathbf{F}_i^e + \mathbf{F}_i^\phi \tag{3.24}$$

Using Equation (3.23) and Equation (3.18) we get:

$$\begin{aligned}\mathbf{F}_i &= m_i \frac{d}{dt} \left[\mathbf{c}_i + \mathbf{r}_i \cdot \boldsymbol{\nabla}\mathbf{v}\theta\left(t\right)\right] \\ &= m_i\dot{\mathbf{c}}_i + m_i\dot{\mathbf{r}}_i \cdot \boldsymbol{\nabla}\mathbf{v}\theta\left(t\right) + m_i\mathbf{r}_i \cdot \boldsymbol{\nabla}\mathbf{v}\delta\left(t\right)\end{aligned}$$

Now $\dot{\mathbf{r}}_i = \mathbf{v}_i$ and use Equation (3.18) to get:

$$\mathbf{F}_i = m_i\dot{\mathbf{c}}_i + m_i\left(\mathbf{c}_i + \mathbf{v}\left(\mathbf{r}_i\right)\theta\left(t\right)\right) \cdot \boldsymbol{\nabla}\mathbf{v}\theta\left(t\right) + m_i\mathbf{r}_i \cdot \boldsymbol{\nabla}\mathbf{v}\delta\left(t\right)$$

Using Equation (3.22) and Equation (3.24) we obtain:

$$m_i\dot{\mathbf{c}}_i = \mathbf{F}_i^\phi - m_i\mathbf{c}_i \cdot \boldsymbol{\nabla}\mathbf{v}\theta\left(t\right) \tag{3.25}$$

For time $t \geq 0$ and combinig with Equation (3.18) we get the SLLOD equations of motion:

$$\begin{aligned}\dot{\mathbf{r}}_i &= \mathbf{c}_i + \mathbf{r}_i \cdot \boldsymbol{\nabla}\mathbf{v} \\ m_i\dot{\mathbf{c}}_i &= \mathbf{F}_i^\phi - m_i\mathbf{c}_i \cdot \boldsymbol{\nabla}\mathbf{v}\end{aligned} \tag{3.26}$$

If we express the thermal momentum $m_i\dot{\mathbf{c}}_i = \mathbf{p}_i$, we get

$$\begin{aligned}\dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \mathbf{r}_i \cdot \boldsymbol{\nabla}\mathbf{v} \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i^\phi - m_i\mathbf{c}_i \cdot \boldsymbol{\nabla}\mathbf{v}\end{aligned} \tag{3.27}$$

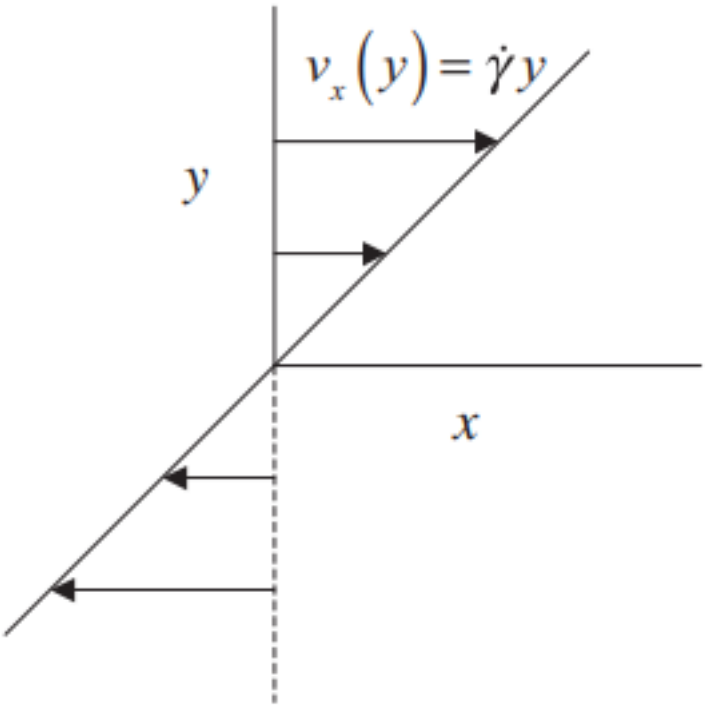


Figure 3.4: Velocity profile for shear flow along x direction with a gradient along y direction [21]

For shear flow, in x direction with a velocity gradient in the y direction (see Figure 3.4), we have:

$$\frac{dv_x}{dy} = \dot{\gamma}_{xy}$$

So that we have:

$$\boldsymbol{\nabla}\mathbf{v} = \begin{pmatrix} 0 & 0 & 0 \\ \dot{\gamma}_{xy} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

This gives us the SLLOD equations for the shear flow:

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \hat{\mathbf{x}}\dot{\gamma}_{xy} y_i \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i^{\phi} - \hat{\mathbf{x}}\dot{\gamma}_{xy} p_{y_i} \end{aligned} \tag{3.28}$$

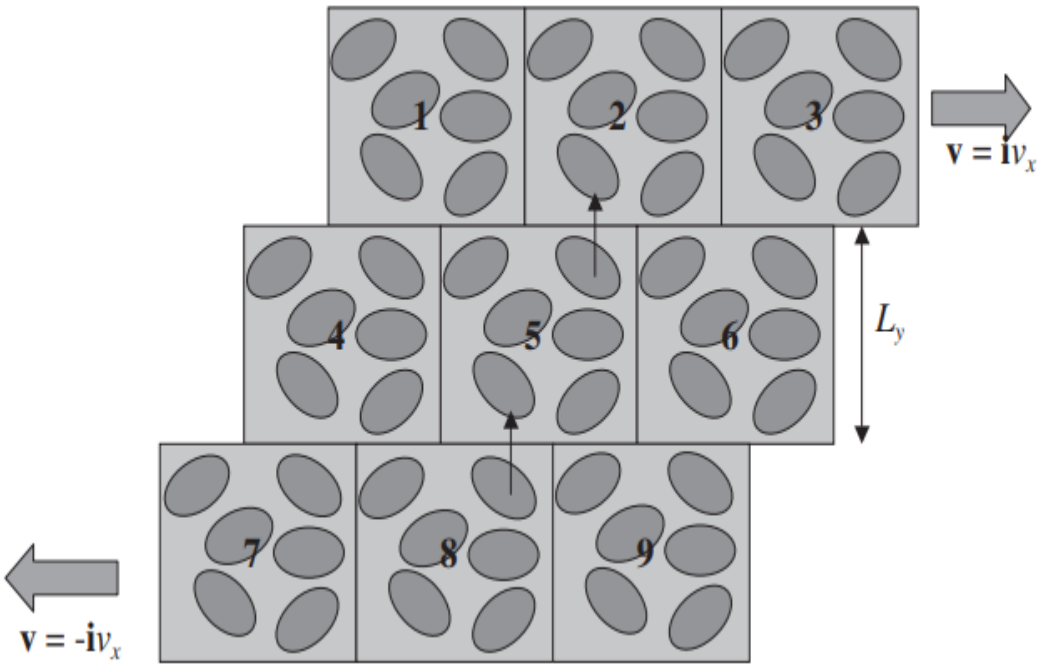


Figure 3.5: Lees Edwards periodic boundary conditions [21]

3.3.2 Lees Edwards periodic boundary conditions

For the proper implementation of the SLLOD equations for the shear flow, we must employ periodic boundary conditions that are compatible with the equations of motion. One such method is proposed by the Lees and Edwards [14]. For zero shear rate, we have the normal periodic boundary conditions for cubic simulation cell. As we apply the shearing force with some shear rate $\dot{\gamma}_{xy}$, there is a velocity gradient generated along say y direction. This is implemented by sliding layers of simulation boxes (stacked along the y direction on top of each other and extending in the x direction) on top of each other. This is shown in Figure 3.5:

In the Figure 3.5, the central box is labelled as 5. A particle that leaves this box from the top, that is, from $y = L_y$, would re-enter back from box labelled 8. Due to the shear, the particle is also displaced along the x axis by an amount $-L_y\dot{\gamma}_{xy}\Delta t$, where, Δt is the integration time step. There is no shift if the particle exits from $x = L_x$ or $z = L_z$. Simillarly, if the particle exits from $y = -L_y$ into box 8, it would re-enter from box 2 but shifted by an amount $L_y\dot{\gamma}_{xy}\Delta t$ in x direction. It must be noted that when $\dot{\gamma}_{xy}t = 1$ (t is the simulation time), there is a displacement of one box length. This situation is equivalent to having zero strain and the shearing starts all over again from zero value.

Wrapping (PBC)

The wrapping of particles is done as follows:

$$
\begin{aligned}
x\left(i\right) &= x\left(i\right) - \delta x \text{ NINT}\left[\frac{y\left(i\right)}{L_y}\right] \times L_y \\
x\left(i\right) &= x\left(i\right) - \text{NINT}\left[\frac{x\left(i\right)}{L_x}\right] \times L_x \\
y\left(i\right) &= y\left(i\right) - \text{NINT}\left[\frac{y\left(i\right)}{L_y}\right] \times L_y \\
z\left(i\right) &= z\left(i\right) - \text{NINT}\left[\frac{z\left(i\right)}{L_z}\right] \times L_z
\end{aligned}
$$

Nearest Image Convention (NIC)

The nearest image convention compatible with the Lees Edwards PBCs is implemented as follows:

$$
\begin{aligned}
dx &= x\left(i\right) - x\left(j\right) \\
dy &= y\left(i\right) - y\left(j\right) \\
dz &= z\left(i\right) - z\left(j\right) \\
dx &= dx - \delta x \text{ NINT}\left[\frac{dy}{L_y}\right] \times L_y \\
dx &= dx - \text{NINT}\left[\frac{dx}{L_x}\right] \times L_x \\
dy &= dy - \text{NINT}\left[\frac{dy}{L_y}\right] \times L_y \\
dz &= dz - \text{NINT}\left[\frac{dz}{L_z}\right] \times L_z
\end{aligned}
$$

3.3.3 Nosé Hoover Thermostatted SLLOD

$$
\begin{aligned}
\dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \hat{\mathbf{x}}\dot{\gamma}_{xy} y_i \\
\dot{\mathbf{p}}_i &= \mathbf{F}_i^{\phi} - \hat{\mathbf{x}}\dot{\gamma}_{xy} p_{y_i} - \mathbf{p}_i \frac{p_\eta}{Q} \\
\dot{\eta} &= \frac{p_\eta}{Q} \\
\dot{p_\eta} &= \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} - \frac{L}{\beta}
\end{aligned}
\tag{3.29}
$$

The integration of these equations of motion can be done using the operator splitting method. Let, $\mathbf{\Gamma} = \{\mathbf{r}_i, \mathbf{p}_i, \eta, p_\eta\}$ denote the phase space variables. Then the short time approximation becomes:

$$\mathbf{\Gamma}(\Delta t) = \exp\left[iL_{\mathrm{SLLOD}}\Delta t\right]\mathbf{\Gamma}(0)$$

The total Liouvillian of the system is:

$$iL_{\mathrm{SLLOD}} = iL_{\mathrm{NH}} - \hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\cdot\frac{\partial}{\partial\mathbf{v}_i} + iL_{\mathrm{mvV}}$$

with

$$iL_{\mathrm{mvV}} = \frac{\mathbf{F}_i^\phi}{m_i}\cdot\frac{\partial}{\partial\mathbf{v}_i} + (\mathbf{v}_i + \hat{\mathbf{x}}\dot{\gamma}_{xy}y_i)\cdot\frac{\partial}{\partial\mathbf{r}_i}$$
$$iL_{\mathrm{NH}} = -\sum_{i=1}^{N} v_\eta\mathbf{v}_i\cdot\frac{\partial}{\partial\mathbf{v}_i} + G\frac{\partial}{\partial v_\eta} + v_\eta\frac{\partial}{\partial\eta}$$

and

$$G = \frac{1}{Q}\left(\sum_{i=1}^{N} m_i\mathbf{v}_i^2 - \frac{L}{\beta}\right)$$

The symbols have usual meaning as the Nosé Hoover thermostat in the equilibrium simulation. Using the Trotter factorization, we get:

$$\exp\left(iL_{\mathrm{SLLOD}}\Delta t\right) = \exp\left[\frac{\Delta t}{2}\left(iL_{\mathrm{NH}} - \hat{\mathbf{x}}\dot{\gamma}_{xy}y_i\frac{\partial}{\partial\mathbf{v}_i}\right)\right]\times\exp\left(\Delta t iL_{\mathrm{mvV}}\right)\times\exp\left[\frac{\Delta t}{2}\left(iL_{\mathrm{NH}} - \hat{\mathbf{x}}\dot{\gamma}_{xy}y_i\frac{\partial}{\partial\mathbf{v}_i}\right)\right]$$

The mvV operator can be split as:

$$\exp\left(iL_{\mathrm{mvV}}\Delta t\right) = \exp\left(\frac{\Delta t}{2}\frac{\mathbf{F}_i^\phi}{m_i}\cdot\frac{\partial}{\partial\mathbf{v}_i}\right)\times\exp\left[\Delta t\left(\mathbf{v}_i + \hat{\mathbf{x}}\dot{\gamma}_{xy}y_i\right)\cdot\frac{\partial}{\partial\mathbf{r}_i}\right]\times\exp\left(\frac{\Delta t}{2}\frac{\mathbf{F}_i^\phi}{m_i}\cdot\frac{\partial}{\partial\mathbf{v}_i}\right)$$

The central part of above mvV split needs to be further factorized as:

$$(\mathbf{v}_i + \hat{\mathbf{x}}\dot{\gamma}_{xy}y_i)\cdot\frac{\partial}{\partial\mathbf{r}_i} = (v_{x_i} + \dot{\gamma}_{xy}y_i)\frac{\partial}{\partial x} + v_{y_i}\frac{\partial}{\partial y} + v_{z_i}\frac{\partial}{\partial z}$$

The first two operators in the RHS of above equation do not commute. So the the Trotter

factorisation for the position coordinates is:

$$\exp\left[\Delta t\left(\mathbf{v}_i+\hat{\mathbf{x}}\dot{\gamma}_{xy}y_i\right)\cdot\frac{\partial}{\partial\mathbf{r}_i}\right]=\exp\left(\frac{\Delta t}{2}v_{y_i}\frac{\partial}{\partial y_i}\right)\times\exp\left[\Delta t\left(v_{x_i}+\dot{\gamma}_{xy}y_i\right)\frac{\partial}{\partial x_i}\right]\times\exp\left(\Delta t v_{z_i}\frac{\partial}{\partial z_i}\right)\times\exp\left(\frac{\Delta t}{2}v_{y_i}\frac{\partial}{\partial y_i}\right)$$

This can be converted into an algorithm as Algorithm 8:

Algorithm 8 Modified Velocity Verlet SLLOD

for i = 1 to N **do**
  $\mathbf{v}_i \to \mathbf{v}_i + \frac{\mathbf{F}_i}{m_i}\frac{\Delta t}{2}$
  $y_i \to y_i + v_{y_i}\frac{\Delta t}{2}$
  $x_i \to x_i + \left(v_{x_i}+\dot{\gamma}_{xy}y_i\right)\Delta t$
  $z_i \to z_i + v_{z_i}\Delta t$
  $y_i \to y_i + v_{y_i}\frac{\Delta t}{2}$
end for
Apply PBC for wrapping ▷ section 3.3.2
Calculate Forces ▷ Use nearest image convention section 3.3.2
for i = 1 to N **do**
  $\mathbf{v}_i \to \mathbf{v}_i + \frac{\mathbf{F}_i}{m_i}\frac{\Delta t}{2}$
end for

Now consider the operator

$$iL_{\text{SLLOD}} = iL_{\text{NH}} - \hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\cdot\frac{\partial}{\partial\mathbf{v}_i}$$

So that the time evolution can be obtained from:

$$\exp\left[\frac{\Delta t}{2}\left(iL_{\text{NH}}-\hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\cdot\frac{\partial}{\partial\mathbf{v}_i}\right)\right]=\exp\left[\frac{\Delta t}{2}\left\{\left(-v_\eta\mathbf{v}_i-\hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\right)\cdot\frac{\partial}{\partial\mathbf{v}_i}+v_\eta\frac{\partial}{\partial\eta}+G\frac{\partial}{\partial v_\eta}\right\}\right]$$

A Trotter factorization for this operator can then be written as:

$$\exp\left[\frac{\Delta t}{2}\left(iL_{\text{NH}}-\hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\cdot\frac{\partial}{\partial\mathbf{v}_i}\right)\right]=\exp\left(\frac{\Delta t}{4}G\frac{\partial}{\partial v_\eta}\right)\times\exp\left[-\frac{\Delta t}{2}\left(v_\eta\mathbf{v}_i+\hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\right)\cdot\frac{\partial}{\partial\mathbf{v}_i}\right]\times\exp\left(\frac{\Delta t}{2}v_\eta\frac{\partial}{\partial\eta}\right)\times\exp\left(\frac{\Delta t}{4}G\frac{\partial}{\partial v_\eta}\right)$$

Now,

$$\left(v_\eta\mathbf{v}_i+\hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\right)\cdot\frac{\partial}{\partial\mathbf{v}_i}=\left(v_\eta v_{x_i}+\dot{\gamma}_{xy}v_{y_i}\right)\frac{\partial}{\partial v_{x_i}}+v_\eta v_{y_i}\frac{\partial}{\partial v_{y_i}}+v_\eta v_{z_i}\frac{\partial}{\partial v_{z_i}}$$

With this we have the Trotter factorization of the square bracketed term as:

$$\exp\left[-\frac{\Delta t}{2}\left(v_\eta \mathbf{v}_i + \hat{\mathbf{x}}\dot{\gamma}_{xy}v_{y_i}\right)\cdot\frac{\partial}{\partial \mathbf{v}_i}\right] = \exp\left(-\frac{\Delta t}{4}\left(v_\eta v_{x_i} + \dot{\gamma}_{xy}v_{y_i}\right)\frac{\partial}{\partial v_{x_i}}\right)\times\exp\left(-\frac{\Delta t}{2}v_\eta v_{y_i}\frac{\partial}{\partial v_{y_i}}\right)\times \exp\left(-\frac{\Delta t}{2}v_\eta v_{z_i}\frac{\partial}{\partial v_{z_i}}\right)\times\exp\left(-\frac{\Delta t}{4}\left(v_\eta v_{x_i} + \dot{\gamma}_{xy}v_{y_i}\right)\frac{\partial}{\partial v_{x_i}}\right)$$

Again, we split the v_{x_i} update operator as follows:

$$\exp\left(-\frac{\Delta t}{4}\left(v_\eta v_{x_i} + \dot{\gamma}_{xy}v_{y_i}\right)\frac{\partial}{\partial v_{x_i}}\right) = \exp\left(-\frac{\Delta t}{8}v_\eta v_{x_i}\frac{\partial}{\partial v_{x_i}}\right)\times\exp\left(-\frac{\Delta t}{4}\dot{\gamma}_{xy}v_{y_i}\frac{\partial}{\partial v_{x_i}}\right)\times \exp\left(-\frac{\Delta t}{8}v_\eta v_{x_i}\frac{\partial}{\partial v_{x_i}}\right)$$

With this the shearing and thermostatting algorithm is given by Algorithm 9

Algorithm 9 LNH SLLOD

Calculate G
$v_\eta \to v_\eta + G\frac{\Delta t}{4}$
$\eta \to \eta + v_\eta\frac{\Delta t}{2}$
for i = 1 to N **do**
  $v_{x_i} \to v_{x_i}\exp\left(-\frac{\Delta t}{8}v_\eta\right)$
  $v_{x_i} \to v_{x_i} - \dot{\gamma}_{xy}v_{y_i}\frac{\Delta t}{4}$
  $v_{x_i} \to v_{x_i}\exp\left(-\frac{\Delta t}{8}v_\eta\right)$
  $v_{z_i} \to v_{z_i}\exp\left(-\frac{\Delta t}{2}v_\eta\right)$
  $v_{y_i} \to v_{y_i}\exp\left(-\frac{\Delta t}{2}v_\eta\right)$
  $v_{x_i} \to v_{x_i}\exp\left(-\frac{\Delta t}{8}v_\eta\right)$
  $v_{x_i} \to v_{x_i} - \dot{\gamma}_{xy}v_{y_i}\frac{\Delta t}{4}$
  $v_{x_i} \to v_{x_i}\exp\left(-\frac{\Delta t}{8}v_\eta\right)$
end for
calculate G
$v_\eta \to v_\eta + G\frac{\Delta t}{4}$

The final SLLOD algorithm can be written as Algorithm 10

Algorithm 10 NVT SLLOD Simulations

Initialize positions and **velocities** ▷ Use equilibrated configurations
Calculate **Forces**
$\delta x = 0$ ▷ Initialize the strain to zero
for $t =$ start $\rightarrow$ end **do**
  $\delta x = \delta x + \dot{\gamma}_{xy}\Delta t$ ▷ Update the strain
  $\delta x = \delta x - \text{int}\,(\delta x)$ ▷ reset to zero after one box length traversed
  Call LNH SLLOD ▷ Algorithm 9
  Call Modified Velocity Verlet SLLOD ▷ Algorithm 8
  Call LNH SLLOD ▷ Algorithm 9
end for ▷ End of simulation

A code was written in FORTRAN and applied on the WCA potential Equation (3.4) with $N = 256$ and $N = 2048$ particles. Simulations are performed at the reduced number density $\rho = 0.8442$ and reduced temperature of $T = 0.722$. The units are in LJ units. For each case 10 independent runs were performed. The initial configuration for the shear simulations were obtained by the equilibrium simulations.

The results are then compared with the work of Travis et al. [11].

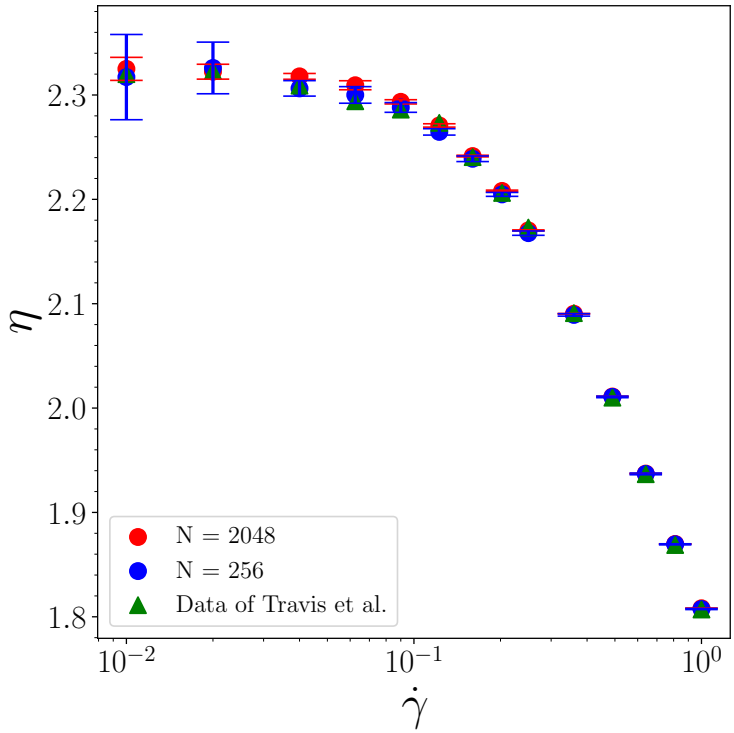


Figure 3.6: SLLOD simulation results for WCA fluid at reduced density $\rho = 0.8442$ and reduced temperature $T = 0.722$

Chapter 4

Methods: Differentiable Simulations

Molecular Dynamics simulations discussed in Chapter 2 forms the basis of calculation of many dynamic properties. However, for the calculation of viscosity at low temperatures, we need to probe dynamics at time scales that are not easily accessible by the methods discussed previously. Here, we propose a new method of viscosity calculation using the differentiable simulations. We first define the viscosity as the derivative of stress with respect to strain rate [4]. This definition has an added advantage that it accounts for the strain rate dependence more accurately. This definition is particularly applicable in the case of vanishing shear rate. The dynamic viscosity, which is defined as the ratio of steady state stress to the applied strain rate η_{dyn}, has the disadvantage that at low shear rates, it diverges to infinity, since the stress does not reduce to zero as fast as the shear rates:

$$\eta_{0,\text{dyn}} = \lim_{\dot{\gamma}\to 0} \frac{\sigma_{xy}}{\dot{\gamma}} \to \infty$$

Thus to calculate the vanishing strain rate viscosity, one needs to do equilibrium simulations and compute viscosity using the Green Kubo formalism. However, the differential viscosity is defined as:

$$\eta_{\text{diff}} = \frac{d\sigma_{xy}}{d\dot{\gamma}} \tag{4.1}$$

This definition can be extended to any shear rate and clearly captures the shear rate dependence of the viscosity.

In this chapter we discuss how the automatic differentiation can be used to compute the differential viscosity. This is implemented in the JAX-MD [18] package in python.

4.1 Automatic Differentiation (AD)

In AD all numerical computations are composed of known functions and operations with known derivatives [23]. Any complicated function can be decomposed into simpler functions and arithmetic operations. These basic operations typically include binary arithmetic operations and sign flips. These simple functions involve polynomial functions, trigonometric functions and transcendental functions. These functions have known derivatives that can be hard coded into the computer program. Further, the chain rule can be implemented into the program allowing to differentiate practically any function that can be analytically differentiated.

Suppose that we have a function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$

$$\mathbf{f}\left(x_1, x_2, \cdots, x_n\right) = \left(y_1, y_2, \cdots, y_m\right)$$

It can be broken down into intermediate steps with variables z_i with dimensionality $p >> m + n$, such that the computation follows the following order:

$$\mathbf{x} = \left(x_1, x_2, \cdots, x_n\right)$$

$$\downarrow$$

$$\mathbf{z} = \left(z_1, z_2, \cdots, z_p\right) \;\; p >> m + n$$

$$\downarrow$$

$$\mathbf{y} = \left(y_1, y_2, \cdots, y_m\right)$$

The intermediate variable z can be written as $z_k = f_{\text{intermediate}}\left(z_i\right)$. This $f_{\text{intermediate}}$ consists of mathematical operations like addition, multiplication, division and sign flip along with known functions like exponentiation, trigonometric functions etc. Further, the intermediate values z are related among themselves:

$$z_k = f^k_{\text{intermediate}}\left(z_i, z_j\right) \;\; i < k \text{ and } j < k$$

Typically the number of intermediate values is larger than the input and output dimensions.

AD works in two modes, forward and reverse modes. The forward mode for the above function can be expressed as:

$$\frac{dz_k}{dx} = \frac{\partial f_{\text{intermediate}}}{\partial z_i}\frac{dz_i}{dx} + \frac{\partial f_{\text{intermediate}}}{\partial z_j}\frac{dz_j}{dx} \tag{4.2}$$

This sum continues for all the intermediate values and the final output value as the chain rule is propagated to yield $\frac{dy}{dx}$.

The reverse mode involves the computation of the derivatives in the reverse order, that is, from y to x. The intermediate values can be written as:

$$z_k = f_{\text{intermediate}}\left(z_i, z_j\right)$$

The chain rule in reverse mode is traversed as:

$$\frac{dy}{dz_i} = \frac{\partial f_{\text{intermediate}}}{\partial z_i}\frac{dy}{dz_k}$$

Similarly this chain is propagated for other intermediate variables as well.

4.2 A quick example

We now present a quick example of application of AD in forward and reverse mode on the function:

$$f\left(x_1, x_2\right) = \ln\left(x_1\right) + x_1 x_2 - \sin\left(x_2\right)$$

The corresponding computational graph is Figure 4.1 :

In Figure 4.1 the variable v_i are:

- $v_{-1} = x_1$
- $v_0 = x_2$
- $v_1 = \ln x_1$
- $v_2 = x_1 x_2$

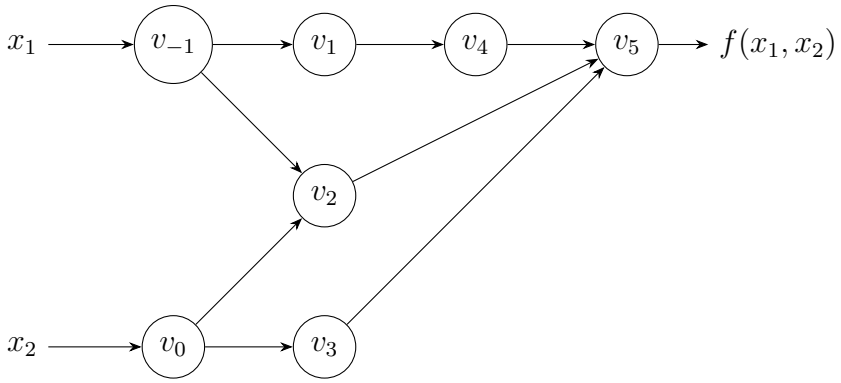


Figure 4.1: Computational graph representing the function $f(x_1, x_2) = \ln(x_1) + x_1 x_2 - \sin(x_2)$.

- $v_3 = \sin x_2$
- $v_4 = \ln x_1 + x_1 x_2$
- $v_5 = \ln x_1 + x_1 x_2 - \sin x_2$

4.2.1 Forward Mode AD

In the forward mode AD we propagate the derivatives in forward direction, that is, from input to outputs. Suppose that in the above function we want to evaluate $\partial f/\partial x_1$ at $(x_1, x_2) = (2, 5)$. We define at node i, $\dot{v}_i = \partial v_i/\partial x_1$ Then we can write:

- $\dot{v}_{-1} = \dot{x}_1 = 1$
- $\dot{v}_{-1} = \dot{x}_1 = 1$
- $\dot{v}_1 = \frac{\dot{v}_{-1}}{v_{-1}} = \frac{1}{2} = 0.5$
- $\dot{v}_2 = \dot{v}_{-1} v_0 + v_{-1} \dot{v}_0 = 1 \times 5 + 2 \times 0 = 5$
- $\dot{v}_3 = \dot{v}_0 \cos v_0$
- $\dot{v}_4 = \dot{v}_1 + \dot{v}_2 = \frac{1}{2} + 5 = 5.5$
- $\dot{v}_5 = \dot{v}_4 - \dot{v}_3 = 5.5 - 0 = 5.5$
- $\dot{f} = \dot{v}_5 = 5.5$

Thus, we obtain the derivative of f with respect to x_1 at $(2, 5)$ as 5.5

4.2.2 Reverse Mode AD

In the reverse mode the derivatives are propagated in backward direction, that is, from output to input nodes. We define adjoint of any node as the partial derivative of that node with respect to the output. The adjoint of any node is represented as:

$$\bar{v}_i = \frac{\partial f}{\partial v_i} \tag{4.3}$$

For any node we have:

$$\bar{v}_i = \bar{v}_j \frac{\partial v_j}{\partial v_i} \tag{4.4}$$

Here, the j-th node is the node fed by node i in the computational graph. If there are multiple nodes that the i-th node feeds into, we take sum over all such nodes. Using this definition of the adjoint we get:

- $\bar{v}_5 = \partial f / \partial v_5 = 1$
- $\bar{v}_4 = \bar{v}_5 \ \partial v_5 / \partial v_4 = 1 \times 1 = 1$
- $\bar{v}_3 = \bar{v}_5 \ \partial v_5 / \partial v_3 = -1 \times 1 = -1$
- $\bar{v}_2 = \bar{v}_4 \ \partial v_4 / \partial v_2 = 1 \times 1 = 1$
- $\bar{v}_1 = \bar{v}_4 \ \partial v_4 / \partial v_1 = 1 \times 1 = 1$
- $\bar{v}_0 = \bar{v}_2 \ \partial v_2 / \partial v_0 + \bar{v}_3 \ \partial v_3 / \partial v_0 = 1 \times v_{-1} + (-1) \times \cos v_0 = 2 - \cos 5$
- $\bar{v}_{-1} = \bar{v}_1 \partial v_1 / \partial v_{-1} + \bar{v}_2 \partial v_2 / \partial v_{-1} = 1 \ \times (1/v_{-1}) + 1 \times v_0 = 0.5 + 5 = 5.5$

The reverse mode AD requires more storage. However, we observe that we obtained both derivatives with respect to x_1 and x_2 in one pass. If the dimensionality of output space is much smaller than the dimensionality of input space, then reverse mode is more efficient as multiple derivatives with respect to the input parameters can be evaluated in single pass.

4.3 AD to calculate the viscosity

In the MD simulations, the equations of motion govern the per particle position and velocity updates. The time evolution of the atomic system in MD is derived by integration of the

equations of motion through the discrete update rules at small time step. The update rule can be written as [24]:

$$x_t = F_{\dot{\gamma}}\left(x_{t-1}, v_{t-1}, \eta_{t-1}\right) dt + x_{t-1} \tag{4.5}$$

Here, F is the time evolution function, x_t is the position at time t, v is the velocity and η is the thermostat variable and $\dot{\gamma}$ is the strain rate, a parameter of the simulation. To obtain the viscosity, we need to get the derivative of the stress with respect to the strain rate. Consider the gradient of stress σ_{xy} which is the output of the simulation. At each step we have:

$$\begin{aligned} \frac{d\sigma_{xy}}{dx_t} &= \frac{\partial \sigma_{xy}}{\partial x_t} + \frac{d\sigma_{xy}}{dx_{t+1}} \\ &= \frac{\partial \sigma_{xy}}{\partial x_t} + \frac{d\sigma_{xy}}{dx_{t+1}} + \frac{d\sigma_{xy}}{dx_{t+1}} \frac{\partial F}{\partial x_t} dt \end{aligned} \tag{4.6}$$

With the $\frac{d\sigma_{xy}}{dx_t}$ computed at every step, we get the gradient of σ_{xy} accumulated from the chain rule,

$$\frac{d\sigma_{xy}}{d\dot{\gamma}} = \sum_t \frac{d\sigma_{xy}}{dx_t} \frac{dx_t}{d\dot{\gamma}} = \sum_t \frac{d\sigma_{xy}}{dx_t} \frac{dF_{\dot{\gamma}}\left(x_{t-1}, \cdots\right)}{d\dot{\gamma}} dt \tag{4.7}$$

This accumulated gradient gives the viscosity through the differentiable simulations. This method provides a way to calculate viscosity without the need to invoke autocorrelation function calculations and the assumptions of Linear Response theory.

4.4 Finite difference method

In this section we present the finite difference simulation results for the case of viscosity. We performed the simulations at two strain rates.

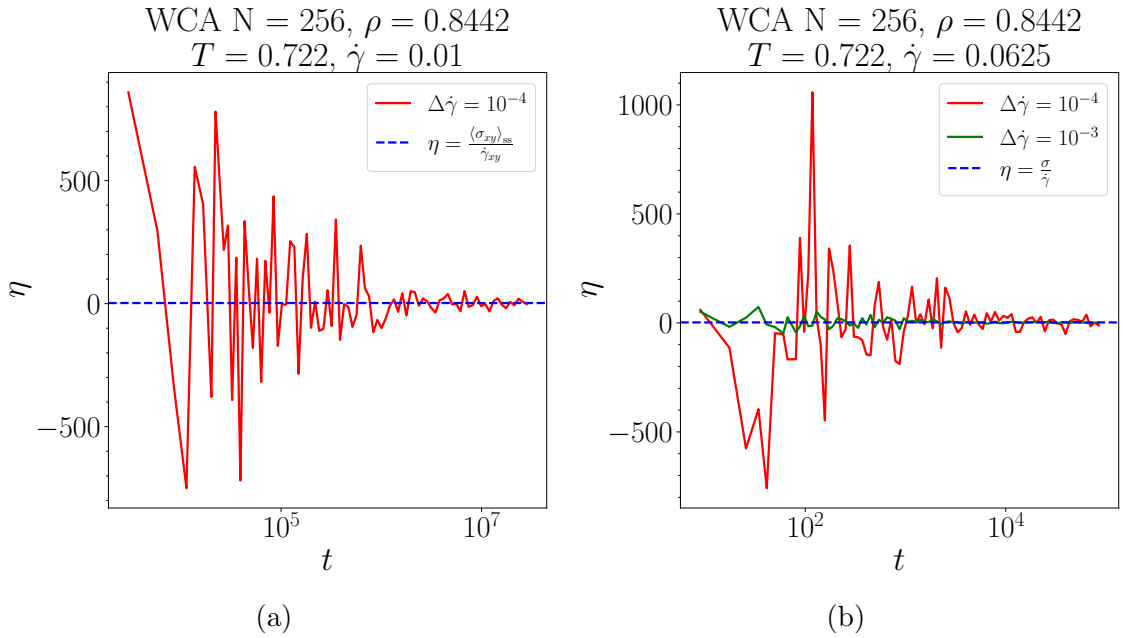


Figure 4.2: **(a)** Differential viscosity using the finite difference method for $\dot{\gamma}_{xy} = 0.01$ and **(b)** $\dot{\gamma} = 0.0625$

$\dot{\gamma}_{xy}$	$\Delta\dot{\gamma}_{xy}$	η_{diff}	η_{dyn}
0.01	10^{-4}	2.5725	2.4075
0.0625	10^{-4}	3.19146	2.2788
0.0625	10^{-3}	1.9812	2.2788

Table 4.1: Values of differential viscosity using finite differences

4.5 Results

We now demonstrate the viscosity calculation using the above method of AD. We use a simple system of WCA potential at the state point $T = 0.722$ and $\rho = 0.8442$. We calculate the viscosity at different strain rates.

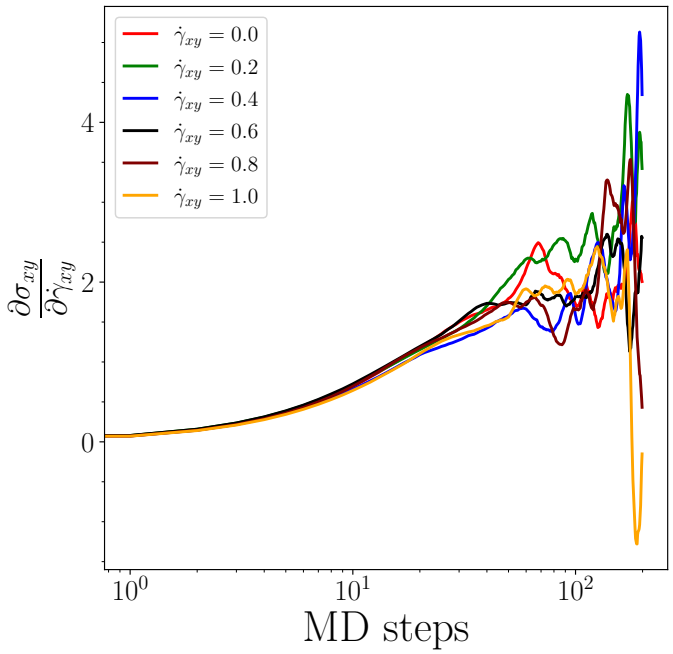


Figure 4.3: Differential viscosity at different strain rates

The viscosity values obtained using the Automatic Differentiation diverge at long times.

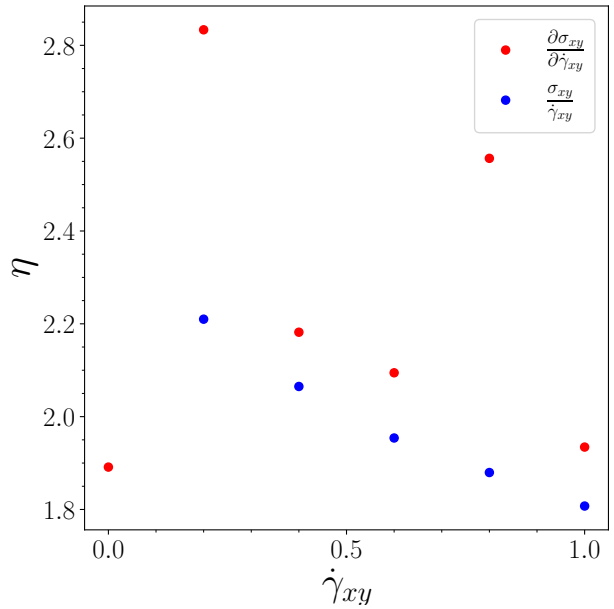


Figure 4.4: Comparison between the differential viscosity and the dynamic viscosity. The values are given in Table 4.2

$\dot{\gamma}_{xy}$	η_{dyn}	η_{diff}
0.0	2.32 (GK Value)	1.8193
0.2	2.21	2.8335
0.4	2.065	2.1819
0.6	1.954	2.0945
0.8	1.8796	2.5565
1.0	1.8074	1.9344

Table 4.2: Values of differential viscosity using Automatic Differentiation

4.6 Comparison with Cross function

The Cross function fits to the viscosity vs strain rate curve and is of the form:

$$\eta\left(\dot{\gamma}\right) = \frac{\eta_0 - \eta_\infty}{1 + \left(K\dot{\gamma}\right)^m} + \eta_\infty \tag{4.8}$$

Here, η_0 is the viscosity in the zero strain limit, η_∞ is the viscosity in the large strain rate limit, K and m are other fit parameters. Our code is compared to the Cross fit using the Newtonian viscosity definition $\eta = \sigma/\dot{\gamma}$

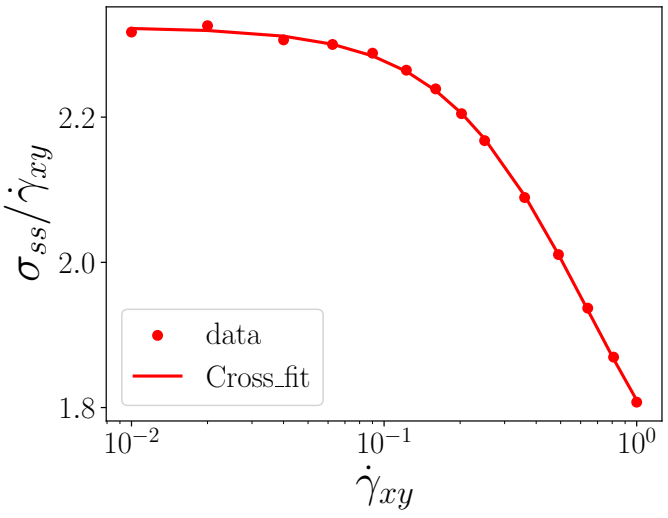


Figure 4.5: Comparison between the Newtonian viscosity and the Cross fit for the WCA potential.

The stress can be obtained in Equation (4.8) by multiplying LHS and RHS by $\dot{\gamma}$.

$$\sigma_{xy} = \dot{\gamma}\frac{\eta_0 - \eta_\infty}{1 + (K\dot{\gamma})^m} + \eta_\infty\dot{\gamma} \tag{4.9}$$

Differentiating the stress in Equation (4.9), we get the expression for the differential viscosity.

$$\frac{d\sigma_{xy}}{d\dot{\gamma}_{xy}} = \eta_\infty + \frac{\eta_0 - \eta_\infty}{\left[1 + (K\dot{\gamma})^m\right]^2}\left[1 + (K\dot{\gamma})^m(1 - m)\right] \tag{4.10}$$

We compare the analytic Cross model differential viscosity in Equation (4.10) and our results in the differentiable simulations.

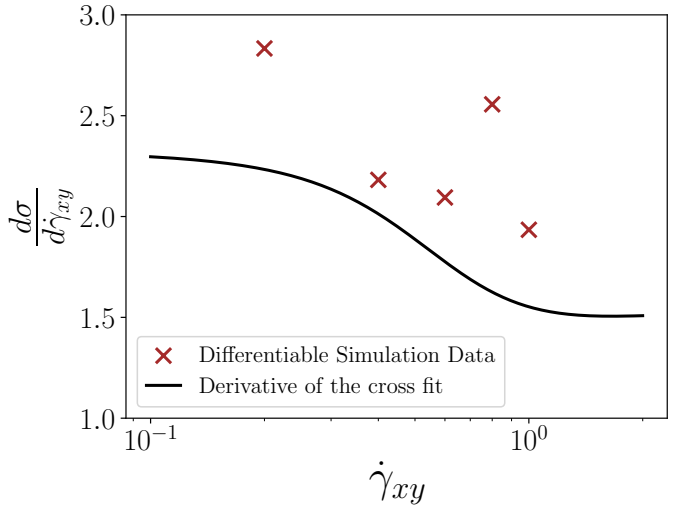


Figure 4.6: Comparison between the derivative of Cross stress and the differentiable simulations

Although the results do not match qualitatively, the computed values of viscosity from both MD and differentiable simulations are within the same order of magnitude. This disagreement could be due to small number of samples.

4.7 Gradient Unrolling

In this section, we describe the concept of gradient unrolling which is crucial in calculation of gradients through automatic differentiation. Let $S(\dot{\gamma}_{xy}, t)$ be the differential simulation that evolves state of the system in discrete time steps of Δt (Δt is the MD time step), and outputs the stress as a function of strain rate $\sigma_{xy}(\dot{\gamma}_{xy}, t)$. The stress at the time T can be

written as:

$$\sigma_{xy}\left(\dot{\gamma}_{xy}, T\right) = S\left(\dot{\gamma}_{xy}, T\right) \tag{4.11}$$

The differential viscosity $d\sigma_{xy}/d\dot{\gamma}_{xy}$ is computed by unrolling the gradient through successive time steps, applying the chain rule at each step. The stress at each time step t, depends on the previous time step's state through a time evolution function $f\left(\dot{\gamma}_{xy}, t\right)$. Thus, the stress at time t can be written as:

$$\sigma_{xy}\left(t\right) = f\left(\dot{\gamma}_{xy}, t-1\right) \tag{4.12}$$

The stress at some time T can be related to the initial time $t = 0$ through the application of f T-times.

$$\sigma_{xy}\left(T\right) = f^{(T)}\left(\dot{\gamma}_{xy}\right) \tag{4.13}$$

where,

$$f^{(T)}\left(\dot{\gamma}_{xy}\right) = \underbrace{\left(f \circ f \circ \cdots \circ f\right)}_{T-\text{times}}\left(\dot{\gamma}_{xy}, t = 0\right)$$

The derivative $d\sigma_{xy}/d\dot{\gamma}_{xy}$ is then obtained as:

$$\frac{d\sigma_{xy}}{d\dot{\gamma}_{xy}} = \frac{\partial f^{(T)}\left(\dot{\gamma}_{xy}\right)}{\partial \dot{\gamma}_{xy}} = \sum_{t=0}^{T} \frac{\partial \sigma_{xy}\left(T\right)}{\partial \sigma_{xy}\left(t\right)} \cdot \frac{\partial \sigma_{xy}\left(t\right)}{\partial \dot{\gamma}_{xy}} \tag{4.14}$$

This represents the application of the chain rule across the T time steps, reflecting the evolution of stress as a function of strain rate at each step. It captures the sensitivity of the stress with respect to the strain rate through automatic differentiation, across multiple time steps in differentiable simulations.

The viscosity calculation through the use of automatic differentiation can possibly be improved by calculating the gradient through different number of unrolling steps and evaluating the sensitivity of the gradient on the number of unrolled steps over a short trajectory. The gradient can then be averaged over different trajectories starting from different initial configurations.

4.8 Further considerations

- Determining clear convergence and identifying the optimal conditions for averaging can be challenging. Averaging over multiple short trajectories may yield more consistent results.
- For a given trajectory, the gradient may be sensitive to unrolling steps. Thus, average over multiple unrolling steps for a single trajectory may be needed.

Chapter 5

Results: Stress-strain rate curves for the Kob-Andersen BMLJ glass former

In this chapter we present the main results. We have performed simulations on modified Kob Andersen Potential [5]

$$\phi_{\alpha\beta}(r) = \begin{cases} 4\epsilon_{\alpha\beta}\left[\left(\frac{\sigma_{\alpha\beta}}{r}\right)^{12} - \left(\frac{\sigma_{\alpha\beta}}{r}\right)^{6}\right] + 4\epsilon_{\alpha\beta}\left[c_0 + c_2\left(\frac{r}{\sigma_{\alpha\beta}}\right)^{2}\right], & r_{\alpha\beta} \leq r_{c\alpha\beta} \\ 0, & r_{\alpha\beta} > r_{c\alpha\beta} \end{cases} \tag{5.1}$$

We first calculated the Green Kubo viscosity at higher temperatures as outlined in the chapter 3.

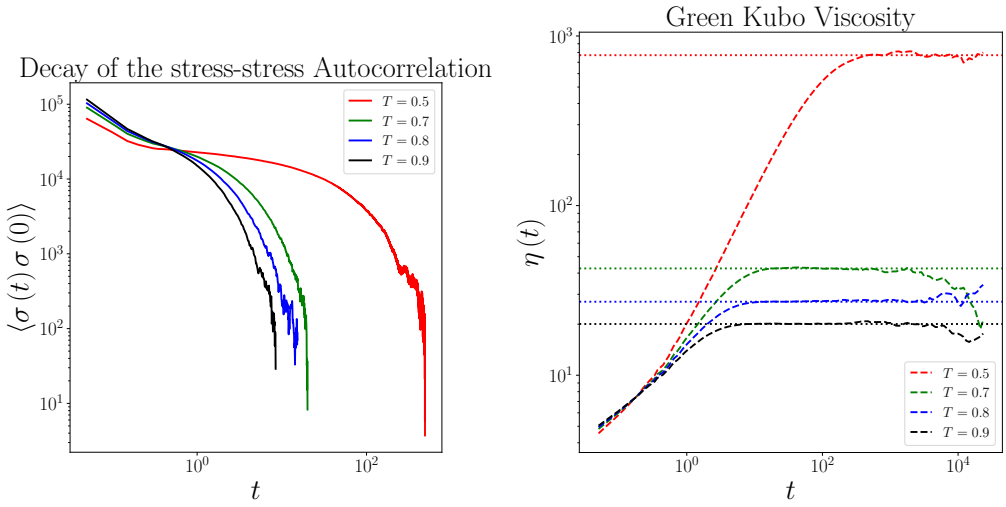


Figure 5.1: **(a)** The stress-stress autocorrelation function for the KABMLJ liquid. **(b)** The cumulative sum (integral) of the stress-stress autocorrelation. The horizontal lines represent the time of approximate plateau where the autocorrelation function first reaches zero. The viscosity is obtained by averaging in this region.

The Green Kubo equilibrium viscosity values obtained in this thesis using the method of [25] is compared with [19]. In this work the error is obtained from standard deviation in the plateau region. The Helfand moment method has been employed as outlined in the chapter 3. For the Helfand moment, each trajectory is subdivided into a tenth of its total length. The stress is then integrated over each of those blocks and then averaged over all blocks. This gives a single run value of the viscosity. 30 such independent runs are performed and and averaged over. The mean value is the estimate of the viscosity and the standard error over these runs is the error estimate.

T	Helfand Moment	Green Kubo	Work of [19]
0.5	784.490 ± 21.292	784.137 ± 18.701	NA
0.7	41.604 ± 1.489	41.496 ± 1.516	42.642 ± 0.741
0.8	26.686 ± 1.188	26.762 ± 1.279	26.433 ± 0.432598
0.9	18.608 ± 0.729	20.016 ± 0.452	18.94 ± 0.451176

Table 5.1: Equilibrium viscosity values for the higher temperatures using Green Kubo and Helfand moment method.

Then we performed SLLOD simulations on the KABMLJ liquid at both low and high temperatures. The equilibrated configurations were provided by Das and Sastry. et al work [5]. The simulations were performed in the LAMMPS package [20]. The stress was calculated

with time. The stress is fitted to an analytical function of the form:

$$\sigma_{xy}(t) = \frac{At - b}{1 + \exp\left(\frac{t - t_1}{\tau}\right)} + \frac{b}{1 + \exp\left(-\frac{t_1}{\tau}\right)} \tag{5.2}$$

This form has four fit parameters namely A, τ, t_1 and b. The τ is the time scale of stress relaxation. The function behaves linearly in short time and in the long time reaches a steady state stress value. The steady state stress can be obtained from the fit parameters:

$$\sigma_{ss} = \frac{b}{1 + \exp\left(-\frac{t_1}{\tau}\right)} \tag{5.3}$$

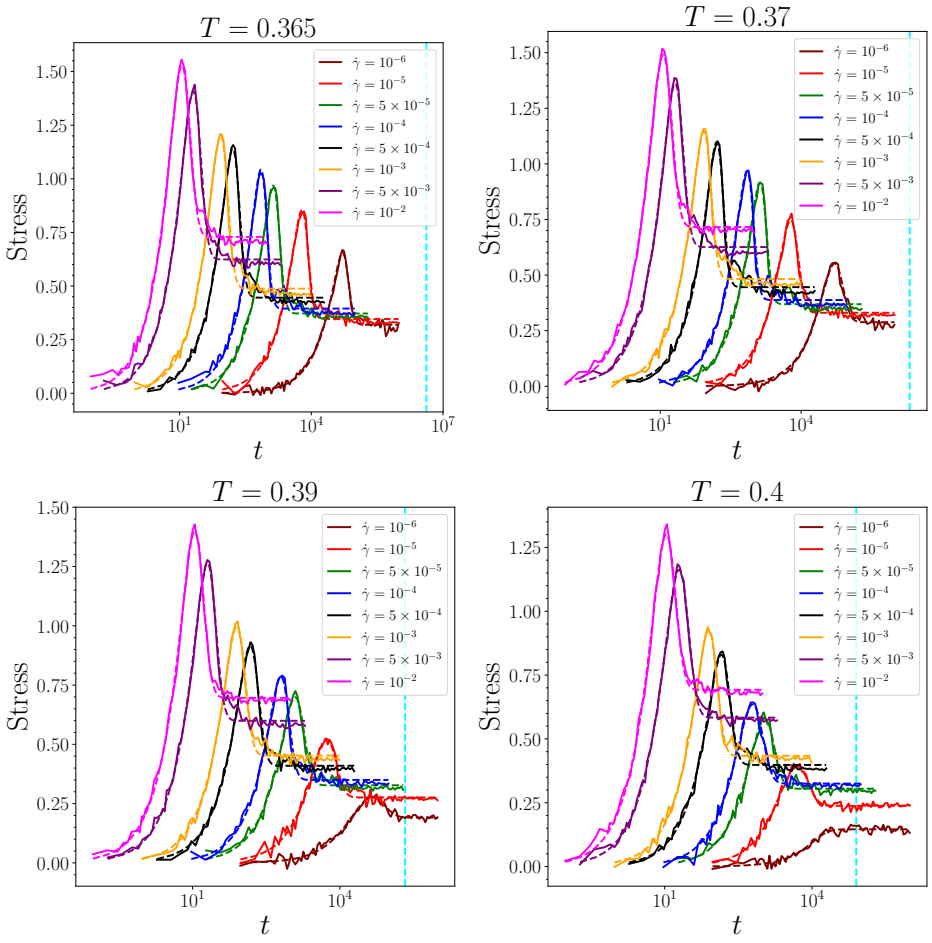


Figure 5.2: Stress vs time data for temperatures 0.365, 0.37, 0.39 and 0.4. The stress is fitted to the functional form given by Equation (5.2). The vertical line represents the τ_α, which is the α-relaxation time obtained from the work [5].

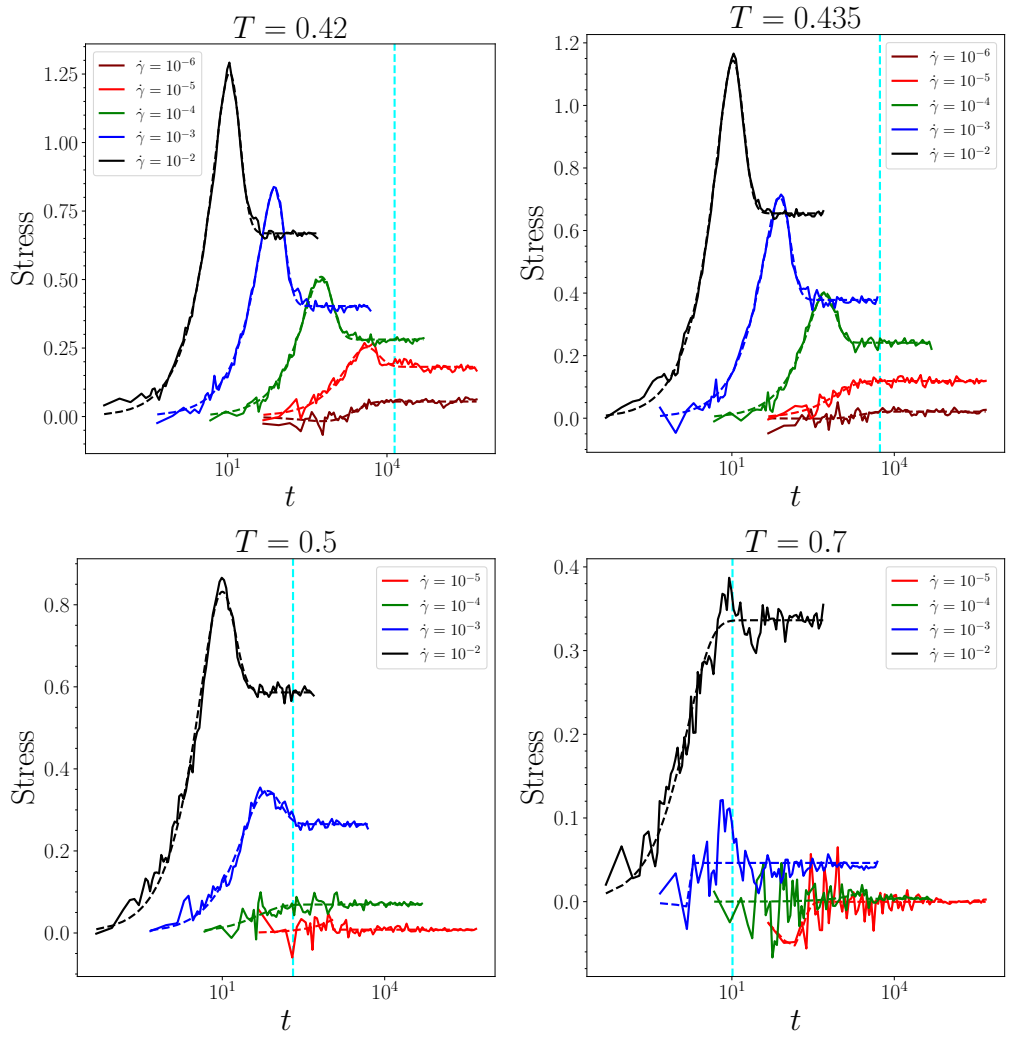


Figure 5.3: Stress vs time data for temperatures 0.42, 0.435, 0.5 and 0.7. The stress is fitted to the functional form given by Equation (5.2). The vertical line represents the τ_α, which is the α-relaxation time obtained from the work [5].

The τ is the relaxation time and it gives the estimate of how long it takes for the stress to reach steady state from the peak values. We then plot this fit parameter τ as a function of inverse $\dot{\gamma}_{xy}$ for different temperatures Figure 5.4. The fit function is of the form $\tau = C\dot{\gamma}_{xy}^{-b}$. The values of these fit parameters are given in Table 5.2

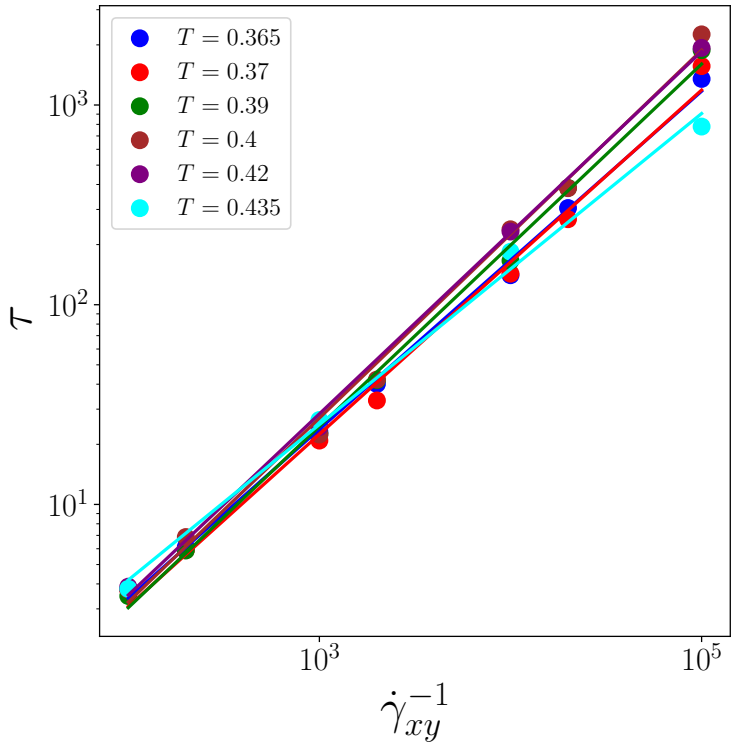


Figure 5.4: Relaxation time τ as a function of inverse strain rate $\dot{\gamma}_{xy}^{-1}$ for different temperatures. The data is fitted in the form $\tau = C\dot{\gamma}_{xy}^{-b}$.

It is observed that the relaxation time τ scales as power law with inverse strain rates. The exponents are given in Table 5.2.

T	Exponent b	Coefficient C	C/τ_α
0.365	0.846	0.0688	1.65×10^{-8}
0.37	0.862	0.0583	2.83×10^{-8}
0.39	0.907	0.0466	2.18×10^{-7}
0.40	0.921	0.0467	5.88×10^{-7}
0.42	0.907	0.0539	3.88×10^{-6}

Table 5.2: Fit parameters for the Figure 5.4

We also plotted the peak values of stress from the Figure 5.2 as a function of $\dot{\gamma}_{xy}$. The peak stress value increases with strain rates. It is also observed that the difference between the peak and the steady state values reduces at the lower strain rates.

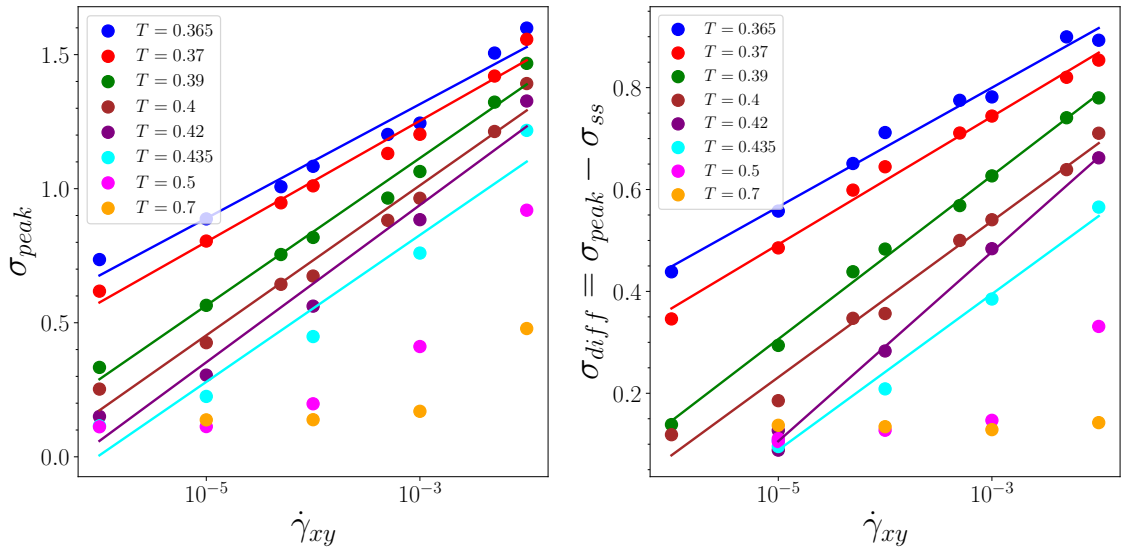


Figure 5.5: **(a)** Peak stress as a function of strain rate for the low T cases with fit function $\sigma_{\text{peak}} = A \ln \dot{\gamma} + B$. **(b)** The difference between the peak stress and the steady state stress with the fit function $\sigma_{\text{diff}} = C \ln \dot{\gamma} + D$. The data for both the cases is fitted for the temperatures upto 0.435.

T	A	B	C	D
0.365	0.0925	1.9543	0.0581	1.1509
0.37	0.0981	1.9307	0.0545	1.1197
0.39	0.1194	1.9390	0.0697	1.8066
0.40	0.1214	1.8513	0.0665	0.9975
0.42	0.1274	1.8188	0.0807	1.0353
0.435	0.1189	1.6488	0.0665	0.8542

Table 5.3: Fit parameters for the Figure 5.5

The time at which the peak stress occurs scales through power law with respect to the strain rate for the different temperatures.

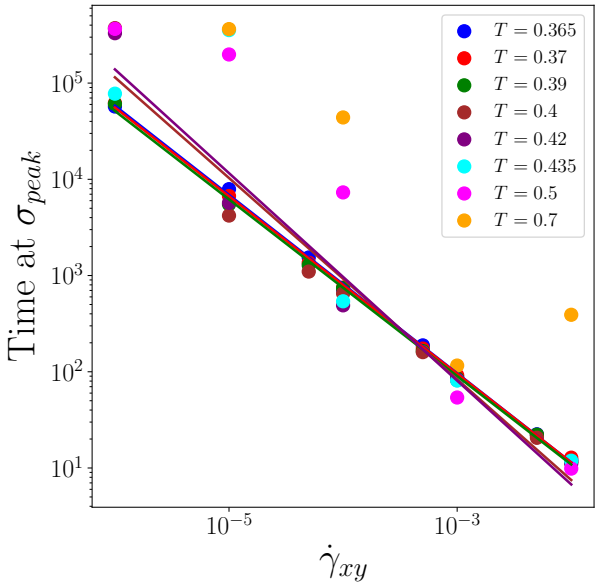


Figure 5.6: The time where the peak occurs in the stress strain curves. There is a power law scaling at lower temperatures of the form $t_{\text{peak}} = At^b$. Power law scaling is observed for temperatures upto $T = 0.42$

T	Exponent b	Coefficient A
0.365	-0.927	0.1598
0.37	-0.921	0.1638
0.39	-0.918	0.1583
0.40	-1.046	0.0606
0.42	-1.078	0.0471

Table 5.4: Fit parameters for the Figure 5.6

The stress is averaged over the steady state which occurs roughly at the timescales of order $1/\dot{\gamma}_{xy}$. 10 independent runs are performed to get the statistical average. The error bars in Figure 5.7a and Figure 5.8 correspond to the standard deviation over the independent runs.

Newtonian viscosity is obtained as $\eta = \sigma_{xy}/\dot{\gamma}_{xy}$ in Figure 5.8. For low strain rates we have the Newtonian regime where the viscosity is constant with strain rate. However, at higher strain rates we observe strain rate dependence in the viscosity. Shear thinning is observed, so that the viscosity decreases with increasing strain rates.

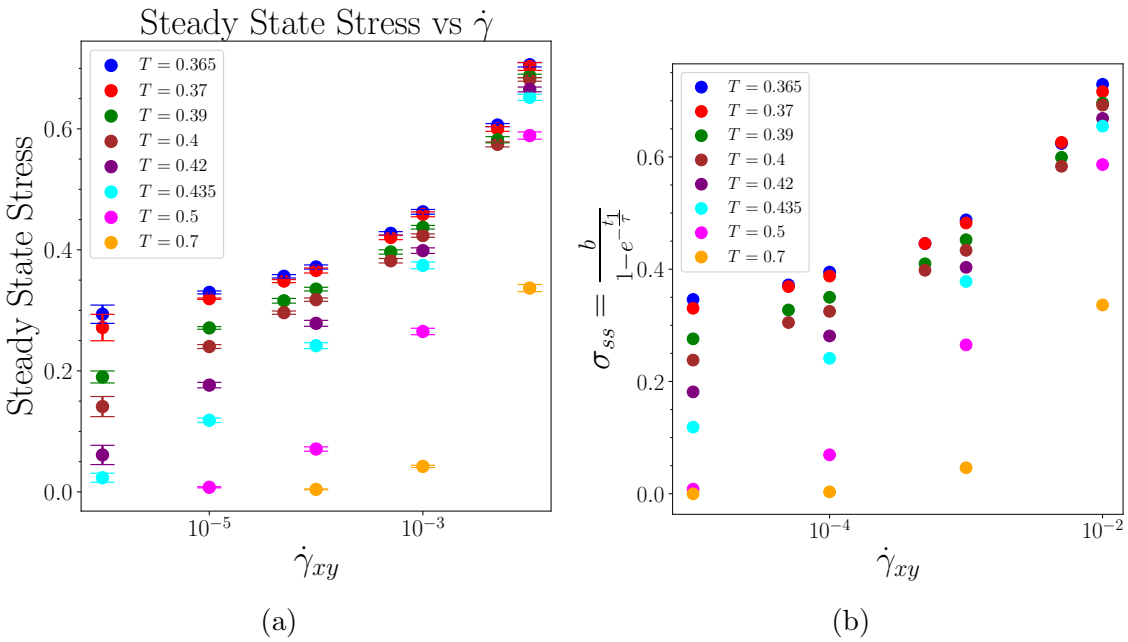


Figure 5.7: **(a)** Steady state stress obtained from 10 independent samples. **(b)** The fitted stress Equation (5.3) for the lower temperatures.

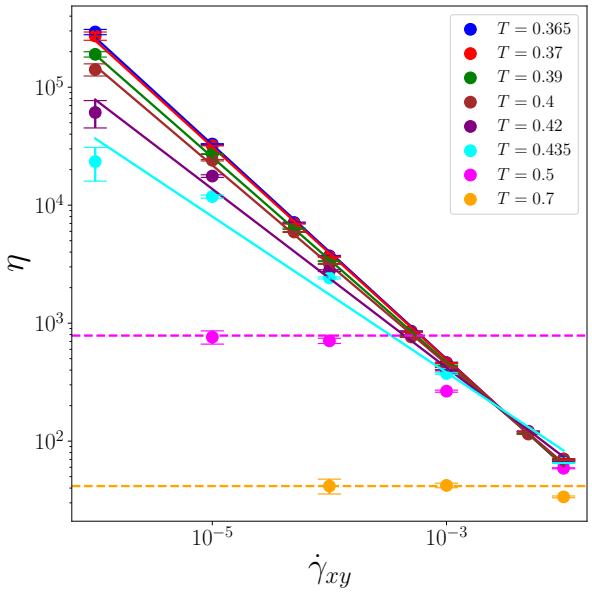


Figure 5.8: Viscosity as a function of strain rate obtained using SLLOD. The horizontal lines are the Green Kubo viscosity at the higher temperatures. Newtonian viscosity is observed for temperatures $T = 0.5$ and $T = 0.7$. Lower temperatures require much lower strain rates for the onset of Newtonian viscosity.

It is observed that the viscosity scales as power law at higher strain rates as observed in Figure 5.8. The fit function in the Figure 5.8 is of the form $A\eta^b$, where A is the coefficient and b is the power-law exponent. These fit parameters are given in Table 5.5. At higher temperatures, the viscosity reaches Newtonian regime at much higher strain rates. This is seen through the convergence of strain rate dependent viscosity to the Green Kubo values. However, on lowering the temperatures, the viscosity requires much lower strain rates to reach the Newtonian regime. This lowering of strain rates is in turn associated with larger time scales, which are typically beyond the MD time scales.

T	Exponent b	Coefficient A
0.365	-0.906	-0.058
0.37	-0.901	-0.024
0.39	-0.869	0.143
0.40	-0.842	0.297
0.42	-0.757	0.809
0.435	-0.661	1.378

Table 5.5: Fit parameters for the Figure 5.8

Chapter 6

Conclusions

In this thesis we have explored different methods of computing the viscosity using MD simulations for different temperatures. The routine equilibrium methods like the Green Kubo method involve calculation of auto-correlation functions and requires multiple trajectories to average over. Another way to calculate the viscosity from equilibrium simulations over multiple trajectories is the Helfand Moment method, which involves blocking and averaging the trajectory data. Helfand Moment has been found to be more efficient than the Green Kubo as it escapes the autocorrelation function calculation by instead using the ensemble average over different trajectories of the stress tensor.

While equilibrium methods are based on the Linear Response Theory, they lack the physical interpretation of the shear deformation to calculate viscosity as done in experiments. The simulation involving shear deformation is performed using Non-Equilibrium MD approach using the SLLOD algorithm at constant temperature. SLLOD method gives strain rate dependent stress and viscosity.

As the temperature is reduced, the dynamics of the system slow down and it is not possible to calculate viscosity at the time scales accessible through the Molecular Dynamics simulations. For dense gases and high temperature liquids, the stress decays at the molecular time scales. However, this is not true at lower temperatures. Hence the equilibrium methods are no longer effective at calculating the viscosity. Further research is needed in this field on developing methods to calculate viscosity efficiently at low temperatures.

For the shear simulations, we observe viscosity plateauing and becoming independent of

the strain rate at high temperatures at high enough strain rates that are tractable through the MD simulations. As we lower the temperatures, the viscosity requires much lower strain rates to plateau which are particularly not accessible through the MD simulations. Further research is required to extract viscosity more efficiently at the lower temperatures for the vanishing strain rates.

We have also proposed a new method to calculate viscosity using the differentiable simulations. The method is still new and needs further research and modifications to be able to work better. The results obtained here do not match qualitatively with the MD simulations. However, they are within the same order of magnitude. This method once modified and results match with the standard methods, would provide an alternative way to calculate zero shear viscosity on the fly without the need of autocorrelation calculation as required in Green Kubo formalism.

Bibliography

[1] B. J. Alder and T. E. Wainwright, *Studies in molecular dynamics. i. general method*, Journal of Chemical Physics **31** (1957), no. 2, 459–466.

[2] MP Allen, *Computer simulation of liquids*, 1987.

[3] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Dynamics of polymeric liquids, vol. 1: Fluid mechanics*, John Wiley & Sons, 1987.

[4] Martin Boisly, Markus Kästner, Jörg Brummund, and Volker Ulbricht, *General aspects of yield stress fluids – terminology and definition of viscosity*, Applied Rheology **24** (2014), no. 1, 26–36.

[5] Pallabi Das and Srikanth Sastry, *Crossover in dynamics in the kob-andersen binary mixture glass-forming liquid*, Journal of Non-Crystalline Solids: X **14** (2022), 100098.

[6] D. Frenkel and B. Smit, *Understanding molecular simulation: From algorithms to applications*, Academic Press, 2002.

[7] Melville S Green, *Markoff random processes and the statistical mechanics of time-dependent phenomena. ii. irreversible processes in fluids*, The Journal of chemical physics **22** (1954), no. 3, 398–413.

[8] Remco Hartkamp, B. D. Todd, and Stefan Luding, *A constitutive framework for the non-Newtonian pressure tensor of a simple fluid under planar flows*, The Journal of Chemical Physics **138** (2013), no. 24, 244508.

[9] Eugene Helfand, *Transport coefficients from dissipation in a canonical ensemble*, Physical Review **119** (1960), no. 1, 1.

[10] William G Hoover, *Canonical dynamics: Equilibrium phase-space distributions*, Physical review A **31** (1985), no. 3, 1695.

[11] DEBRA J. SEARLES KARL P. TRAVIS and DENIS J. EVANS, *Strain rate dependent properties of a simple fluid*, Molecular Physics **95** (1998), no. 2, 195–202.

[12] Ryogo Kubo, *Statistical-mechanical theory of irreversible processes. i. general theory and simple applications to magnetic and conduction problems*, Journal of the physical society of Japan **12** (1957), no. 6, 570–586.

[13] A. R. Leach, *Molecular modelling: Principles and applications*, Prentice Hall, 2001.

[14] A W Lees and S F Edwards, *The computer study of transport processes under extreme conditions*, Journal of Physics C: Solid State Physics **5** (1972), no. 15, 1921.

[15] Glenn J Martyna, Douglas J Tobias, and Michael L Klein, *Constant pressure molecular dynamics algorithms*, The Journal of chemical physics **101** (1994), no. 5, 4177–4189.

[16] Glenn J Martyna, Mark E Tuckerman, Douglas J Tobias, and Michael L Klein, *Explicit reversible integrators for extended systems dynamics*, Molecular Physics **87** (1996), no. 5, 1117–1157.

[17] Shūichi Nosé, *A molecular dynamics method for simulations in the canonical ensemble*, Molecular physics **52** (1984), no. 2, 255–268.

[18] Samuel S. Schoenholz and Ekin D. Cubuk, *Jax m.d. a framework for differentiable physics*, Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020.

[19] Shiladitya. Sengupta, *Investigations of the role of spatial dimensionality and interparticle interactions in model glass-formers /*, JNCASR,, Bangalore :, 2013., Thesis.

[20] A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in 't Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott, and S. J. Plimpton, *LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales*, Comp. Phys. Comm. **271** (2022), 108171.

[21] Billy D Todd and Peter J Daivis, *Nonequilibrium molecular dynamics: theory, algorithms and applications*, Cambridge University Press, 2017.

[22] Mark E Tuckerman, José Alejandre, Roberto López-Rendón, Andrea L Jochim, and Glenn J Martyna, *A liouville-operator derived measure-preserving integrator for molecular dynamics simulations in the isothermal–isobaric ensemble*, Journal of Physics A: Mathematical and General **39** (2006), no. 19, 5629.

[23] Arun Verma, *An introduction to automatic differentiation*, Current Science **78** (2000), no. 7, 804–807.

[24] Wujie Wang, Zhenghao Wu, Johannes C. B. Dietschreit, and Rafael Gómez-Bombarelli, *Learning pair potentials using differentiable simulations*, The Journal of Chemical Physics **158** (2023), no. 4, 044113.

[25] Yong Zhang, Akihito Otani, and Edward J Maginn, *Reliable viscosity calculation from equilibrium molecular dynamics simulations: A time decomposition method*, Journal of chemical theory and computation **11** (2015), no. 8, 3537–3546.