

3

SIMULATION METHODOLOGY

In this chapter the basic simulation methodologies used in this work are described. Specific descriptions for the equations of motion, force calculation, periodic boundary conditions, integration algorithm, Gaussian constraints and SLLOD algorithm are given.

3.1 Molecular Dynamics

If we have two masses interacting through gravitational forces we can exactly predict what is going to happen at any point in the future or calculate backwards to any point in history. But adding one extra mass makes it impossible to analytically solve the time evolution of the system. Although we know exactly how the masses interact with each other and what the equations of motion are there is, in general, no way of telling whether they will collide and if so at which instant in time analytically. Only by numerically solving the equations of motion for short successive intervals of time can this problem be tackled. Molecular dynamics (MD) simulation is a powerful tool to be used to solve the classical equations of motion for many body systems and has been successfully used to simulate systems with many millions of atoms interacting with each other.

3.1.1 Equations of motion

The microscopic coordinates and momenta of all the molecules should be determined, as they are used to obtain thermodynamic and transport properties via statistical mechanics. If the simulations are carried out for a sufficient length of time, the resulting

time averages are equivalent to ensemble averages due to ergodicity [47]. It is assumed that the motion of atoms is governed by Newton's classical equations of motion given by

$$\frac{\mathbf{p}_i}{m_i} = \frac{d\mathbf{r}_i}{dt} \quad (3.1)$$

$$\mathbf{F}_i = \frac{d\mathbf{p}_i}{dt} \quad (3.2)$$

where \mathbf{F}_i and \mathbf{p}_i are the force and momentum of atom i respectively. The force is given by

$$\mathbf{F}_i = -\frac{\partial\phi}{\partial\mathbf{r}_i}. \quad (3.3)$$

The potential energy ϕ in the above equation is given by the sum of inter and intramolecular interactions. The equations are integrated using finite difference schemes, which typically involve a calculation of forces between molecules at each time step. Examples of such integration schemes are the velocity Verlet, the Gear-predictor, and the leap-frog algorithms [48, 49].

3.1.2 Force calculation

The force field or the potential energy used to describe interactions between atoms in the system is a function of the set of coordinates of each atom. The accuracy of results directly depends on how well the force field describes the various interactions between the atoms. It is therefore very important to select the appropriate model for each system. In this work, the atomic fluids are simulated according to a model with truncated and shifted Lennard-Jones (LJ) interactions between all atoms. The LJ potential describing all interactions in the system is truncated and shifted so that the potential has no discontinuity and is zero beyond a cut-off distance $r_c = 2^{1/6}\sigma$ (where σ is the distance at which the unshifted potential is zero). A LJ potential with this truncation point is often known as the WCA potential [50], and it results in purely repulsive interactions. This

potential is convenient for computational work because it is short-ranged, and therefore computationally undemanding, but still retains the essential physics, i.e. the repulsive (excluded volume) interaction. The truncated and shifted LJ potential is defined as

$$\phi(r) = \begin{cases} 4\varepsilon [(\sigma/r)^{12} - (\sigma/r)^6] + \varepsilon & r \leq 2^{1/6}\sigma \\ 0 & r \geq 2^{1/6}\sigma \end{cases} \quad (3.4)$$

where ε is the LJ energy parameter and σ the LJ length parameter. When using the LJ potentials in simulation, it is customary to work in a system of units where $\sigma = 1$ and $\varepsilon = 1$. Comparison of WCA and LJ potentials is shown in figure 3.1.

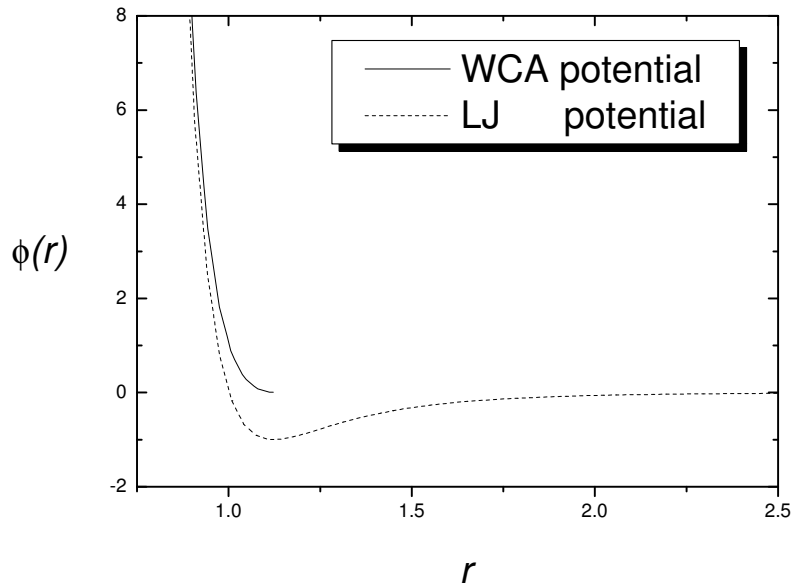


Figure 3.1: Comparison of WCA and LJ potentials. Reduced units used ($\sigma = \varepsilon = 1$).

In order to validate our pressure tensor and heat flux vector derivation in section 2.3, we also perform some simulations on fluid and solid atoms that interact via the Barker-Fisher-Watts two-body potential [51] and Axilrod-Teller three-body potential [52]. The

total intermolecular potential ϕ is a contribution from two-body interactions $\phi^{(2)}$ and three-body dispersion interactions $\phi^{(3)}$

$$\phi(r) = \phi^{(2)}(r) + \phi^{(3)}(r) . \quad (3.5)$$

The two-body interaction of argon is well represented by the Barker-Fisher-Watts (BFW) potential [51]. The BFW potential is a linear combination of the Barker-Pompe [53] (ϕ_{BP}) and Bobetic-Barker [54] (ϕ_{BB}) potentials

$$\phi^{(2)}(r) = 0.75\phi_{BB}(r) + 0.25\phi_{BP}(r) , \quad (3.6)$$

where the potentials of Barker-Pompe and Bobetic-Barker have the following form:

$$\phi^{(2)}(r) = \varepsilon \left[\sum_{i=0}^5 A_i (x-1)^i \exp[\alpha(1-x)] - \sum_{j=0}^2 \frac{C_{2j+6}}{\delta + x^{2j+6}} \right] . \quad (3.7)$$

Here, $x = r/r_m$ where r_m is the intermolecular separation at which the potential has a minimum value and the other parameters are listed in Table 3.1.

The triple-dipole Axilrod-Teller (AT) potential [52] is

$$\phi^{(3)}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) = \frac{v_{DDD}(1 + 3\cos\theta_i \cos\theta_j \cos\theta_k)}{(r_{ij}r_{ik}r_{jk})^3} , \quad (3.8)$$

where v_{DDD} is the nonadditive coefficient, and the angles and intermolecular separations refer to a triangular configuration of atoms. The nonadditive coefficient for argon is 518.3 a.u. [55]. Recent work [56] has demonstrated that the Axilrod-Teller term can significantly improve the prediction of liquid phase properties.

3.1.3 Periodic boundary conditions

The problem of surface effects in small systems can be overcome through the use of periodic boundary conditions (PBC) [47] whereby the simulation box is replicated to form an infinite lattice by rigid translation in all the three principal directions, completely

Argon		
$\varepsilon/\kappa(K)$	142.095	
$\sigma(\text{\AA})$	3.3605	
$r_m(\text{\AA})$	3.7612	
	Barker-Pompe	Bobetic-Barker
$\varepsilon/\kappa(K)$	147.70	140.235
$\sigma(\text{\AA})$	3.341	3.3666
$r_m(\text{\AA})$	3.7560	3.7630
A_0	0.2349	0.29214
A_1	-4.7735	-4.41458
A_2	-10.2194	-7.70182
A_3	-5.2905	-31.9293
A_4	0.0	-136.026
A_5	0.0	-151.00
C_6	1.0698	1.11976
C_8	0.1642	0.171551
C_{10}	0.0132	0.013748
α	12.5	12.5
δ	0.01	0.01

Table 3.1: Parameters for the Barker-Fisher-Watts potential.

filling the space. In other words, if one of the particles is located at position \mathbf{r} in the box, we assume that this particle really represents an infinite set of particles located at

$$\mathbf{r} + l\mathbf{a} + m\mathbf{b} + n\mathbf{c} \quad (l, m, n = -\infty, +\infty)$$

where l, m, n are integer numbers, and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ are the vectors corresponding to the edges of the box. The key point is that now each particle i in the box should be thought as interacting not only with other particles j in the box, but also with their images in nearby boxes. Consequently, the position of the box boundaries has no effect (that is, there are no artificial walls in the systems) and surface effects are eliminated. Apparently, the number of interacting pairs increases enormously as an effect of the PBC, which turns out to be very computationally intensive when the number of molecules, N , becomes large. In practice, this is not true, because most common intermolecular potential energy functions such as the LJ potential decay rapidly at large separation distances. For example, the LJ potential asymptotes rapidly to zero for distances greater than 2.5σ . Customarily, a potential cut-off radius is used whereby the potential is computed only for separations smaller than a distance r_c . Interactions are disregarded for separations greater than r_c . The cut-off distance relates to the periodic boundary conditions, in that it can be any distance up to half the length of the simulation box. The effects of truncating a full-ranged potential can be approximately estimated by treating the system as a uniform (constant density) continuum beyond r_c . For a bulk periodic system, this usually amounts to a constant additive correction to quantities such as potential energy and pressure.

3.1.4 Time integration algorithm

The time evolution of a set of interacting particles is followed by integrating their equations of motion. Any changes in particle position will also affect the strength of inter-particle forces, velocities, and accelerations. Consequently, a time integration algorithm

is required to integrate the equations of motion of the interacting particles and follow their trajectory. Time integration algorithms are based on finite difference methods, where time is discretized on a finite grid, the time step Δt being the distance between consecutive points on the grid. Knowing the positions and some of their time derivatives at time t (the exact details depend on the type of algorithm), the integration scheme gives these quantities at a later time $t + \Delta t$. By iterating the procedure, the time evolution of the system can be followed for long times. Of course, these schemes are approximate and there are errors associated with them. These errors are due to truncation, which is related to the accuracy of the finite difference method with respect to the true solution. Finite difference methods are usually based on a Taylor expansion truncated at some term. These errors do not depend on the implementation: they are intrinsic to the algorithm. Another source of errors is roundoff error, which is associated with a particular implementation of the algorithm, for instance, the finite number of digits used in the computer simulation. Both errors can be reduced by decreasing the time step Δt .

We use the Gear predictor-corrector algorithm [35, 47, 57] to integrate the equations of motion for its efficiency and accuracy. It consists of three steps:

- *Predictor.* From the positions and their time derivatives up to a certain order n (we choose $n = 4$), all known at time t , the same quantities at time $t + \Delta t$ are predicted by means of a Taylor expansion. We use $\mathbf{r}_i^{(n)}$ to represent the n^{th} order derivative term in the Taylor expansion of the position vector of particle i , \mathbf{r}_i , with respect to time t ,

$$\mathbf{r}_i^{(n)} = \frac{1}{n!} (\Delta t)^n \left(\frac{d^n \mathbf{r}_i}{dt^n} \right). \quad (3.9)$$

Similarly,

$$\mathbf{p}_i^{(n)} = \frac{1}{n!} (\Delta t)^n \left(\frac{d^n \mathbf{p}_i}{dt^n} \right). \quad (3.10)$$

The predicted position vector, \mathbf{r}_i^p , momentum, \mathbf{p}_i^p , and their scaled time derivatives, $\mathbf{r}_i^{(n)p}$, and $\mathbf{p}_i^{(n)p}$ at time $t + \Delta t$ are as follows,

$$\begin{pmatrix} \mathbf{r}_i^p(t + \Delta t) \\ \mathbf{r}_i^{(1)p}(t + \Delta t) \\ \mathbf{r}_i^{(2)p}(t + \Delta t) \\ \mathbf{r}_i^{(3)p}(t + \Delta t) \\ \mathbf{r}_i^{(4)p}(t + \Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{r}_i(t) \\ \mathbf{r}_i^{(1)}(t) \\ \mathbf{r}_i^{(2)}(t) \\ \mathbf{r}_i^{(3)}(t) \\ \mathbf{r}_i^{(4)}(t) \end{pmatrix} \quad (3.11)$$

$$\begin{pmatrix} \mathbf{p}_i^p(t + \Delta t) \\ \mathbf{p}_i^{(1)p}(t + \Delta t) \\ \mathbf{p}_i^{(2)p}(t + \Delta t) \\ \mathbf{p}_i^{(3)p}(t + \Delta t) \\ \mathbf{p}_i^{(4)p}(t + \Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 3 & 6 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{p}_i(t) \\ \mathbf{p}_i^{(1)}(t) \\ \mathbf{p}_i^{(2)}(t) \\ \mathbf{p}_i^{(3)}(t) \\ \mathbf{p}_i^{(4)}(t) \end{pmatrix} \quad (3.12)$$

- *Force evaluation.* The force is computed taking the gradient of the potential at the predicted positions. The resulting acceleration $\mathbf{a}_i^c(t + \Delta t)$ will be in general different from the predicted acceleration $\mathbf{a}_i^p(t + \Delta t)$. The difference between the two

$$\Delta \mathbf{a}_i(t + \Delta t) = \mathbf{a}_i^c(t + \Delta t) - \mathbf{a}_i^p(t + \Delta t) \quad (3.13)$$

can be an error signal.

- *Corrector*. This error signal is used to correct positions and their derivatives. All the corrections are proportional to the error signal. The corrector step is simply:

$$\begin{pmatrix} \mathbf{r}_i^c(t + \Delta t) \\ \mathbf{r}_i^{(1)c}(t + \Delta t) \\ \mathbf{r}_i^{(2)c}(t + \Delta t) \\ \mathbf{r}_i^{(3)c}(t + \Delta t) \\ \mathbf{r}_i^{(4)c}(t + \Delta t) \end{pmatrix} = \begin{pmatrix} \mathbf{r}_i^p(t + \Delta t) \\ \mathbf{r}_i^{(1)p}(t + \Delta t) \\ \mathbf{r}_i^{(2)p}(t + \Delta t) \\ \mathbf{r}_i^{(3)p}(t + \Delta t) \\ \mathbf{r}_i^{(4)p}(t + \Delta t) \end{pmatrix} + \begin{pmatrix} k_0 \\ k_1 \\ k_2 \\ k_3 \\ k_4 \end{pmatrix} \Delta \mathbf{a}_i(t + \Delta t) \quad (3.14)$$

Here k_0, k_1, k_2, k_3, k_4 , are the corrector coefficients which depend upon the order of the differential equation being solved [57]. In our case, $k_0 = 251/720$, $k_1 = 1$, $k_2 = 11/12$, $k_3 = 1/3$, $k_4 = 1/24$.

Other methods can be used to integrate equations of motion of the particles. The Commonly used Verlet algorithm has a truncation error proportional to Δt^4 for each integration time step. For more detailed information on time integration algorithm, the reader is referred to [47, 49]. A comparison between the Verlet and Gear schemes can be found in [58].

3.1.5 Gaussian constraints

In this work, we simulate Poiseuille flow. The simulation geometry is shown in Fig. 5.1. In order to keep the wall temperature constant in our simulation, we apply Gaussian constraints by introducing a perturbative force into the equations of motion, and minimising the least squares difference between the Newtonian and constrained trajectories. Generally speaking, there are two types of constraints: holonomic and nonholonomic. A

holonomic constraint is one which can be expressed as equations connecting the coordinates of the particles (and possibly the time) having the form

$$f(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots, t) = 0 . \quad (3.15)$$

Holonomic constraints may be integrated out of the equations of motion. Perhaps the simplest example of holonomic constraints is the rigid body, where the constraints are expressed by equations of the form

$$(\mathbf{r}_i - \mathbf{r}_j)^2 = l^2 , \quad (3.16)$$

where l is the distance between particle i and particle j . If we have M such equations then the original $3N$ coordinates are no longer free to vary independently. Connecting the M equations together establishes equations of constraint between the coordinates. For such a system having $3N$ coordinates and M constraint equations, there are $(3N - M)$ degrees of freedom.

Constraints which cannot be expressed as either algebraic or integrable differential equations are known as nonholonomic constraints. Nonholonomic constraints usually involve velocities, are not integrable, and usually do work on a system. Thermodynamic constraints are generally nonholonomic. In particular, there are two types of nonholonomic constraints in practical problems. The first type is the inequality constraint of the form

$$f(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots) \geq 0 . \quad (3.17)$$

The second type of nonholonomic constraints is the nonintegrable differential equation,

$$f(\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3, \dots ; \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \dots ; t) = 0 . \quad (3.18)$$

Consider the function f , a function of the set of particle accelerations

$$f(\ddot{\mathbf{r}}_1, \ddot{\mathbf{r}}_2, \ddot{\mathbf{r}}_3, \dots \ddot{\mathbf{r}}_N) = \frac{1}{2} \sum_{i=1}^N \left[\ddot{\mathbf{r}}_i - \frac{\mathbf{F}_i}{m_i} \right]^2 . \quad (3.19)$$

If we impose constant temperature constraints,

$$G(\dot{\mathbf{r}}_1, \dot{\mathbf{r}}_2, \dot{\mathbf{r}}_3, \dots, \dot{\mathbf{r}}_N; t) = \sum_{i=1}^N \frac{m_i \dot{\mathbf{r}}_i^2}{2} - \frac{3NkT}{2} = 0, \quad (3.20)$$

then we have the differential form of the constraint equation,

$$g(\ddot{\mathbf{r}}_1, \ddot{\mathbf{r}}_2, \ddot{\mathbf{r}}_3, \dots, \ddot{\mathbf{r}}_N; \dot{\mathbf{r}}_1, \dot{\mathbf{r}}_2, \dot{\mathbf{r}}_3, \dots, \dot{\mathbf{r}}_N; t) = \frac{dG}{dt} = \sum_i^N m_i \dot{\mathbf{r}}_i \cdot \ddot{\mathbf{r}}_i = 0. \quad (3.21)$$

Gauss's principle states that the actual physical acceleration corresponds to the minimum value of f . If the system is not subject to a constraint then $f = 0$ and (3.19) reduces to the normal Newtonian equation of motion, otherwise the constrained equations of motion are obtained by finding the minimum of f in (3.19), subject to the constraint, (3.21),

$$\frac{\partial}{\partial \ddot{\mathbf{r}}_i} (f - \zeta g) = 0 \quad (3.22)$$

$$\frac{\partial}{\partial \ddot{\mathbf{r}}_i} \left[\frac{1}{2} \sum_{i=1}^N \left(\ddot{\mathbf{r}}_i - \frac{\mathbf{F}_i}{m_i} \right)^2 - \zeta \sum_i^N m_i \dot{\mathbf{r}}_i \cdot \ddot{\mathbf{r}}_i \right] = 0 \quad (3.23)$$

Differentiating this equation and performing the necessary algebraic manipulations, we obtain

$$m_i \ddot{\mathbf{r}}_i = \mathbf{F}_i - \zeta m_i \dot{\mathbf{r}}_i \quad i = 1, 2, 3, \dots, N \quad (3.24)$$

where ζ is the Gaussian multiplier. Eq. (3.24) is a 2^{nd} order differential equation, equivalent to the two first order differential equations:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} \quad (3.25)$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \zeta m_i \dot{\mathbf{r}}_i \quad (3.26)$$

Substituting Eqs. (3.25) and (3.26) into the differential form of the constraint equation (3.21) yields the result

$$\zeta = \frac{\sum_{i=1}^N \dot{\mathbf{r}}_i \cdot \mathbf{F}_i}{\sum_{i=1}^N m_i \dot{\mathbf{r}}_i^2} \quad (3.27)$$

3.2 SLLOD algorithm

Gosling *etc.* [59] pioneered the use of nonequilibrium molecular dynamics (NEMD) by imposing a spatially periodic external force on the particles to generate an oscillatory velocity profile while maintaining conventional periodic boundary conditions. The amplitude of this velocity profile at steady state is inversely related to the shear viscosity, and hence the viscosity can be calculated. The shear viscosity determined in this manner is wavelength dependent. However, the Newtonian shear viscosity is obtained only in the long wavelength limit i.e. in the limit $k \rightarrow 0$, where k is the wave vector of the oscillatory perturbation. This means that a very large simulation box is required to get reasonably accurate values of shear viscosity, which limits the usefulness of this technique. The more successful NEMD techniques are the zero wave vector techniques which involve imposing a planar Couette flow velocity profile shown in Fig. 3.2. The most efficient direct NEMD algorithm for shear viscosity is the SLLOD algorithm [1]. For an atomic fluid the SLLOD equations of motion [1] are

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m_i} + \mathbf{r}_i \cdot \nabla \mathbf{u} \quad (3.28)$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \mathbf{p}_i \cdot \nabla \mathbf{u} - \zeta \mathbf{p}_i \quad (3.29)$$

where ζ is the thermostating multiplier, $\mathbf{u} = (u_x, 0, 0)$, and $u_x = \dot{\gamma} y$, where $\dot{\gamma}$ is the strain rate; y is the direction normal to the direction of flow in the x direction. The SLLOD algorithm is combined with the Lees-Edwards “sliding brick” periodic boundary

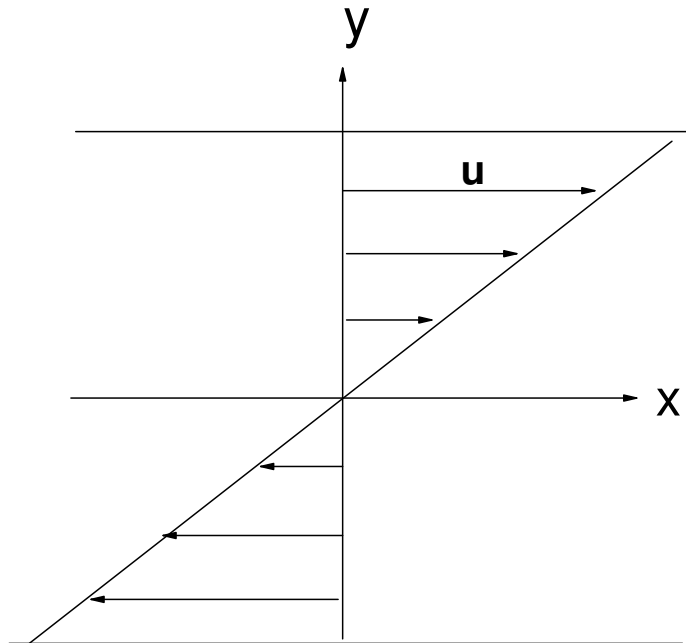


Figure 3.2: Planar Couette flow velocity profile. $\mathbf{u} = \mathbf{i}\dot{\gamma}y$.

conditions [60], where the periodic images above and below the central simulation box (in the y direction) are moved in opposite directions. The strain rate dependent shear viscosity is obtained from the constitutive equation

$$\eta(\dot{\gamma}) = -\frac{\langle P_{xy} \rangle}{\dot{\gamma}} \quad (3.30)$$

The Newtonian shear viscosity is estimated by extrapolating the shear viscosities to zero shear rate.