4. Molecular dynamics simulation

4.1. Brief overview of different computer simulation methods

Computer simulation is a powerful and modern tool for solving scientific problems as numerical experiments can be performed for new materials without synthesizing them. One of the aims of computer simulation is to reproduce experiment to elucidate the invisible microscopic details and further explain experiments. On the other hand, simulation can also be used as a useful predictive tool. The most widely used simulation methods for molecular systems are Monte Carlo, Brownian dynamics and molecular dynamics.

The Monte Carlo method is a stochastic strategy that relies on probabilities (Sadus, 1999). The Monte Carlo sampling technique generates large numbers of configurations or microstates of equilibrated systems by stepping from one microstate to the next in a particular statistical ensemble. Random changes are made to the positions of the species present, together with their orientations and conformations where appropriate. Quantities of interest can be averaged over these microstates. The advantages of the Monte Carlo simulation technique include the ease of extending it to simulate different ensembles, flexibility in the choice of sampling functions and the underlying matrix or trial move which must satisfy the principle of microscopic reversibility as well as time-saving as only the potential energy is required (Allen and Tildesley, 1993).

Brownian dynamics is an efficient approach for simulations of large polymer molecules or colloidal particles in a small molecule solvent. In this approach, the solvent is treated as a viscous continuum which dissipates energy as macromolecules or particles move through it. The Brownian motion of the macromolecules produced by random collisions with solvent molecules is mimicked by a stochastic force generated by pseudo-random numbers.

Molecular dynamics is the most detailed molecular simulation method (Allen and Tildesley, 1987) which computes the motions of individual molecules. Coupled Newton's equations of motion, which describe the positions and momenta, are solved for a large number of particles in an isolated cluster or in the bulk using periodic

boundary conditions. The equations of motion for these particles which interact with each other via intra- and inter-molecular potentials can be solved accurately using various numerical integration methods such as the common predictor-corrector or Verlet methods. Molecular dynamics efficiently evaluates different configurational properties and dynamic quantities which cannot generally be obtained by Monte Carlo (Haile, 1997).

4.2. Coarse-grained model

Early models of liquids (Morrell and Hildebrand, 1936) involved the physical manipulation and analysis of the packing of a large number of gelatine balls, representing the molecules. This resulted in a surprisingly good three-dimensional picture of the structure of a liquid. Later, there was some interest in the study of assemblies of metal ball bearings, kept in motion by mechanical vibration (Pieranski et al., 1978). However the use of large numbers of physical objects to represent molecules can be very time-consuming, there are obvious limitations on the types of interactions between them, and the effects of gravity can never be eliminated. The natural extension of this approach is to use a mathematical, rather than a physical, model, and to perform the analysis by computer (Allen and Tildesley, 1987).

The first computer simulation of liquids was carried out in 1953 (Metropolis et al., 1953). The model was an idealized two-dimensional representation of molecules as rigid disks. Only four years later, the simulation was carried out on the three-dimensional Lennard-Jones fluids (Wood and Parker, 1957) which made it possible to compare data obtained from experiments with computer-generated data. In the late 1950s, the molecular dynamics method was first introduced to study the interactions of hard spheres (Alder and Wainwright, 1957, Alder and Wainwright, 1959).

Nowadays simulation of materials can be performed at different scales, from a coarse-grained approach to a fully atomistic model. Although the fully atomistic method is more accurate, it is computationally expensive due to a large number of degrees of freedom and provides a large amount of data that is sometimes irrelevant to the properties studied. For macromolecular systems, the coarse-grained approach is widely

used as the modeling process is simplified, hence becomes more efficient, and the characteristic topological features of the molecule can still be maintained.

The level of detail for a coarse-grained model varies for different cases. The whole molecule can be represented by a single particle in a simulation and interactions between particles incorporate average properties of the whole molecule. With this approach, the number of degrees of freedom is greatly reduced. Different polymers have been modeled using this method (Bolhuis et al., 2001, Harreis et al., 2003, Likos et al., 2002, Likos et al., 2001, Louis and Bolhuis, 2000). On the other hand, a segment of a polymer molecule can also be represented by a particle (bead). Using this method, branched polymers of different architectures have been successfully simulated (Jabbarzadeh et al., 2003).

The first coarse-grained model, called the 'dumbbell' model, was introduced in the 1930s (Bird et al., 1987). This is only a very crude representation of polymers, as can be seen in Figure 4.1. Molecules are treated as a pair of beads interacting via a harmonic potential. However by using this model, it is possible to perform kinetic theory derivations and calculations for nonlinear rheological properties and solve some flow problems. The analytical results for the dumbbell models can also be used to check computer simulation procedures in molecular dynamics and Brownian dynamics (Bird et al., 1987).

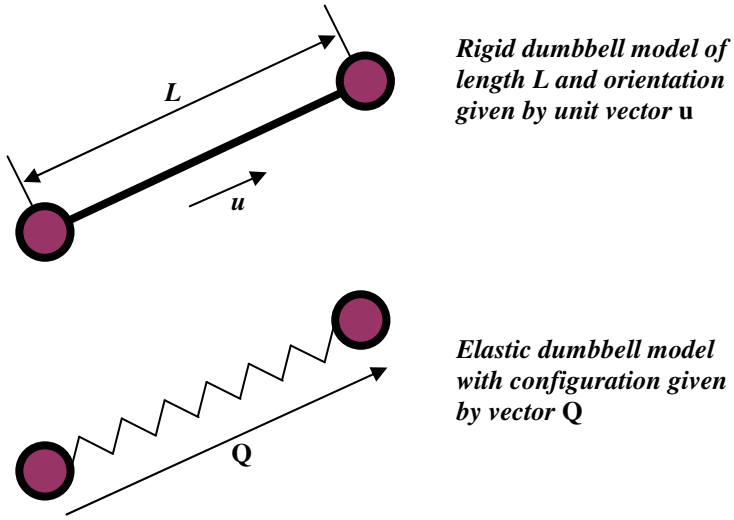

Figure 4.1. The first coarse-grained models – the rigid and elastic dumbbell models.

Some years later, the bead-rod and bead-spring model were introduced to model chainlike macromolecules. Beads in the bead-rod model do not represent the atoms of the polymer chain backbone, but some portion of the chain, normally 10 to 20 monomer units. These beads are connected by rigid and massless rods. While in the bead-spring model, a portion of the chain containing several hundreds of backbone atoms are replaced by a "spring" and the masses of the atoms are concentrated in the mass of beads. Figure 4.2 illustrates the configurations of linear polymers represented by these models.

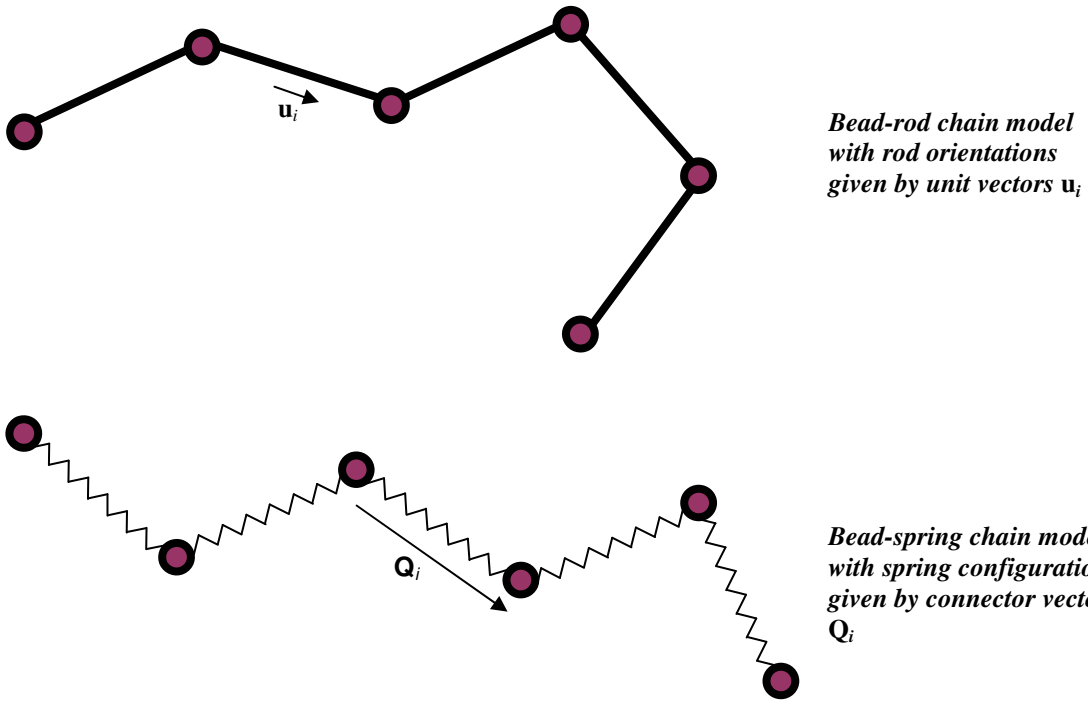

Figure 4.2. The freely jointed bead-rod and bead-spring chain models.

If the springs are taken to be Hookean springs, the bead-spring chain is referred to as a Rouse chain or a Rouse-Zimm chain. This approach has been applied widely as it has a large number of internal degrees of freedom and exhibits orientability and stretchability. However the disadvantage of this model is that it does not have a constant contour length and can be stretched out to any length. Therefore in many cases finitely extensible springs with two more parameters, the spring constant and the maximum extensibility of an individual spring, can be included so the contour length of the chain model cannot exceed a certain limit.

In this work, hyperbranched polymers have been simulated using coarse-grained uniform beads connecting via finitely extensible springs. These basic units correspond to the linear units or branching points of the molecule and are interconnected to create tree-like structures. Beads along the chain can rotate and vibrate freely. All beads are identical and indistinguishable except for their positions in the molecule.

4.3. Intermolecular interaction

The most widely used model to calculate the interactions is the Lennard – Jones (LJ) potential. This model is useful for soft-sphere pair potentials and was introduced in 1924 (Lennard-Jones, 1924a, Lennard-Jones, 1924b):

$$U_{ij}^{LJ} = k\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{n} - \left(\frac{\sigma}{r_{ij}}\right)^{m}\right] \tag{4.1}$$

where

$$k = \frac{n}{n-m}\left(\frac{n}{m}\right)^{m/(n-m)} \tag{4.2}$$

This model attemps to account for both short range, repulsive overlap forces, and longer range, attractive dispersion forces. Short range repulsive forces prevent the substance from collapsing onto itself whereas longer range attractions deter disintegration of the substance (Haile, 1997). These forces have the range and strength determined by the values of *n* and *m*. The value of *m* is normally chosen to be 6 because the leading term in London's theory (London, 1930) for dispersion varies as $1/r^6$ and the value of *n* is set as $n = 2m = 12$. The resulting LJ model is then given as:

$$U_{ij}^{LJ} = 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \tag{4.3}$$

In order to shorten the computer time, the pair interactions beyond the distance of r_c are neglected. Therefore the LJ potential actually used in simulations is a truncated version defined as:

$$\begin{aligned} U_{ij}^{LJ} &= 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \qquad \text{for} \quad r_{ij} \le r_c \\ U_{ij}^{LJ} &= 0 \qquad\qquad\qquad\qquad\qquad\quad \text{for} \quad r_{ij} > r_c \end{aligned} \tag{4.4}$$

r_c is normally chosen to be 2.5σ, at which $U = -0.0163\varepsilon$, hence when $r_{ij} = r_c$, atom j only makes a small contribution to the force on atom i (Haile, 1997).

In this work, the pairwise Weeks-Chandler-Anderson (WCA) (Weeks et al., 1971) potential is used. This is actually a truncated and shifted LJ potential given by:

$$\begin{aligned} U_{ij}^{WCA} &= 4\varepsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] + \varepsilon \quad \text{for} \quad r_{ij}/\sigma < 2^{1/6} \\ U_{ij}^{WCA} &= 0 \qquad\qquad\qquad\qquad\qquad\quad \text{for} \quad r_{ij}/\sigma \geq 2^{1/6} \end{aligned} \tag{4.5}$$

where r_{ij} is the separation between the sites represented by beads i and j, ε is the potential well depth, and σ is the effective diameter of the beads. This potential results in a purely repulsive force that includes the effect of excluded volume.

For beads with chemical bonds, a finitely extensible nonlinear elastic (FENE) (Grest and Kremer, 1986) potential is added. The FENE potential is given as:

$$\begin{aligned} U_{ij}^{FENE} &= -0.5kR_0^2 \ln\left[1-\left(r_{ij}/R_0\right)^2\right] \quad \text{for} \quad r_{ij} \leq R_0 \\ U_{ij}^{FENE} &= \infty \qquad\qquad\qquad\qquad\qquad\quad \text{for} \quad r_{ij} \geq R_0 \end{aligned} \tag{4.6}$$

where R_0 is a finite extensibility and k is a spring constant. In this work, these FENE parameters, R_0 and k, were set to 1.5 and 30 respectively. For this choice of parameters the maximal extent of bonds is short enough to prevent crossing of branches, whereas the magnitude of the bonding force is small enough to enable simulations with relatively large time steps (Bosko et al., 2004a, Kroger et al., 1993). The FENE chain model is not only suitable for entangled and unentangled polymer melts but is also applicable to monodisperse and polydisperse polymers.

Figure 4.3 illustrates the interaction potential between non-bonded and bonded beads. Non-bonded beads only have WCA potential interactions whereas bonded beads have both FENE and WCA interactions which creates a potential well for the flexible bonds that maintains the architecture of the molecules.

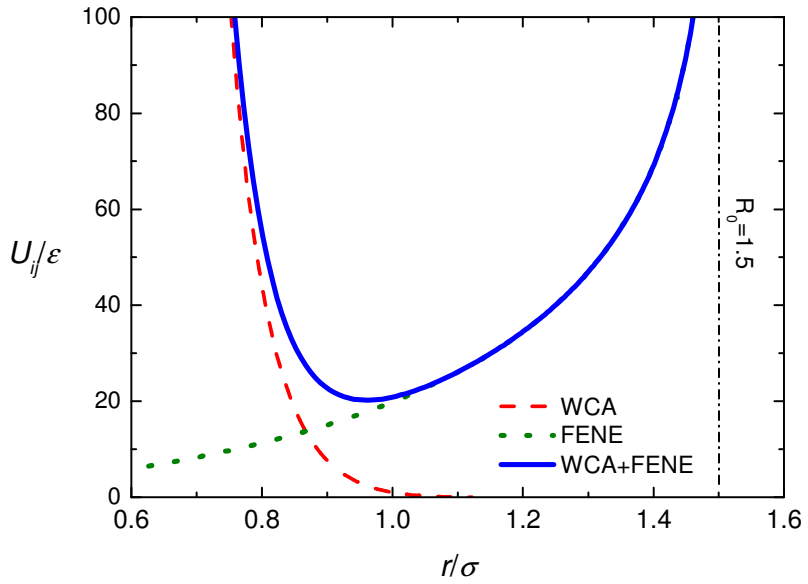

Figure 4.3. Interaction potential between non-bonded beads which interact via only the WCA potential and bonded beads which interact via both WCA and FENE potential.

From the interaction potentials between beads, the force acting on each bead for the system of beads at position $\mathbf{r}_i$ can be calculated as:

$$\mathbf{F}_i = -\nabla_{\mathbf{r}_i}\left(\sum_{bonds} U_{bond}\left(r_{ij}\right) + \sum_{i=1}^{N_{total}}\sum_{j>i}^{N_{total}} U_{non-bond}\left(r_{ij}\right)\right) \tag{4.7}$$

where N_{total} is the total number of interacting beads. The overwhelming proportion of the computing effort is spent in the calculation of the forces on each atom/bead.

All quantities reported hereafter in this thesis will be presented in reduced LJ units:

$$r_{ij}^{*} = r_{ij}/\sigma, \tag{4.8}$$

$$\rho^{*} = \rho\sigma^{3}, \tag{4.9}$$

$$T^{*} = k_B T/\varepsilon, \tag{4.10}$$

$$\mathbf{P}^{*} = \mathbf{P}\sigma^{3}/\varepsilon, \tag{4.11}$$

$$\dot{\gamma}^{*} = \left(m_\alpha\sigma^2/\varepsilon\right)^{1/2}\dot{\gamma}, \tag{4.12}$$

$$\eta^{*} = \left(\sigma^4/m_\alpha\varepsilon\right)^{1/2}\eta, \tag{4.13}$$

with both ε and σ being assigned a value of one, ρ is the bead density, T is the kinetic temperature, $\mathbf{P}$ is the pressure tensor, $\dot{\gamma}$ is the strain rate, and η is the shear viscosity. In

addition, the mass of the beads was assigned to one. For simplicity of notation, the asterisk will be omitted hereafter.

4.4. Periodic boundary conditions

Molecular dynamics is typically applied to small systems containing thousands of particles. Unless the behaviours near the walls (surface effects) are of interest, they can be eliminated and periodic boundary conditions are employed to simulate the bulk material. Particles are generated in a volume V which is called the primary cell. The bulk is assumed to be composed of the primary cell surrounded by its exact replicas to model a macroscopic sample. The image cells not only have the same size and shape as the primary one but also contain particles that are images of the particles in the primary cell. Cells are separated by open boundaries so particles can freely enter or leave any cell. When particles leave the cell, their images simultaneously enter the cell through the opposite face. Therefore the shape of the cells must be space filling. The number of image cells needed depends on the range of intermolecular forces. When the forces are sufficiently short ranged (e.g. in truncated Lennard-Jones model), only image cells that adjoin the primary cell are needed (minimum image convention). For squares in two dimensions, there are eight adjacent image cells whereas for cubes in three dimensions, the number of adjacent images is 26 (Haile, 1997). Figure 4.4 illustrates the periodic boundary condition in two dimensions with the primary cell surrounded by eight of its image cells.

For simulations of shear flows, the application of the standard periodic boundary condition would lead to a discontinuity in the coordinates and momenta on the edge of the simulation box. Therefore an appropriate modification is necessary in order to maintain a steady linear velocity profile for planar shear flows. In this work, molecules were generated in a cubic simulation box which was surrounded by its periodic images and Lees-Edwards shear boundary conditions (Lees and Edwards, 1972) were applied to eliminate effects associated with surfaces and the small volume of the system.

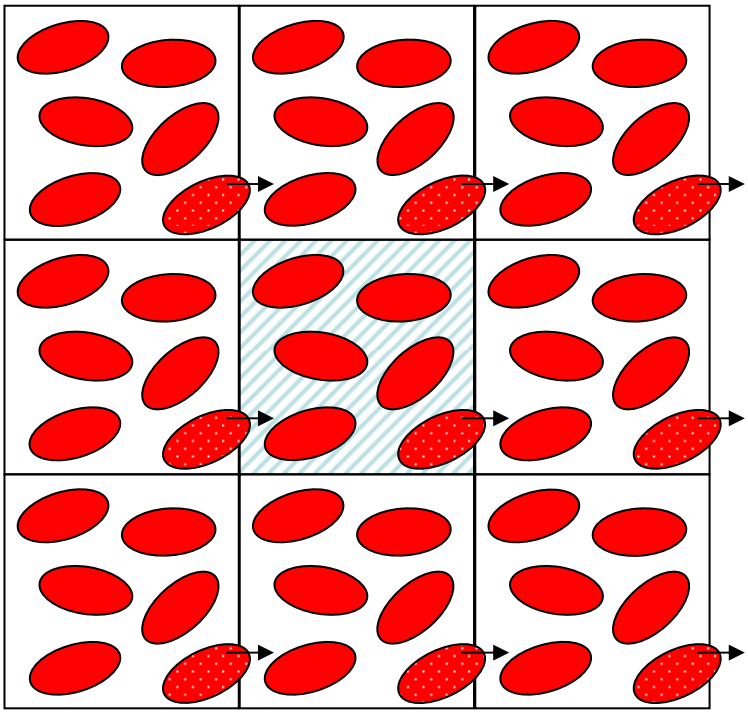

Figure 4.4. Periodic boundary condition in two dimensions with the primary cell surrounded by its image cells. Molecules that leave the cell will be replaced by their images entering the cell from the opposite side.

It can be seen in Figure 4.5 that for the linear velocity profile with the streaming velocity in the *x* direction and its gradient in the *y* direction, the cells in the *x* and *y* directions are still periodically repeated. However in the *y* directions, the image cells are displaced with reference to the central one according to the streaming velocity. The displacement of these cells with reference to the central cell is given by:

$$D = \mathbf{i}\dot{\gamma}L_y \tag{4.14}$$

where L_y is the size of the simulation box along the *y* axis in the absence of any other external forces. The motion of the periodic images of particles above and below the simulation box after some time would induce a linear streaming velocity profile.

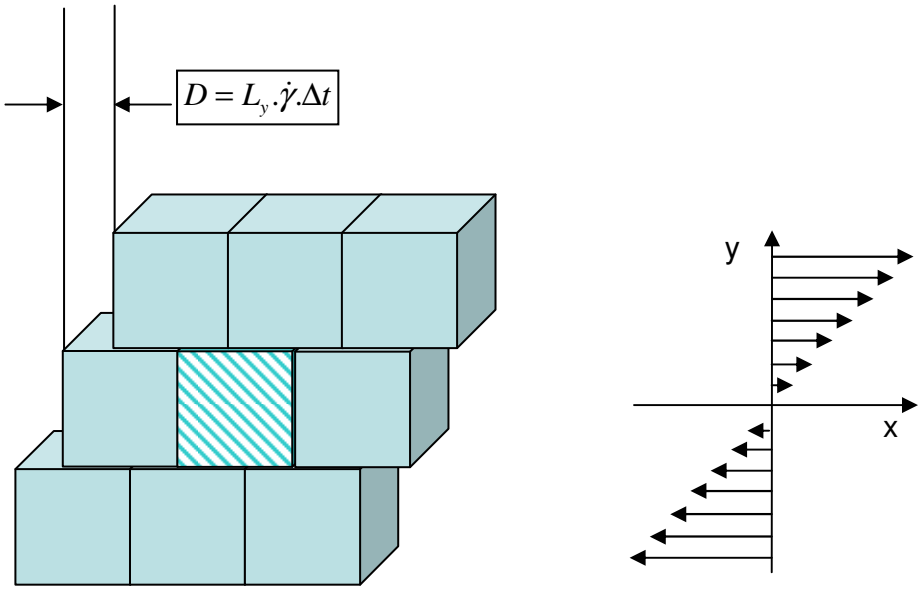

Figure 4.5. Lees-Edwards periodic boundary conditions for steady shear.

4.5. Equations of motion

There are many ways of writing the equations of motion for systems composed of interacting molecules (Allen and Tildesley, 1987).

The most fundamental form is the Lagrangian equation of motion which is given as:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \left(\frac{\partial L}{\partial q_k}\right) = 0 \tag{4.15}$$

where $L(\mathbf{q}, \dot{\mathbf{q}})$ is the Lagrangian function defined in terms of kinetic energy K and potential energy U as $L(\mathbf{q}, \dot{\mathbf{q}}) = K - U$. The Lagrangian is a function of the generalized coordinates q_k and their time derivatives $\dot{q}_k$. If a system of particles with mass m, in the Cartesian coordinates $\mathbf{r}_i$, is considered, the Lagrangian equation of motion reduces to Newton's second-order differential equation of motion:

$$m\ddot{\mathbf{r}}_i = \mathbf{F}_i \,. \tag{4.16}$$

Defining the generalized momentum p_k conjugate to q_k as:

$$p_k = \frac{\partial L}{\partial \dot{q}_k} \tag{4.17}$$

the Hamiltonian is then defined as:

$$H(\mathbf{p},\mathbf{q}) = \sum_k \dot{q}_k p_k - L(\mathbf{q},\dot{\mathbf{q}}) \tag{4.18}$$

Therefore the Hamiltonian form of the equations of motion is given as:

$$\begin{aligned} \dot{q}_k &= \frac{\partial H}{\partial p_k} \\ \dot{p}_k &= -\frac{\partial H}{\partial q_k} \end{aligned} \tag{4.19}$$

The advantage of the Hamiltonian equations of motion is that for Cartesian coordinates, Hamilton's equations reduce to a set of first-order differential equations given by:

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} \\ \dot{\mathbf{p}}_i &= -\nabla_{\mathbf{r}_i} V = \mathbf{F}_i \end{aligned} \tag{4.20}$$

The solution of the Lagrangian and Hamiltonian equations of motion yields the complete dynamics of the system.

For the purpose of simulating systems undergoing shearing flow, therefore away from equilibrium, two different approaches can be applied.

One of the two common methods to derive transport properties is through analyzing the equilibrium correlation functions. The disadvantage of this approach is the significant statistical errors with time correlation functions representing the average response to naturally occurring small fluctuations in the system properties and the unfavorable signal-to-noise ratio at long times, where there may be a significant contribution to the integral defining a transport coefficient. The finite system size also limits the maximum time for which reliable correlations can be calculated (Allen and Tildesley, 1987).

The second method that can be employed to study systems away from equilibrium is to sample a non-equilibrium ensemble directly. A perturbation field is incorporated into the classic molecular dynamics model. The advantages of this approach are a much larger fluctuation induced and dramatically improved signal-to-noise level of the measured response. In order to simulate fluids under shear, two different methods can

be employed. The first one is called a boundary-driven algorithm in which the periodic boundary conditions are modified to simulate systems far from equilibrium. The most useful of these is the Lees-Edwards boundary conditions. However this algorithm has two serious shortcomings (Allen and Tildesley, 1987). Firstly there is no connection with response theory, hence links to the statistical mechanics of transport are not possible. Furthermore long times are required for the effects of translation of atoms between boundaries to communicate throughout the fluids or through the simulation cell. Therefore a linear streaming velocity will only evolve after a sufficiently long time and time-dependent transport properties cannot be studied (Todd and Daivis, 2007). Alternatively a synthetic field method such as the SLLOD algorithm (Evans and Morriss, 1990) can be used to study systems under shear. The history of this algorithm can be traced back to 1980 when the first homogenous NEMD algorithm, the DOLLS Hamiltonian, was invented (Hoover et al., 1980). In this approach, the effect of a boundary that drives the flow is replaced by a fictitious external field which is designed to guarantee that the required streaming velocity profile is maintained indefinitely. The flow field is incorporated into the equations of motion which are given as:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m} + \mathbf{r}_i \cdot \nabla \mathbf{u} \tag{4.21}$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \nabla \mathbf{u} \cdot \mathbf{p}_i$$

where $\dot{\mathbf{r}}_i$ and $\dot{\mathbf{p}}_i$ are the position and thermal momentum of an atom/bead α in molecule i, $\mathbf{F}_i$ is the interatomic force on atom i and $\nabla \mathbf{u}$ is the velocity gradient tensor which takes the form:

$$\nabla \mathbf{u} = \begin{pmatrix} \dot{\varepsilon}_{xx} & \dot{\gamma}_{xy} & \dot{\gamma}_{xz} \\ \dot{\gamma}_{yx} & \dot{\varepsilon}_{yy} & \dot{\gamma}_{yz} \\ \dot{\gamma}_{zx} & \dot{\gamma}_{zy} & \dot{\varepsilon}_{zz} \end{pmatrix} \tag{4.22}$$

This algorithm is suitable for simulating flows in the linear response limit but not for generating physically realistic shear flow at high field strengths (Evans and Morriss, 1984). It was then proposed that the correct set of equations of motion was given by:

$$\dot{\mathbf{r}}_i = \frac{\mathbf{p}_i}{m} + \mathbf{r}_i \cdot \nabla \mathbf{u} \tag{4.23}$$

$$\dot{\mathbf{p}}_i = \mathbf{F}_i - \mathbf{p}_i \cdot \nabla \mathbf{u}$$

and named SLLOD to indicate the transpose of DOLLS.

In this work, the SLLOD algorithm has been applied to simulate hyperbranched polymer fluids under shear. Depending on the velocity gradient tensor, the SLLOD algorithm will correspond to different forms of flow.

When only diagonal elements of the velocity gradient tensor are nonzero (either positive or negative), the equations of motion will correspond to the expansion or compression at a steady state.

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m} + \mathbf{r}_i \cdot \nabla \mathbf{u} \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \mathbf{p}_i \cdot \nabla \mathbf{u} \end{aligned} \qquad \nabla \mathbf{u} = \begin{pmatrix} \dot{\varepsilon}_{xx} & 0 & 0 \\ 0 & \dot{\varepsilon}_{yy} & 0 \\ 0 & 0 & \dot{\varepsilon}_{zz} \end{pmatrix} \tag{4.24}$$

This approach is used in this work to prepare the initial configuration for simulations of polymer melts under shear. Systems of polymer molecules are first generated at a low density and then compressed until the required density is achieved.

When only one off-diagonal element of the velocity gradient tensor is nonzero, the SLLOD algorithm corresponds to planar Couette flow:

$$\begin{aligned} \dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m} + \mathbf{r}_i \cdot \nabla \mathbf{u} \\ \dot{\mathbf{p}}_i &= \mathbf{F}_i - \mathbf{p}_i \cdot \nabla \mathbf{u} \end{aligned} \qquad \nabla \mathbf{u} = \begin{pmatrix} 0 & 0 & 0 \\ \dot{\gamma}_{yx} & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad \frac{\partial u_x}{\partial y} = \dot{\gamma}_{yx} \tag{4.25}$$

This is the standard approach employed in this work to study the melts of different hyperbranched polymers under shear.

Using the SLLOD algorithm has many advantages (Allen and Tildesley, 1987, Allen and Tildesley, 1993). Firstly, the SLLOD equations of motion only involve the coordinates and the peculiar momenta of the particles. Furthermore the thermostat, internal energy and pressure involve peculiar, rather than laboratory velocities. Secondly, the SLLOD momenta are continuous as particles leave and enter the primitive cell, so the calculation is more stable than it otherwise might be. Finally, the SLLOD equations transform a boundary driven system into one with an external field, which makes the system more amenable to theoretical analysis.

However in order to apply the above atomic SLLOD algorithm for molecular systems, the equations of motion need to be redefined as the streaming velocity is imposed on the molecule and all its parts according to the position of its centre of mass. This is different

from the atomic SLLOD where the linear velocity profile is imposed on atoms/beads according to their current position. Nevertheless, in the steady state, both atomic and molecular methods produce the same results if the atomic/bead streaming velocity is correctly accounted for (Edberg et al., 1986, Edberg et al., 1987).

The atomic version of the SLLOD equations of motion for molecular systems is given as:

$$\dot{\mathbf{r}}_{i\alpha} = \frac{\mathbf{p}_{i\alpha}}{m_{i\alpha}} + \mathbf{r}_{i\alpha} \cdot \nabla \mathbf{u} \tag{4.26}$$

$$\dot{\mathbf{p}}_{i\alpha} = \mathbf{F}_{i\alpha} - \mathbf{p}_{i\alpha} \cdot \nabla \mathbf{u}$$

where $\mathbf{r}_{i\alpha}$ and $\mathbf{p}_{i\alpha}$ are the position and momentum of an atom/bead α in molecule *i* and $m_{i\alpha}$ is an atomic mass.

The molecular version of the SLLOD equations of motion, which is the form of the algorithm used in this work, is given by:

$$\dot{\mathbf{r}}_{i\alpha} = \frac{\mathbf{p}_{i\alpha}}{m_{i\alpha}} + \mathbf{r}_{i} \cdot \nabla \mathbf{u} \tag{4.27}$$

$$\dot{\mathbf{p}}_{i\alpha} = \mathbf{F}_{i\alpha} - \left(m_{i\alpha} / M_{i}\right) \mathbf{p}_{i} \cdot \nabla \mathbf{u}$$

where N_s is the number of atoms/beads in a molecule, M_i is the molecular mass, $\mathbf{r}_i$ is the position of the molecular centre of mass, and $\mathbf{p}_i$ is momentum of the molecule. These quantities can be defined as:

$$M_i = \sum_{\alpha=1}^{N_\alpha} m_{i\alpha} \tag{4.28}$$

$$\mathbf{r}_i = \sum_{\alpha=1}^{N_\alpha} m_{i\alpha} \mathbf{r}_{i\alpha} / M_i \tag{4.29}$$

$$\mathbf{p}_i = \sum_{\alpha=1}^{N_\alpha} \mathbf{p}_{i\alpha} \tag{4.30}$$

In constant temperature simulations, the equations of motion are modified and given by:

$$\dot{\mathbf{r}}_{i\alpha} = \frac{\mathbf{p}_{i\alpha}}{m_{i\alpha}} + \mathbf{r}_{i} \cdot \nabla \mathbf{u} \tag{4.31}$$

$$\dot{\mathbf{p}}_{i\alpha} = \mathbf{F}_{i\alpha} - \left(m_{i\alpha} / M_{i}\right) \mathbf{p}_{i} \cdot \nabla \mathbf{u} - \zeta (m_{i\alpha} / M_{i}) \mathbf{p}_{i}$$

where ζ is the thermostat constraint multiplier which is given as:

$$\zeta = \frac{\sum_{i=1}^{N} \mathbf{F}_i \cdot \mathbf{p}_i - \dot{\gamma} \sum_{i=1}^{N} p_{ix} p_{iy}}{\sum_{i=1}^{N} \mathbf{p}_i^2} \tag{4.32}$$

This molecular version of the Gaussian isokinetic thermostat ζ is used to keep the translational kinetic temperature fixed. ζ is determined by the condition that the molecular kinetic energy is constant and the deviations of the trajectories from the unthermostated ones is minimized The justification for using the molecular thermostat algorithm is discussed in detail elsewhere (Todd and Daivis, 2007).

For constant pressure simulations, the equations of motion (Todd and Daivis, 2007) are given by:

$$\begin{aligned} \dot{\mathbf{r}}_{i\alpha} &= \frac{\mathbf{p}_{i\alpha}}{m_{i\alpha}} + \mathbf{r}_i \cdot \nabla \mathbf{u} + \dot{\varepsilon} \mathbf{r}_i \\ \dot{\mathbf{p}}_{i\alpha} &= \mathbf{F}_{i\alpha} - \left(m_{i\alpha} / M_i\right) \mathbf{p}_i \cdot \nabla \mathbf{u} - \zeta \left(m_{i\alpha} / M_i\right) \mathbf{p}_i - \dot{\varepsilon} \left(m_{i\alpha} / M_i\right) \mathbf{p}_i \\ \dot{V} &= 3 \dot{\varepsilon} V \end{aligned} \tag{4.33}$$

where $\dot{\varepsilon}$ is the multiplier which depends on the difference between the instantaneous pressure p, the required pressure p_0 and a damping factor Q chosen by trial and error to give good damping of the pressure fluctuations as follows:

$$\ddot{\varepsilon} = \frac{\left(p - p_0\right) V}{Q N k_B T}. \tag{4.34}$$

In all constant pressure simulations of hyperbranched polymers in this work, the damping factor was chosen to be 1000 to minimize the effect of unphysical oscillations of volume and molecular pressure caused by coupling the simulated system to the barostat. Therefore the melts are allowed to relax to compensate for the flow-induced changes in the pressure.

4.6. Integration

As the most computationally intensive component of the computation is the force evaluation, any integration method requiring more than one such calculation per time-step is wasteful unless it can deliver a proportionate increase in the size of the time-step

Δt while maintaining the same accuracy. Therefore the well-known Runge-Kutta method is not suitable as it is unable to enlarge the time-step beyond the limit of Δt due to the strongly repulsive force at short distances in the Lennard-Jones potential (Haile, 1997, Rapaport, 1995). The multiple time-step algorithms (Tuckerman et al., 1992), in which the potential is divided into the short range interactions with a shorter time step and the long range interactions with a much longer time step, have been shown to be efficient in molecular dynamics simulations (Frenkel and Smit, 1996).

There are two common algorithms, which are used to solve the equations of motion, namely the Gear predictor-corrector and the Verlet algorithms, as both of these algorithms require only one evaluation of the forces at each time step.

The Verlet algorithm uses information from the current and previous time steps to advance the atomic positions (Allen and Tildesley, 1993). The algorithm can be constructed by considering the two Taylor expansions from t to $t+\Delta t$ and $t-\Delta t$. Adding or subtracting these equations produces the equation for advancing the positions or velocity. Some modifications to the Verlet scheme, such as 'leap-frog' and 'velocity-Verlet' algorithms, have been suggested to try to improve the method of handling the velocities. These algorithms are simpler to use and require less memory than the Gear predictor-corrector (Allen and Tildesley, 1987, Allen and Tildesley, 1993).

The Gear predictor-corrector method is composed of three steps (Allen and Tildesley, 1987, Allen and Tildesley, 1993). From a significant number of time derivatives of the atomic positions at time *t*, fourth-order Taylor series expansions are used to predict the position at time $t+\Delta t$. For the first-order differential equation:

$$\dot{\mathbf{r}} = f(\mathbf{r}) \tag{4.35}$$

the predictor step would have the form:

$$\begin{pmatrix} \mathbf{r}_0^p(t+\Delta t) \\ \mathbf{r}_1^p(t+\Delta t) \\ \mathbf{r}_2^p(t+\Delta t) \\ \mathbf{r}_3^p(t+\Delta t) \\ \mathbf{r}_4^p(t+\Delta t) \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 1 & 2 & 3 \\ 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{r}_0^p(t) \\ \mathbf{r}_1^p(t) \\ \mathbf{r}_2^p(t) \\ \mathbf{r}_3^p(t) \\ \mathbf{r}_4^p(t) \end{pmatrix} \tag{4.36}$$

where $\mathbf{r}_i$ are the scaled time derivatives: $\mathbf{r}_1 = \delta t\left(d\mathbf{r}_0/dt\right), \mathbf{r}_2 = (1/2)\delta t^2\left(d^2\mathbf{r}_0/dt^2\right)$, $\mathbf{r}_3 = (1/6)\delta t^3\left(d^3\mathbf{r}_0/dt^3\right)$ and $\mathbf{r}_4 = (1/24)\delta t^4\left(d^4\mathbf{r}_0/dt^4\right)$. The order of the method depends on the number of elements taken into account in the expansion.

In the evaluation step which is the most time consuming, the forces at $t+\Delta t$ are computed using the series of predicted positions and their derivatives to give accelerations. The difference between the predicted value of the first scaled derivative $\mathbf{r}_1^p$ and the correct one $\mathbf{r}_1^c$ calculated by the substitution of $\mathbf{r}_0^p$ into the equation of motion (the right side of the equation) can be derived from:

$$\Delta\mathbf{r} = \mathbf{r}_1^c - \mathbf{r}_1^p \tag{4.37}$$

and used in the corrector step.

In the final step, the corrected values at the end of the integration step take the form:

$$\begin{pmatrix} \mathbf{r}_0(t+\Delta t) \\ \mathbf{r}_1(t+\Delta t) \\ \mathbf{r}_2(t+\Delta t) \\ \mathbf{r}_3(t+\Delta t) \\ \mathbf{r}_4(t+\Delta t) \end{pmatrix} = \begin{pmatrix} \mathbf{r}_0^p(t+\Delta t) \\ \mathbf{r}_1^p(t+\Delta t) \\ \mathbf{r}_2^p(t+\Delta t) \\ \mathbf{r}_3^p(t+\Delta t) \\ \mathbf{r}_4^p(t+\Delta t) \end{pmatrix} + \begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \Delta\mathbf{r} \tag{4.38}$$

The difference between the predicted and corrected accelerations is used with a set of Gear coefficients c_i to correct all of the predicted derivatives of the motion. These Gear coefficients are chosen to optimize the stability and accuracy of the trajectories. They depend on the order of the differential equation and the number of derivatives used in the Taylor series prediction of the positions. Gear has discussed and proposed the best choice, and for the fourth-order algorithm applied to the first-order differential equation the values are: $c_0 = 251/720$, $c_1 = 1$, $c_2 = 11/12$, $c_3 = 1/3$, $c_4 = 1/24$ (Gear, 1971).

This method offers great flexibility and good stability as the corrector step amounts to a feedback mechanism that can dampen instabilities that might be introduced by the predictor. For computing particle trajectories using small time-steps, this algorithm is more accurate than the Verlet algorithm (Haile, 1997).

4.7. Simulation conditions

In this work, melts of hyperbranched polymers of different molecular weights and architecture have been simulated. The original program for simulating our polymer melts was written by Bosko (Bosko, 2005) and modified in this project to model a variety of hyperbranched polymers and their blends with linear chains. Samples of these dense fluids are prepared and modelled under shear. For polymers of the smallest size, the sample consists of 216 molecules whereas for all other polymers, the sample comprises 125 molecules. The total number of beads in the simulation box varies from approximately 4000 to 23,500. The system is first generated at low density and then compressed until the desired density is achieved. The SLLOD algorithm with equal negative diagonal elements of the velocity gradient tensor has been applied during this density equilibration phase.

Constant temperature simulations were performed at the constant density of 0.84 and temperature of 1.25. The reason for choosing these parameters is that at this thermodynamic state point, the monomer WCA fluid is a dense fluid. This suggests that the molecular WCA fluid is also a liquid under the same conditions (Bosko, 2005).

Constant pressure simulations were performed at the pressure of 5.42. This is the equilibrium pressure of the generation 2 dendrimer melt at the density of 0.84. Modelling hyperbranched polymers with the same conditions allows comparison between our results with those for dendrimers and linear polymers reported previously (Bosko et al., 2005).

In this work, hyperbranched polymers of different molecular weights were simulated using the *NVT* ensemble. Hyperbranched polymers of the same molecular weight but with different architectures were modelled using both *NVT* and *NpT* ensembles. *NpT* simulations were also performed for the blends of hyperbranched polymers and linear chains composed of the same number of particles.

The SLLOD equations of motion were integrated with a time-step $\Delta t = 0.001$ in reduced units using a fourth-order Gear predictor corrector differential equation solver. This value of the time-step is sufficiently small to ensure that the simulation results do not depend on it.

The range of strain rates was chosen to be from 0.0001 up to 0.2 as with this wide range, the transition of the polymeric fluid from Newtonian to non-Newtonian regimes can be captured.