A parallel Lowe-Andersen thermostat for soft matter simulations

K. R. Prathyusha
Department of Physics,
Indian Institute of Technology Madras
Chennai, India 600036
prathyusha@physics.iitm.ac.in

P. B. Sunil Kumar
Department of Physics
Indian Institute of Technology Madras
Chennai, India 600036
sunil@iitm.ac.in

ABSTRACT
Simulating colloidal dynamics, with solvent mediated hydrodynamics interactions, is a challenging task. The traditional approach is to use Dissipative Particle Dynamics thermostat and soft interactions. However this can yield only low Schmidt number Sc flows. We have implemented an MPI version of Lowe-Andersen thermostat, with soft interactions, to obtain high Sc flows. Some results from the simulations of polymers, with this thermostat will be discussed.

Categories and Subject Descriptors
D.1.3 [Concurrent Programming]: Parallel programming; J.2 [Physical sciences and engineering]: Physics

General Terms
Algorithms, Performance

Keywords
Molecular Dynamics, Thermostats, MPI programing

1. INTRODUCTION
Complex fluids such as polymer solutions, micellar systems, colloidal suspensions and micro emulsions manifest many dynamical behaviours, where hydrodynamics plays an important role. The characteristic time scales in these systems can be in the range from nanosecond to seconds and the length scales from nano to micrometers. Thus a variety of models, at different length and time scales, are required to study their dynamics. Any model that fulfill the conservation of particle number, momentum, isotropy and Galilean invariance will reproduce hydrodynamic behavior on long enough length and time scales. Molecular Dynamics (MD) simulations being a popular tool to elucidate the properties of soft matter systems such as polymers, colloids, bio-membranes etc., is a natural choice for simulating hydrodynamic flows at the microscopic level.

The conventional MD simulation describes a dynamical system, which is microcanonical NVE ensemble by default. The trajectory of the system is governed by Newton’s equations of motion, given by

$$\dot{r}_i = \frac{p_i}{m_i} \text{ and } \dot{p}_i = F_i$$  \hspace{1cm} (1)

where $r_i$ and $p_i$ are the position and momentum of particle $i$ with mass $m_i$. $F_i$ is the total force on particle $i$. But in many cases it is more realistic to simulate system at constant temperature and volume (NVT ensemble), because most of the experimental conditions are NVT in nature. Here we discuss some of the popular thermostats used for NVT simulations.

In the Andersen approach [2] to isothermal MD, the system is coupled to a heat bath. Newton’s equations of motion are first integrated in time, with time step $\Delta t$. The system maintains the temperature by stochastic collision of randomly selected particles with the heat bath. The velocity of particles undergoing collisions with the heat bath are reassigned from the Maxwell-Boltzmann distribution, with a probability $\Gamma \Delta t$, where $\Gamma$ is the bath collision frequency. With this stochastic coupling method we can arrive at a canonical distribution [2]. The Andersen method is non-deterministic and time-irreversible. It doesn’t produce smooth trajectories due to the random collisions in the system. Thermostat acts locally in the system. Obviously there is no local momentum conservation in this method and it is not expected to preserve hydrodynamic modes. Similar is the case with the Stochastic Dynamics (SD) thermostat [9], where each particle coupled to a stochastic heat bath in an implicit background solvent, such that

$$\dot{r}_i = \frac{p_i}{m_i} \text{ and } \dot{p}_i = F_i - \zeta \frac{p_i}{m_i} + f_i$$  \hspace{1cm} (2)

where $\zeta$ is friction constant and $f_i$ is stochastic force with zero mean and satisfies the following relationship,

$$< f_i(t) f_j(t') > = 2 \zeta k_B T \delta_{ij} \delta(t - t').$$

This algorithm violates Galilean invariance and doesn’t obey strict local momentum conservation [10].

Dissipative Particle Dynamics (DPD) is similar in spirit with SD, but now with pair-wise forces between the particles. The DPD was developed by Hoogerbrugge and Koelman [5]. And it is modified by Espanol and Warren[3] to get the correct thermal equilibrium state. The DPD equation of motion can be written as

$$\dot{r}_i = \frac{p_i}{m} \text{ and } \dot{p}_i = F_i + F_i^{D} + F_i^{R}$$  \hspace{1cm} (3)
where $F^D_i$ represents dissipative force, $F^R_i$ corresponds to random force. The forms of thermostating forces $F^D_i$ and $F^R_i$ are given by,

$$F^D_i = \sum_j F^D_{ij} = -\zeta \omega^D(r_{ij})(\mathbf{v}_{ij} - \mathbf{v}_i)$$

$$F^R_i = \sum_j F^R_{ij} = \sigma \omega^R(r_{ij})\theta_{ij} \Delta t^{-1/2}$$

where $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$ is the relative velocity between the particles $i$ and $j$, while $\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$, $\mathbf{r}_{ij} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$, $r = |r_{ij}|$, $\zeta$ is the friction constant and $\sigma$ the noise strength. The $\theta_{ij}$ is symmetric random noise with zero mean and unit variance. The amplitudes and weight functions in equations (4) and (5) are related as follows

$$\omega^D(r) = [\omega^R(r)]^2 \quad \text{and} \quad \sigma^2 = 2\zeta k_B T.$$ \hspace{1cm} (6)

And a simple choice for $\omega^D(r)$ is

$$\omega^D(r) = \begin{cases} 
(1 - r^2), & r < r_c \\
0, & r \geq r_c \end{cases}$$ \hspace{1cm} (7)

Thermostat now acts locally. Since all the forces are pairwise canceling, momentum is conserved locally and hydrodynamics is preserved. In normal DPD formalism, the form of conservative force, $F$ is linear soft repulsive one given by

$$F_i = \sum_j F^C_{ij} = \begin{cases} 
a_{ij}(1 - r/r_c), & r < r_c \\
0, & r \geq r_c \end{cases}$$ \hspace{1cm} (8)

where $a_{ij}$ is the interaction strength and $r_c$ is the force cutoff.

Since the interaction is soft, we can use a higher $\Delta t$ (comparable to other algorithms) which will end up in a faster simulation. However, the presence of soft interaction leads to low Schmidt number ($Sc$). The Schmidt number is the ratio between kinematic viscosity and self-diffusion coefficient ($Sc = \nu/D$). $\nu$ represents the rate at which the momentum is transported while $D$ depicts the rate at which particles move diffusively. For higher values of $Sc$ hydrodynamic interactions propagate rapidly in comparison with time taken for any particle to move. Groot and Warren [4] showed that in DPD simulation value of $Sc = 1.00 \pm 0.03$, which is indeed very low compared to water, which has $Sc = 10^3$. The origin of this disparity is due to the increased mobility of the particles, or the absence of cage effect which occur if we use a hard-core interaction. Lower values of $Sc$ can gives rise to dynamically unrealistic simulations. It can lead to artificial screening of hydrodynamics. One way around for this problem is increasing the cutoff radius $r_c$ for dissipative interaction. But it is computationally very inefficient, as the number of interacting pairs is proportional to the cube of the cut-off distance. Millan and Laradji [7] showed that another method to increase the $Sc$ is decreasing the exponent of $(1 - r)$ in equation (7). They pointed that $Sc$ is increased to $\sim 12.5$ when we decrease the exponent from 2 to 0.25. Another method to increase the $Sc$ is to use Lennard-Jones (LJ) type interaction instead of soft repulsive interaction. Soddemann et al. [10] simulated a system of LJ particles using DPD thermostat and estimated $Sc(\approx 30)$. Alternate method which works along similar lines as DPD to increase the Schmidt number is the Lowe-Andersen thermostat, which replaces the thermostating forces in DPD formalism. We will discuss this in detail in the next section.

### 1.1 Lowe Andersen thermostat

Lowe combined the essence of both Andersen thermostat and DPD and formulated a thermostat which conserves the momentum locally and enhances the viscosity of the fluid [6]. The main idea is to act the thermostat on relative velocity of particle pairs. To thermalize the system we take all the pairs of particles within the cut-off distance $r_{ij} < r_c$, and determine with a probability $\Gamma \Delta t$ whether to assign a new relative velocity from a Maxwell distribution. The algorithm is as follows:

- Solve Newton’s equations of motion for positions and velocities of the particles at time $t + \Delta t$ using the velocity-Verlet algorithm[1].

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \mathbf{v}_i(t) + \frac{\Delta t^2}{2m} F^D_i(t)$$ \hspace{1cm} (9)

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{m} (F^D_i(t) + F^R_i(t + \Delta t))$$ \hspace{1cm} (10)

- Generate a relative velocity $\mathbf{v}_{ij}, \mathbf{r}_{ij}$ from a distribution $\xi_{ij}/\sqrt{(2m\Delta t/m)}$ for all pair of particles for which $r_{ij} < r_c$, with a probability $\Gamma \Delta t$, where $\xi_{ij}$’s are Gaussian random numbers with zero mean and unit variance.

- Change the velocity according to the following equation

$$\mathbf{v}_i = \mathbf{v}_i + \Delta \mathbf{v}_{ij}; \quad \mathbf{v}_j = \mathbf{v}_j - \Delta \mathbf{v}_{ij},$$ \hspace{1cm} (11)

where, $2\Delta \mathbf{v}_{ij} = \mathbf{v}_{ij}(\mathbf{v}_{ij} - \mathbf{v}_j - \Delta \mathbf{v}_{ij}).\mathbf{r}_{ij}$.

Lowe calculated the value of $Sc$ and found that $Sc$ as high as $10^3$ could be obtained and that $Sc \propto \Gamma^2$. The action of thermostat can be considered as an impulsive force which contributes to the stress and enhances the viscosity of the solvent.

### 2. MPI PARALLEL ALGORITHM FOR LOWE-ANDERSEN THERMOSTAT

Serial code is not a practical solution when one has to simulate millions of particles. Parallel algorithms enable ones to do a system containing larger number of particles. To the best of our knowledge there is no parallel version of Lowe-Andersen thermostat available in literature. We use LAMMPS [8], which uses spatial decomposition, to carry out our simulations. So each processor computes the positions and velocities of all atoms within its domain at each time step. The part that distinguishes the LAT from other thermostats is the velocity update given in equation (11). This collision step requires information about the velocity of the neighbors.

Particles which are local to each domain as well as particles which are interacting across each domain boundary undergo collision according to the Lowe’s scheme. When particles collide across boundaries, the processors need to know the updated velocities of the particles at that instant. Since we do not a priori know if a particle involved in a collision is at the boundary of the domain, whenever there is a collision, we need to communicate the velocity of the particles. This puts high load on the communication. Here we propose a modified LAT algorithm, in order to make it more suitable for parallel simulations.


2.1 Modified Algorithm for Lowe-Andersen thermostat

The work around is to communicate and update the velocities after every few collision events \( n_p \). To minimize errors it is important to keep the number of boundary particles undergoing collision between any two communications \( n_b \) as low as possible. In the original LAT scheme, since pairs are picked up sequentially, \( n_b \) is unevenly distributed. Thus changing \( n_p \) does not reduce the errors. The first step is to choose a scheme such that \( n_b \) is evenly distributed and choose \( n_p \) such that \( n_b \sim 1 \). This will involve picking up \( n_p \) pairs randomly with uniform probability within each domain. The velocity of particles, in each pair thus picked up, is then updated according to equation (11). We make sure that each particle is picked up only once in between communications. The parameter that controls \( Sc \) is now \( \epsilon = n_p \times n_c \), where \( n_c \) is the number of communication events within one time step. The mapping between \( \epsilon \) and \( \Gamma \) is then given by \( \epsilon = \Gamma \times \Delta t \times \text{the number of pairs in a domain} \). (We define \( \epsilon^* = \epsilon / (\Delta t \times \text{Total no. of pairs in each domain}) \) and we use this parameter in the plots for simplicity)

3. RESULTS

We have simulated a system of soft particles, with conservative force defined in equation (8), with \( a_{ij} = 25 e / r_{ij} \), where \( e \) and cut off distance \( r_c \) sets the energy and length scale respectively in the simulations. The density is set to \( \rho = 3.0 \epsilon^{-3} \), and temperature to \( k_B T = e \), in separate simulations using DPD and the modified LAT thermostats. We choose periodic boundary conditions in all 3 dimensions. All particles have same mass \( m \) which is the unit of mass. For DPD simulations, the random noise strength in equation (4) is chosen as \( \sigma = 3.0 (e^3 m / r_c^2)^{1/4} \). The smallest time scale in the simulation is then \( \tau = (m e^2 / r_c^2)^{1/2} \).

We carried out LAT simulations on single processor and compared the \( Sc \) obtained with the original version of LAT (where \( Sc \) is set by \( \Gamma \)) and the modified version (\( Sc \) set by \( \epsilon \)). For a given \( \Gamma \), we find the value of \( \epsilon \) to obtain the same \( Sc \) on a single processor. In figure 1 we show that \( Sc \) obtained from modified LAT, for a fixed \( \epsilon \), does not change with the number of processors. Having validated the parameter \( \epsilon \), we carry out MPI parallel simulations as a function of \( \epsilon \) and show that \( Sc \) increases monotonically (figure 2). Inset of figure 2 shows that the relation between \( \Gamma \) and \( \epsilon \) is linear as expected. The efficiency of the algorithm demonstrated in figure 3, where one can see the cpu time taken, for a simulation of 1000 time units, by the modified LAT on different number of processors for different values of \( \epsilon^* \).

Next we study the relaxation dynamics of a polymer chain in an explicit solvent interacting either through DPD or LAT. The self avoidance of the polymer is imposed through the short repulsive LJ interaction, \( U(r) = 4 \epsilon \eta \eta (\frac{r}{r_c})^{12} - (\frac{r}{r_c})^{6} \), with \( \epsilon_{LL} = \epsilon \) and \( \eta_{LL} = r_c \), and all beads with mass \( m \). The connectivity of the polymer chain is through harmonic springs \( U_{bond} = k (r - r_c)^2 \), with \( k = 100 \epsilon m r_c^{-2} \). Simulations were carried out on a polymer consisting of 50 beads in a box of size \( 32 \times 32 \times 32 \), with \( \Delta t = 0.005 \tau \). We calculate the autocorrelation of the end to end vector of the polymer \( R_1(t) \) both in the case of DPD and LAT (\( Sc \) is com-
Figure 3: Time taken for the modified LAT simulation, of 100τ duration, for different ε* in a system of size 40 × 40 × 40, with density ρ = 3.0τ−3.

earlier DPD simulations. From figure 5 one can see that the structures obtained are again similar at similar Sc.

![Figure 4: A slice of the isotropic network of living polymer obtained using DPD (left) and LAT (right) thermostat for a solution consisting of 5% polymer beads. Solvent is not shown for clarity. See reference [11] for details](image)

Figure 5: Sheared lamellar phase of living polymer using DPD (left) and LAT (right) thermostat for a solution consisting of 5% polymer beads with a shear rate Δτ = 1.0τ−1

4. CONCLUSIONS
We developed an MPI version of Lowe-Andersen thermostat. We validate its suitability by comparing the original serial version with the modified parallel version. We show that the viscosity parameter in the original and modified versions of LAT are linearly related. The algorithm is shown to exhibit good scaling with the number of processors. We compare the relaxation of polymer chain to show that the DPD and LAT simulations give comparable relaxation times at similar Sc. The new algorithm is also shown to reproduce results blue of living polymers obtained earlier using DPD.

5. ACKNOWLEDGMENTS
The computations were carried out at the High Performance Computing Facility at IIT Madras.

6. REFERENCES