Multiscale Monte Carlo for simple fluids

A. C. Maggs
Laboratoire de Physico-Chime Théorique, UMR CNRS-ESPCI 7083,
10 rue Vauquelin, 75231 Paris Cedex 05, France.

We introduce a multiscale Monte Carlo algorithm to simulate dense simple fluids. The probability of an update follows a power law distribution in its length scale. The collective motion of clusters of particles requires generalization of the Metropolis update rule to impose detailed balance. We apply the method to the simulation of a Lennard-Jones fluid and show improvements in efficiency over conventional Monte Carlo and molecular dynamics, eliminating hydrodynamic slowing down.

Both molecular dynamics and Monte Carlo are widely used in the study of fluids. The first, which is often preferred for large scale studies, allows one to access both static and dynamic information; however use of a large time step leads to systematic errors in the thermodynamics [1]. Monte Carlo has the advantage of converging to the exact equilibrium distribution without any systematic errors; recently the introduction of collective updates has led to substantial improvements in the efficiency of Monte Carlo simulation of fluids. Particularly notable is a cluster algorithm for hard spheres [2] and its generalization [3] to Lennard-Jones fluids. These papers introduced updates based on reflection or rotation of groups of particles with respect to a randomly chosen centre. The algorithms are particularly well adapted to the simulation of dilute, heterogeneous systems. They work less well for dense homogeneous fluids. In this Letter we introduce collective updates which simulate such fluids more efficiently.

Our method displaces blocks of particles on a scale, ℓ, which is intermediate between the dimensions of the particles and the simulation cell. By moving groups of particles we accelerate the dynamics of long wavelength fluctuations. Three technical problems are to be solved in the implementation of the method: Firstly we must move particles without substantially increasing the energy of the system; otherwise the Metropolis algorithm will refuse the trial. Secondly we create a set of moves that are *reversible*, in order to apply detailed balance. Thirdly we calculate the Jacobian that is implied by our collective moves, which complicates the counting of the number of states. We treat each of these three points before presenting an implementation and studying the relaxation of long wavelength density fluctuations.

Rigidly displacing a block of particles a distance ϵ generates a mismatch in the structure of the fluid at the interface between the moved and stationary particles. This mismatch rapidly leads to a high energy cost and low efficiency. We mitigate the problem by introducing updates in which the mismatch is *spread over the whole volume* ℓ^d in d dimensions. We choose updates along a single coordinate direction α

$$r'_{i,\alpha} = r_{i,\alpha} + \epsilon\, g\left(\frac{\mathbf{r}_i - \mathbf{c}_0}{\ell}\right), \qquad \forall i \tag{1}$$

for particles, i within ℓ of $\mathbf{c}_0$. The continuous function g has the properties $g(0) = 1$, and $g(\mathbf{r}) = 0$ for $|\mathbf{r}| > 1$. ϵ is a random amplitude, $\mathbf{c}_0$ a randomly chosen centre in the simulation volume. Thus only particles within ℓ of the point $\mathbf{c}_0$ move.

The proposed displacement, ϵg generates a map $x \to G(x)$ on the configuration space of the system, where x designates the $N \times d$ coordinates of the N particles. The simplest manner of generating dynamics that converge to the Boltzmann distribution is the imposition of *detailed balance* between pairs of states that are linked in both directions by a transformation. However as yet G *is not paired with an inverse transformation*, G^{-1}. We construct explicitly the inverse. We add to the moves eq. (1) a second set found by solving eq. (1) for $\mathbf{r}_i$ as a function of $\mathbf{r}'_i$. We then choose between direct and inverted moves with probability 0.5.

We now turn to the imposition of detail balance. Standard derivations consider a discrete space: In the presence of transitions between two states i, j with energies E_i and E_j one requires that

$$p(i \to j)\pi_i = p(j \to i)\pi_j \tag{2}$$

where the occupation probabilities are given by the Boltzmann weight $\pi_i = e^{-\beta E_i}/Z$. One solution for the transition rates $p(i \to j)$ is the Metropolis choice:

$$p(i \to j) = \min\left(1, e^{-\beta(E_j - E_i)}\right) \tag{3}$$

In the continuum one should count the number states in some neighborhood dx of x and not just work with the probability density at x. Consider two states x, x' linked by the transformations $x' = G(x)$, $x' = G^{-1}(x)$. The transformation distorts the the volumes dx, and dx' which are related by the Jacobian $J_G = \left|\frac{\partial(x')}{\partial(x)}\right|$. The pair $\{G, G^{-1}\}$ bring two volumes dx and dx/J_G into correspondence. The number of states near x is $\pi_x dx$; near x' there are $\pi_{x'} dx/J_G$. The transition probabilities should then satisfy

$$p(x \to x')\pi_x = p(x' \to x)\frac{\pi_{x'}}{J_G} \tag{4}$$

in order to generate the probability density π. This equation has a solution

$$p(x \to x') = \min\left(1, e^{-\beta(E_{x'} + T\log J_G - E_x)}\right) \tag{5}$$

with $T = \beta^{-1}$ the temperature. We note that Jacobian weighting factors are used in Monte Carlo simulations of polymers [4] when working with torsional degrees of freedom. Standard continuum Monte Carlo updates of the form $r'_{i,\alpha} = r_{i,\alpha} + \epsilon$ have $J_G = 1$ and fall within our formalism.

The final choices that need to be made concern ϵ and ℓ. The energy cost of a trial is estimated from the strain induced in the fluid by the transformation: $S \sim \epsilon \nabla g(\mathbf{r}/\ell) \sim \epsilon/\ell$. This energy varies as $\int S^2 \sim S^2 \ell^d \sim \epsilon^2 \ell^{d-2}$. We match this to the thermal energy and choose

$$\langle \epsilon^2 \rangle \sim \frac{T}{\ell^{d-2}} \qquad (6)$$

setting the scale of the random amplitude ϵ [12]. We sample ℓ from a distribution: Standard arguments give the density of states in Fourier space as $N(q)dq \sim q^{d-1}dq$. When this is expressed as a function of a length scale $\ell = 1/q$ we find $N(\ell)d\ell = d\ell/\ell^{d+1}$. We cut off the distribution at short distances $\ell < l_c$ and sample ℓ with probability

$$P(\ell) \sim \frac{l_c^d}{\ell^{d+1}}, \qquad l_c < \ell < L/2 \qquad (7)$$

Fluctuations at all length scales are then sampled at the same rate.

The asymptotic cost of the algorithm per trial can be estimated by recognizing that the effort needed to update the scale ℓ is $O(\ell^d)$ when interactions are short ranged. The average work per collective update is then $\int \ell^d P(\ell) d\ell \sim l_c^d \log(L/l_c)$, where L is the system size. The effort needed to perform the collective updates diverges only weakly with system size.

We implemented the algorithm, simulating Lennard-Jones particles with the potential

$$U = 4\epsilon_{LJ}\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \qquad (8)$$

truncating the interactions at a radius of 2.5σ and shifting U by a constant to eliminate the discontinuity [5]. We use length units of σ and energy units of ϵ_{LJ}. Temperature is measured in units of ϵ_{LJ}. Periodic boundary conditions were imposed.

A convenient choice for g is

$$g(\mathbf{r}/\ell) = 1 - r^2/\ell^2 \qquad (9)$$

for $r < \ell$, $g = 0$ otherwise. This allows the direct inversion of eq. (1). However, we also implemented more elaborate expressions for which analytic inversion was not possible. For these cases we performed a numerical inversion of eq. (1) using Newton iteration. Even here generating the trial moves was a minor contribution to the total computational cost of the algorithm.

The Jacobian is calculated as a product of individual particle contributions. For a displacement in the direction α

$$J_G = \prod_i (1 + \epsilon \partial g(\mathbf{r}_i)/\partial r_{i,\alpha}) \qquad (10)$$

The first simulations were performed on a very small system in order to check the correctness of our arguments as to the need for a Jacobian in the Metropolis criterion. To do this we multiplied the term in $\log J_G$ by a numerical prefactor γ. $\gamma = 1$ corresponds to the algorithm as described above, $\gamma = 0$ corresponds to neglecting the difference between the volumes dx and dx'. We took five particles in a two dimensional box of side $L = 10$ and simulated using a conventional Monte Carlo algorithm as well as the multiscale algorithm. As can be seen from Figure 1 use of the correct Jacobian is essential. These trials on small systems allowed the generation of high statistics runs, from which we checked the correctness of the code using both the energy and partial structure factors.

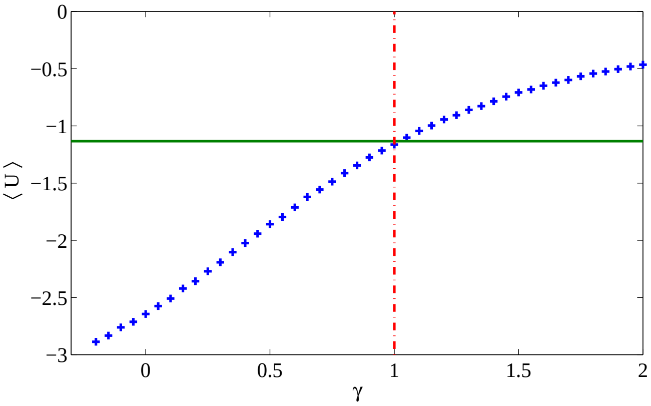

FIG. 1: Variation of the mean energy, $\langle U \rangle$, as a function of the prefactor of $\log J_G$. The correct average energy is generated with $\gamma = 1$. Errors result if one neglects the Jacobian, $\gamma = 0$. Solid line: local Monte Carlo. Points: multiscale Monte Carlo. Statistical errors small than symbol size. Vertical line guide to eye for $\gamma = 1$.

In order to study the efficiency of the algorithm we simulated a two-dimensional fluid with cell size $L = 60$, 80, 120 and 180 at a density $\rho = 0.755$, $T = 0.5$. We performed simulations with conventional, single particle Monte Carlo, molecular dynamics and multiscale Monte Carlo. In the simulations we measured the static structure factor $S_q = \left|\frac{1}{N}\sum_i e^{i\mathbf{q}.\mathbf{r}_i}\right|^2$. This correlation function is used to calculate the compressibility for $q \to 0$. In charged systems similar, long wavelength, polarization correlation functions must be measured with high accuracy in order to deduce the dielectric constant [6]. S_q is slow to equilibrate at small q due to the locality of particle motion in conventional Monte Carlo: A hydrodynamic description of the dynamics involves a local conserved quantity, the density, which diffuses: *i.e.* Model B dynamics [7]. Relaxation rates vary as $\Gamma_q = Dq^2$, with D the collective diffusion coefficient. Similarly in molecular dynamics propagating sound waves imply that $\Gamma_q^2 = c^2 q^2$.

Slow long wavelength fluctuations have consequences for the equilibration of the energy. For Monte Carlo the autocorrelation of the energy for the mode q can be expressed in the form $C_q(t) = \tilde{C}(Dq^2 t)$ with $\tilde{C}(0) = O(T^2)$, so that in d dimensions the two time energy-energy correlation function is given by $C_u = \int \tilde{C}(Dq^2 t)\, d^d q$. One finds $C_u(t) \sim 1/t^{d/2}$ [13]. It is the integral of C_u that determines the convergence rate of the average energy [8]. In two dimensions, where $\int C_u(t)\, dt$ diverges for large t, short ranged functions converge slowly due to hydrodynamics. Good thermodynamic statistics require that all modes in the sample are equilibrated. The situation is better in three dimensions, the integral is dominated by short wavelength fluctuations.

We verified that the algorithm worked correctly when only using updates from the multiscale distribution. However this turns out to be sub-optimal for the performance of the simulation. Standard Monte Carlo methods work very well at the scale of single particles; it is only at larger distances that they become slow. We stochastically mixed a standard Monte Carlo algorithm with the multiscale method using a lower cut off $l_c = 7.5$. We used a proportion of 1000 local updates for each multiscale update. With these choices of parameters the execution time of the program increases approximately 30% for $L = 40$ and 60% for $L = 180$ compared with the purely local code. Acceptance rates of all moves were adjusted to be approximately 40%. We verified that the acceptance rate of multiscale moves was almost independent of ℓ by binning acceptance. We determined the integrated autocorrelation times of S_q with a blocking algorithm [8] and plot the inverse time, Γ_q in Figure (2). As expected local Monte Carlo leads to slow relaxation of long wavelength modes, with $\Gamma_q = Dq^2$.

For the molecular dynamics simulations we used a time step $\tau = 0.006$ and a damping coefficient for the Langevin thermostat of 0.2. The time step is typical for low accuracy molecular dynamics studies, but larger than that which must be used for accurate studies of thermodynamic properties, for example $\tau = .0014$ in [9]. The dynamics (in Fourier space) of a system of particles coupled to an external thermostat lead to a dispersion law familiar from the damped harmonic oscillator [14]: $Kq^2 - i\eta\omega - \rho\omega^2 = 0$ with η a friction coefficient, K a compressibility and ρ the mass density. For large q modes are under-damped so that $\omega^2 = Kq^2/\rho$, corresponding to propagating sound waves. At small q we crossover to an over-damped "Brownian" regime where $\omega = iKq^2/\eta$. In our simulations we placed the crossover near $q^2 = 10^{-2}$ by performing several trial runs with different η. In this way we ensure propagative behavior over most of the range of the curve. At short length scales molecular dynamics is slower than local Monte Carlo, however it does a better job of relaxing long wavelength correlations. We adjusted the number of time steps in the molecular dynamics simulation so that the total simulation time was identical to the conventional Monte Carlo code.

The multiscale results are again displayed with a time scale of inverse sweeps, for a "clock time" comparison one must lower the values of Γ by a factor between 1.3 and 1.6 depending on L. The multiscale algorithm displays two regimes. At short wavelengths local Monte Carlo is active leading to standard q^2 behavior. At long wavelengths the algorithm eliminates hydrodynamic slowing down- all long wavelength modes relax at very similar rates. For $L = 180$ the multiscale algorithm relaxes long wavelength modes 40 times faster (clock time) than conventional local Monte Carlo. Multiscale is also faster than molecular dynamics at this scale.

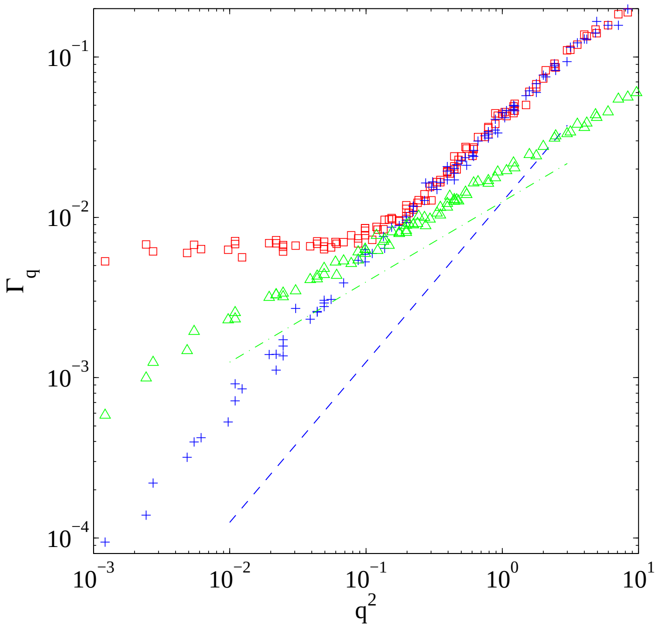

FIG. 2: Inverse integrated autocorrelation time in sweeps, Γ_q, for S_q. Conventional Monte Carlo: $+$. Multiscale: $\square$. Molecular dynamics: $\triangle$. Conventional algorithm has slow, diffusive modes. Multiscale has decay rate independent of q at long wavelengths. Dashed line: $\Gamma \sim q^2$; dot-dashed line $\Gamma \sim q$. Time scale for Monte Carlo inverse sweeps. System sizes from $L = 40$ to $L = 180$. Simulations of $2^{20} \approx 10^6$ sweeps, taking up to two weeks of simulation time for the largest systems.

Given the success of the algorithm we tried to find other updates which couple even more strongly to density fluctuations. We implemented a trial update in which motion is purely radial about a random centre $\mathbf{c}_0$. If $\mathbf{r}$ denotes the position of a particle with respect to $\mathbf{c}_0$ we tried

$$\mathbf{r}' = \mathbf{r} + \epsilon\, \mathbf{r}(1 - r/\ell) \tag{11}$$

for $r/\ell < 1$, $\mathbf{r}' = \mathbf{r}$ otherwise. The results were disappointing. Similar mediocre results were found for purely rotational updates of the form

$$\theta' = \theta + \epsilon\,(1 - r^2/\ell^2) \qquad (12)$$

with θ the angular position of a particle seen from $\mathbf{c}_0$. Mixtures of eq. (11) and eq. (12) were just as poor. Clearly, not all multiscale updates are created equal.

Whilst we performed the most detailed simulations in two dimensions we did implement the three dimensional version of the code and used it to simulate a single system with $L = 40$. We performed both conventional and multiscale Monte Carlo and found, Figure 3, qualitatively similar results to Figure 2. It is to be noted that three dimensional simulations require considerably more computing resources for a given system size. The number of particles is larger, as well as the number of neighboring particles within 2.5σ.

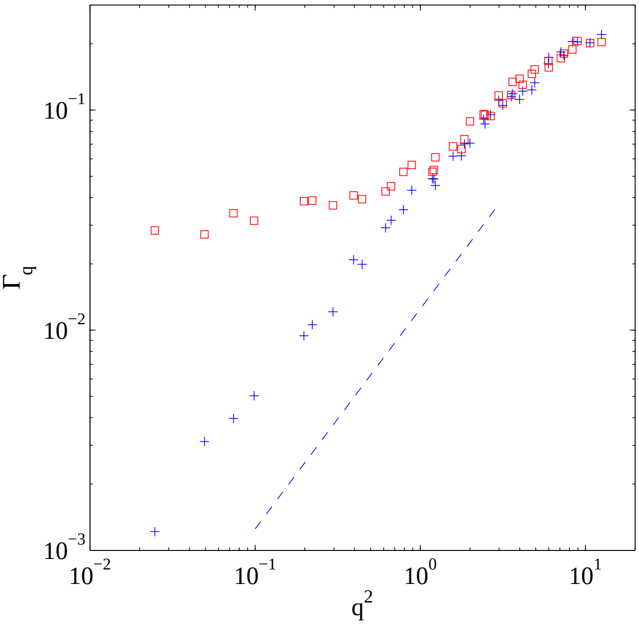

FIG. 3: Inverse integrated autocorrelation time in sweeps, Γ_q, for S_q in three dimensions. Conventional Monte Carlo: +. Multiscale: □. $T = 1$, $L = 40$, $l_c = 2.5$. Line $\Gamma \sim q^2$. Γ_q tends to a constant at long wavelengths for the multiscale algorithm.

In this Letter we have introduced a wider choice of Monte Carlo moves for the simulation of simple fluids. A good choice of updates leads to increases in the efficiency of simulations in a manner reminiscent of multigrid Monte Carlo simulation of lattice models [10]. Using conventional Monte Carlo we demonstrated that the relaxation rate varies as $\Gamma = Dq^2$; density fluctuations in large systems take $O(L^2)$ sweeps to equilibrate. Multiscale updates, with an average cost of $l_c^d \log L/l_c$ per update, eliminate long wavelength slowing down, leading to an algorithm which is asymptotically faster.

Our method permits more general implementations of preferential sampling [11] in which updates are performed more often in the neighborhood of an interesting site or surface. It may also have application in heterogeneous colloidal systems where co-motion of small and large particles or molecules may be an alternative to cluster methods at high densities.

[1] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration* (Springer-Verlag, 2002).
[2] A. Buhot and W. Krauth, Phys. Rev. Lett. **80**, 3787 (1998).
[3] J. Liu and E. Luijten, Phys. Rev. Lett. **92**, 035504 (2004).
[4] L. Dodd, T. Boone, and D. Theodorou, Mol. Phys. **78**, 961 (1993).
[5] B. Smit and D. Frenkel, J. Chem. Phys. **94**, 5663 (1991).
[6] J. G. Kirkwood, J. Chem. Phys. **7**, 911 (1939).
[7] P. C. Hohenberg and B. I. Halperin, Rev. Mod. Phys. **49**, 435 (1977).
[8] H. Flyvbjerg and H. G. Petersen, J. Chem. Phys. **91**, 461 (1989).
[9] S. K. Das, M. E. Fisher, J. V. Sengers, J. Horbach, and K. Binder, Phys. Rev. Lett. **97**, 025702 (pages 4) (2006).
[10] J. Goodman and A. D. Sokal, Phys. Rev. Lett. **56**, 1015 (1986).
[11] J. Owicki and H. Scheraga, Chem. Phys. Lett. **47**, 600 (1977).
[12] On repeating the argument with a block of rigidly displaced particles we find the exponent $d-2$ replaced by $d-1$. Large blocks can not move as far.
[13] We also checked this decay in two dimensions using Kawasaki dynamics for a lattice gas.
[14] We neglect internal friction of the fluid which leads to weaker dampening.