Molecular dynamics for nonequilibrium systems in which there are a small number of very hot particles in a cold bath: Reference system propagator methods

Ruhong Zhou, Steven J. Stuart, and B. J. Berne
Department of Chemistry, Columbia University, New York, New York 10027

(Received 17 January 1996; accepted 25 March 1996)

In molecular dynamics simulations of systems with very hot particles in contact with a bath of cold particles one must choose a very small time step dictated by the fast (hot) particles. This requires the recalculation of all the forces after each of these small time steps. We show how reference system propagator algorithms (RESPA) can be used to speed up these simulations, and also comment on some interesting physical phenomena involved in these nonequilibrium systems. © *1996 American Institute of Physics.* [S0021-9606(96)50125-X]

I. INTRODUCTION

In this paper we consider the problem of simulating a nonequilibrium condensed system in which one or several particles are initially very hot compared to all of the other particles in the system. These hot particles could be the result of radiation damage or highly exoergic chemical reactions.[1–3] The aim of the simulation would be to see how the hot particles perturb the many body system. These very hot particles produce local defects. In solids this can result in local melting. As the initial hot spot cools off, the remaining system heats up and after sufficient time the excess energy will equipartition leading to a new, higher ambient temperature.

The simulation of such systems poses certain interesting problems. Even though the very hot particles might contribute a very small fraction of the particles in the system they move so rapidly that the force exerted by them and on them by neighboring atoms will be a rapidly and strongly varying function of time. In order to avoid instabilities in the numerical integrator, it will be necessary to choose a very small time step for the integration even when the vast majority of the other atoms are moving slowly with respect to each other. Were there no hot particles a much larger integration time step could be used for the cold system. This problem has been addressed in the literature.[4] Here we show that a variant of r-RESPA[5–7] (reversible reference system propagator algorithm) can be used to treat this problem cleanly and can greatly increase the efficiency of such simulations.

II. THEORY

The reversible reference system propagator algorithm (r-RESPA)[5,6] is based upon the reversible Trotter factorization of the classical Liouville propagator. The Liouville operator L for a system of N degrees of freedom in Cartesian coordinates is defined as

$$iL=\{\dots,H\}=\sum_{i=1}^{N}\left[v_i\cdot\frac{\partial}{\partial r_i}+F_i\cdot\frac{\partial}{\partial p_i}\right], \tag{1}$$

where $\{...,H\}$ is the Poisson bracket with the Hamiltonian, H and (r_i,p_i) are the position and conjugate momentum for atom i. The state of the system at a time t, $\Gamma(t)$, is defined as the collective set of positions and conjugate momenta $(r(t),p(t))$. It evolves with time as

$$\Gamma(t)=e^{iLt}\Gamma(0), \tag{2}$$

where e^{iLt} is the classical time evolution propagator. Since the classical Liouville operator is self-adjoint, e^{iLt} is a unitary operator and the time evolution generated by the above equation is reversible. And, for any system whose Hamiltonian is an even function of the momenta, this also implies reversibility under momentum reflection.

For systems with two different time scales, we reversibly Trotter factorize[8] the propagator e^{iLt} into a reference system propagator with a smaller time-step δt and a correction propagator with a larger time step Δt. Decomposing the total Liouvillian as

$$iL=iL_h+iL_c, \tag{3}$$

where L_c and L_h denote the Liouvillians corresponding to the cold and hot particle subsystems, the Trotter factorization then becomes,

$$\begin{aligned}e^{iL\Delta t}&=e^{iL_c\Delta t/2}e^{iL_h\Delta t}e^{iL_c\Delta t/2}+O(\Delta t^3)\\&=e^{iL_c\Delta t/2}[e^{iL_h\delta t}]^n e^{iL_c\Delta t/2}+O(\Delta t^3).\end{aligned} \tag{4}$$

To partition the system into hot and cold subsystems, we need some criterion to distinguish the hot atoms from the cold ones. This is done by defining a velocity threshold. We define an instantaneous ''temperature'' of a particle such that $3kT/2=mv^2/2$ where $mv^2/2$ is the kinetic energy of the particle and we define a threshold temperature T^*. Then the hot subset will be all those particles which have, at a given time, a temperature greater than T^*, and the cold subset contains the remaining particles.

It is important to recognize that as the velocities change, particles will switch between the hot and cold subsets. We choose to update the definition of the hot and cold particles at the end of each large timestep, Δt. The effect of this choice of algorithm is discussed in more detail below.

Although the very hot particles might be only a small portion of the whole system, they move so rapidly that the force exerted by them and on them by neighboring atoms will be a rapidly and strongly varying function of time. Thus, an obvious and straightforward subdivision of the Liouville operator could be selected as follows:

$$iL_h=\sum_{j\in\mathrm{hot}}\left\{v_j\cdot\frac{\partial}{\partial r_j}+F_j\cdot\frac{\partial}{\partial p_j}\right\}\equiv v_h\cdot\frac{\partial}{\partial r_h}+F_h\cdot\frac{\partial}{\partial p_h}, \tag{5}$$

and

$$iL_c=\sum_{j\in\mathrm{cold}}\left\{v_j\cdot\frac{\partial}{\partial r_j}+F_j\cdot\frac{\partial}{\partial p_j}\right\}\equiv v_c\cdot\frac{\partial}{\partial r_c}+F_c\cdot\frac{\partial}{\partial p_c}. \tag{6}$$

Here, L_h only propagates the positions and momenta of atoms in the hot subset, and L_c propagates positions and momenta of atoms in the cold subset. However, we found that this subdivision gives poor energy conservation. The reason is that when we propagate the motion of the hot atoms, all the cold bath atoms are treated as ''frozen'' particles, which is not true in reality. The cold atoms that suffer collisions with the hot atoms need to be able to move quickly, since they have rapidly and strongly varying forces acting on them. This is different from the previous RESPA application to the disparate mass system,[6] where some particles have mass $m=1$ (reduced unit) while the others have mass $M=100$. The motion of the light particles can be separated from the heavy particles by a similar propagator subdivision, because the intrinsic time scales, which scale as $\sqrt{m}$, differ by a factor of 10 for the light and heavy particles. However, in the present case, the masses are the same for both hot and cold atoms, and it results a bad approximation. A more detailed analysis of this effect is presented elsewhere.[9]

The solution is to pull out the $v_c\cdot(\partial/\partial r_c)$ term from the L_c propagator and put it into the L_h propagator, instead. Furthermore, we need to differentiate the force terms more carefully to include the rapidly changing forces exerted by hot atoms in the L_h propagator. Then, a more reasonable and accurate subdivision of the Liouvillian will be:

$$iL_c=F_{cc}\cdot\frac{\partial}{\partial p_c} \tag{7}$$

and

$$\begin{aligned}iL_h&=v_h\cdot\frac{\partial}{\partial r_h}+v_c\cdot\frac{\partial}{\partial r_c}+(F_{hh}+F_{hc})\cdot\frac{\partial}{\partial p_h}+F_{ch}\cdot\frac{\partial}{\partial p_c}\\&\equiv v\cdot\frac{\partial}{\partial r}+F^*\frac{\partial}{\partial p}.\end{aligned} \tag{8}$$

Here F_{hh} denotes the force on a hot particle from all other hot particles, F_{ch} denotes the force on a hot particle due to all cold particles, F_{hc} denotes the force on a cold particle due to all hot particles, and F_{cc} denotes the force on a cold particle due to all other cold particles. The notation used here is a short hand for sums over the hot and cold particles. For example,

$$iL_c=\sum_{i\in\mathrm{cold}>j\in\mathrm{cold}}F_{ij}\cdot\frac{\partial}{\partial p_i}.$$

It should be noted that iL_h propagates both hot and cold particles under a force field with the forces between cold particles removed.

With this subdivision we can write the propagator as:

$$\begin{aligned}e^{iL\Delta t}&=e^{(\Delta t/2)F_{cc}(\partial/\partial p_c)}e^{\Delta t[v(\partial/\partial r)+F^*(\partial/\partial p)]}e^{(\Delta t/2)F_{cc}(\partial/\partial p_c)}\\&\quad+O(\Delta t^3).\end{aligned} \tag{9}$$

Breaking the hot factors into small time steps and further factorizing the ''hot'' propagator $e^{iL_h\delta t}$ then gives:

$$\begin{aligned}e^{iL\Delta t}&\approx e^{(\Delta t/2)F_{cc}\cdot(\partial/\partial p_c)}\\&\quad\times[e^{(\delta t/2)F^*\cdot(\partial/\partial p)}e^{\delta t v\cdot(\partial/\partial r)}e^{(\delta t/2)F^*\cdot(\partial/\partial p)}]^n\\&\quad\times e^{(\Delta t/2)F_{cc}\cdot(\partial/\partial p_c)}.\end{aligned} \tag{10}$$

These propagators instruct us to:

(i) First propagate the cold system by one half of a large time step Δt. Remember that the ''cold'' propagator involves only the cold–cold force part of the Liouvillian, and that this propagator affects only the momenta of the cold particles without changing their positions.

(ii) Next propagate the hot system by n small time steps $\delta t(n\,\delta t=\Delta t)$ using the velocity Verlet integrator. Remember that the ''hot'' propagator is really the whole system with the cold–cold forces, F_{cc}, turned off. In this step the positions and momenta of all the particles change.

(iii) Finally, propagate the ''cold'' system by one half of a large time step Δt as in step (i).

The hot propagation generated by $\exp(iL_h\delta t)$ includes the hot–cold forces and thus takes into account the fact that when a cold particle collides with a hot particle it experiences a very rapidly varying force, thus requiring a small time step. We note that when the system has equipartitioned such that there are no more hot particles the propagator becomes identical to the velocity Verlet propagator.[6,7]

This approach, based on the Trotter factorization of the Liouville propagator, is more efficient than the straightforward integration of the equations of motion using the small time step δt because the hot system involves the calculation of far fewer forces than the cold subsystem and the cold–cold forces are computed only every large time step instead of every small time step. The exact savings in CPU time will depend on the relative sizes of the hot and cold subsystems, and the difference in their associated time scales. Although we do not illustrate it here, as the system cools off one could make use of a smaller n and thereby increase the size of the small time step δt.

The algorithm, as described above and implemented in the simulations described below, is not time reversible. This is in stark contrast to all other published r-RESPA studies,[6,7,10,11] and is a consequence of the decision to update the hot/cold subsets only once per large timestep. It is possible to make the current algorithm reversible, by updating the hot/cold subsets within the smaller time steps δt

$=\Delta t/n$. Since these subsets can change as often as the velocities are modified, however, a total of $2n+2$ updates would have to be performed, incurring substantially more overhead than with the single update used here. Furthermore, a fairly convoluted algorithm is required in the reversible version of the method in order to avoid either storage or recomputation of the expensive $N_c \times N_c$ force array. In light of these observations, we opted for the simplicity and speed of the nonreversible algorithm. It needs to be mentioned, however, that this nonreversibility will induce an energy drift at long times. This drift, which is not observed to be a problem for the nonequilibrium processes reported here, can be eliminated through the use of a splined cutoff between hot and cold velocities, or with an explicitly reversible RESPA propagator. This and related r-RESPA subtleties will be discussed in a forthcoming paper.[9]

III. RESULTS AND DISCUSSION

To compare RESPA with the standard velocity Verlet integrator for nonequilibrated systems, we treat a Lennard-Jones (LJ) system with 4000 atoms at a reduced density $\hat{\rho}\equiv\rho\sigma^3=1.0$. Periodic boundary conditions are used in the simulation. A cutoff distance of $R_{\mathrm{cut}}=3.0\sigma$ is used for potential and force calculations and a neighbor list is set up for each atom for efficiency in these calculations.[12] All neighbors within a distance of $R_{\mathrm{neigh}}=3.5\sigma$ of an atom are included in the neighbor list of that atom. The neighbor list is updated[12] whenever the maximum displacement of a particle since the last update is larger than $(R_{\mathrm{neigh}}-R_{\mathrm{cut}})$.

The LJ system is first equilibrated to a reduced temperature of $T=0.1$, then the velocity of one particle is scaled to give it a very high ''temperature'' T_i. The very hot particle then participates in a cascade of collisions producing secondary, tertiary,...hot offspring. After a sufficient time, the excess kinetic energy of the very hot atom equipartitions with the cold bath, leading to a new higher equilibrated temperature. To have a clear picture of how the hot atoms (defined as those with temperature greater than a threshold $T^*=2.5$) are generated and annihilated during the equipartition process, we examine a system with an initial very hot particle with $T_i=1000$. This system equilibrates to a final temperature of $T=0.35$.

In Fig. 1 we plot the number of hot atoms in the hot subset as a function of time. A single time step of $\Delta t=0.001$ (reduced units) is used in the integration of Newton's equations of motion. The number of hot atoms $(T\geqslant T^*=2.5)$ increases in the beginning of the simulation due to the collisions of the very hot particle. At ~200 MD time steps, the number of the hot atoms reaches a maximum, ~50 atoms (or 1.2 % of the system) in this case. After approximately 1500 MD time steps, the number of hot atoms decreases gradually from the maximum to near zero because they equipartition with the large cold bath. It should be noted that the peak number of hot atoms, as well as the time taken for them to cool off, depends on the threshold T^* and the initial very hot particle's temperature T_i. In the following we will use two different approaches to address this nonequilibrium problem.

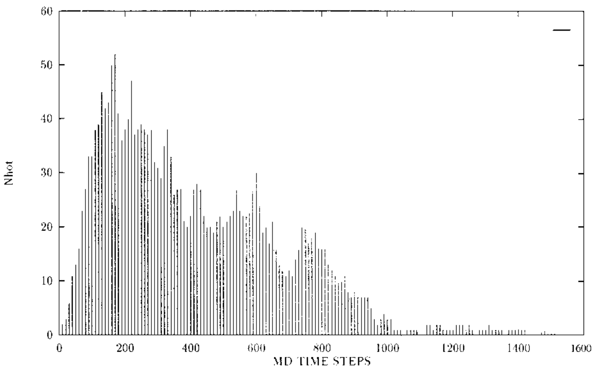

FIG. 1. Number of hot atoms *N*hot in the hot subset vs MD time steps. The temperature $T_i=1000$ for the initial very hot particle and the threshold $T^*=2.5$ for the hot subset are used in simulation.

To compare the efficiency of two different methods considered here, it is necessary to compare them at the same level of accuracy. In constant-energy MD simulations, the energy conservation parameter ΔE is commonly used to assess the accuracy of the simulation. This is defined as

$$\Delta E=\frac{1}{N_T}\sum_{i}^{N_T}\left|\frac{E_0-E_i}{E_0}\right|, \tag{11}$$

where E_i is the total energy at step i, E_0 is the initial total energy, and N_T is the total number of time steps. This quantity has been shown to be a reasonable measure of accuracy in previous simulations.[6,7] It has been found that a value of $\log \Delta E \leqslant -3.0$ gives rise to a stable MD integration.[6,7,13] In the following simulations, we use $\log \Delta E \sim -4.0$ as the accuracy criterion.

In Fig. 2 we show the performance with respect to energy conservation [defined in Eq. (11)] of the velocity Verlet and RESPA integrators when used to simulate the nonequilibrium relaxation of systems with one initial very hot atom having $T_i=1000$. The ΔE is evaluated over 1.0 reduced unit of time in the MD simulation. In the RESPA method, the inner loop, which propagates the hot subset, uses a small time step of $\delta t=0.001$; while the outer loop, the correction propagator, uses a large time step of $\Delta t=n\,\delta t$ $(n=1,2,3...,$ as

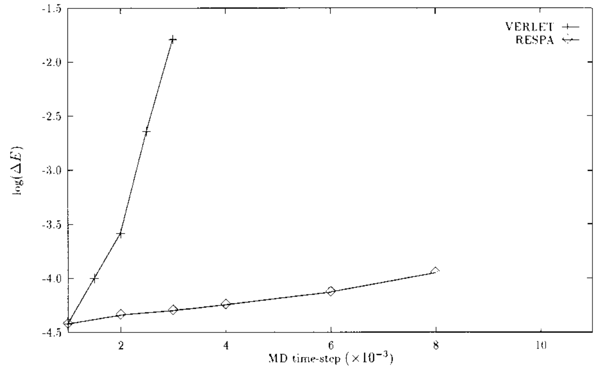

FIG. 2. Comparison of energy conservation as defined by Eq. (11) for velocity Verlet and RESPA vs the overall time step Δt used in integration of Newton's equation of motion.

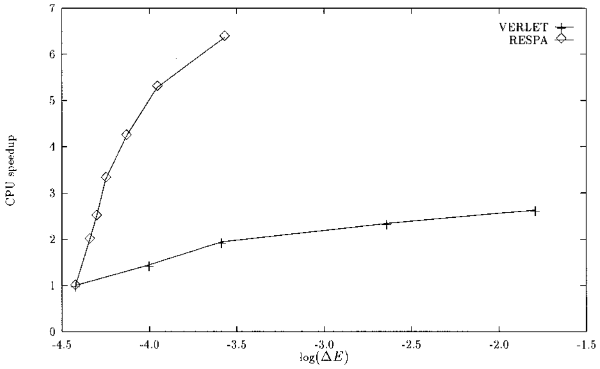

FIG. 3. Comparison of CPU time speed up vs energy conservation for velocity Verlet and RESPA. The CPU time of velocity Verlet with time step of $\Delta t=0.001$ is used as a benchmark. The figure shows that RESPA can save CPU time by a factor of ~4 with comparable accuracy.

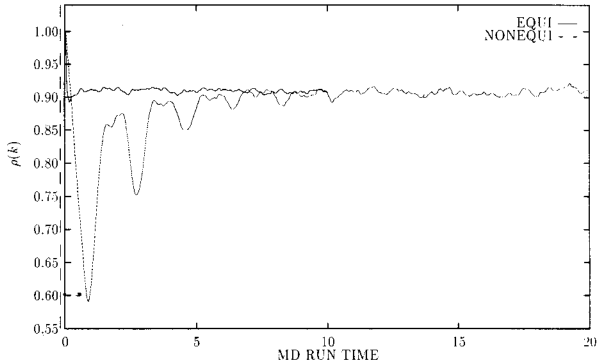

FIG. 4. Change of the structure factor $\rho_k(t)$ as defined by Eq. (12) vs the MD run time in the nonequilibrium process (dashed line). The structure factor of the conjugate equilibrated system with same final temperature $T=1.0$ is also plotted as comparison. The deep dips in the structure factor for the nonequilibrium system indicates the possible local melting or disorders.

described in Sec. II). The time step plotted in Fig. 2 for RESPA is the large time step, or the overall time step. The results show that for the velocity Verlet integrator, $\log(\Delta E)$ increases dramatically with the increase of time step. Even a small increase in the time step, Δt from 0.001 to 0.002, will induce a big error in the integration of the motion of the hot atoms. However, the RESPA method separates the ‘‘fast’’ motions from the ‘‘slow’’ motions systematically, so that large overall time step ($\Delta t \sim 0.008$–0.010) can be used with a satisfactory accuracy. For a particular level of energy conservation, $\log(\Delta E)=-4.0$, RESPA can use a time step ~6 times larger than that of the velocity Verlet.

The overhead associated with the smaller number of forces in the hot loop is small, but not zero. Thus, it is also useful to compare CPU times for the two methods. In this comparison, we used the CPU time for the velocity Verlet with time step of $\Delta t=0.001$ as a benchmark. The CPU time for a single step of RESPA with $\Delta t=0.001$ is about 1.04 times longer than the single step of the benchmark due to the bookkeeping costs of in updating the hot and cold subsets. Figure 3 shows the CPU time for $1000\Delta t$ vs the level of energy conservation $\log \Delta E$. To obtain an energy conservation level of $\log \Delta E \sim -4.0$, for example, we may either use velocity Verlet with time step $\Delta t=0.0015$, which gives a speedup of 1.4, or use RESPA with an overall time step $\Delta t=0.008$ (or $n=8$) which gives a speedup of 5.3. This means that RESPA is approximately a factor of four times faster than standard velocity Verlet for a comparable level of accuracy.

It is worth noting that when the hot particle is very hot, as it is, for example, in radiation damage, where a particle can receive 500 eV of kinetic energy,[4] or $T=5.8\times 10^6$ K (as comparison, $\epsilon/k=120$ K for argon), the Liouville propagator can be subdivided further corresponding to a ‘‘very hot’’ subset, ‘‘hot’’ subset and ‘‘cold’’ subset, thereby permitting the use of a multiplicity of time scales, with further CPU savings.

It is of interest to determine whether local melting is generated in a cold solid bath when a very hot atom is produced. The RESPA method is used to speed up the calculations. When a very hot atom with $T_i=3600$ is introduced into a cold bath at $T=0.1$ the final temperature of the system after equipartition is found to be $T=1.0$, leaving the fully equilibrated system in the crystalline phase.[14,15] The simulation starts with an fcc crystal structure. Two parameters are used to describe ordering and possibility of local melting. One is the structure factor $\rho_k(t)$, is defined by

$$\rho_k(t)=\frac{1}{N}\sum_{i=1}^{N}|e^{i\mathbf{k}\cdot\mathbf{r}_i(t)}|=\sum_{i=1}^{N}\cos(\mathbf{k}\cdot\mathbf{r}_i(t)), \tag{12}$$

where N is the number of particles ($N=4000$), $\mathbf{k}=(2\pi/l)(1,1,1)$ is the reciprocal wave vector of the fcc structure, $l=L/(N/4)^{1/3}$, and L is the length of the simulation box. For the unperturbed fcc crystal structure, $\rho_k=1.0$. If $\rho_k(t)$ stays close to 1.0 over time, the solid remains an fcc crystal; if it decays to zero, the solid melts.

Another parameter of interest is the atomic mean-square displacement $\langle \Delta r^2(t)\rangle$,

$$\langle \Delta r^2(t)\rangle=\left\langle \frac{1}{N}\sum_{i=1}^{N}(\mathbf{r}_i(t)-\mathbf{r}_i(0))^2\right\rangle, \tag{13}$$

where $\mathbf{r}_i(0)$ is the crystal lattice position of atom i at time zero, and $\mathbf{r}_i(t)$ is the position at time t. In a liquid, after an initial quadratic rise,

$$\langle \Delta r^2(t)\rangle \rightarrow 6Dt,$$

where D is the self diffusion coefficient. In a solid or metastable glass, $\langle \Delta r^2(t)\rangle$ rises to a constant value.

In Fig. 4 the evolution of the structure factor $\rho_k(t)$ with time is plotted for the nonequilibrium system. For comparison, $\rho_k(t)$ of the corresponding equilibrated system with the same final temperature $T=1.0$ (called the conjugate system) is also plotted in this figure. It can be seen that $\rho_k(t)$ for the conjugate system decreases from 1.0 to 0.9 quickly and stays at this equilibrated value with only very small fluctuations. The decrease of $\rho_k(t)$ from 1.0 to 0.9 is simply due to the thermal fluctuations of the atomic positions around their crystal lattice points. However, for the nonequilibrated sys-

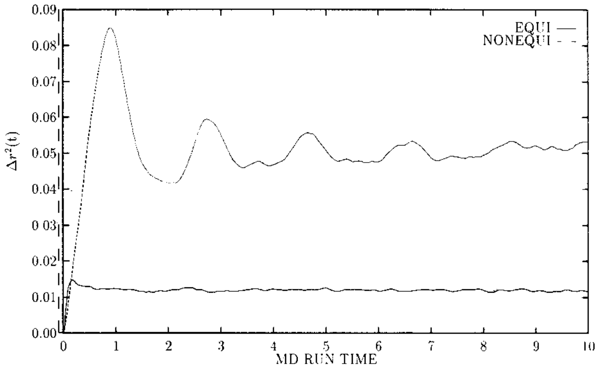

FIG. 5. Change of the diffusion displacement Δr^2 as defined by Eq. (13) vs the MD run time in the nonequilibrium process (dashed line). The diffusion displacement of the conjugate equilibrated system with same final temperature $T=1.0$ is also included as comparison (bold line). The difference in a sympletic displacement of the two systems indicates the switching of the atom positions and the possible local melting.

tem, large amplitude fluctuations resembling damped sinusoidal oscillations are observed in $\rho_k(t)$. We find that these oscillations correspond to the transit time of a sound or shock wave through the walls of the periodic box that leads to significant compression and rarification of the system. This is confirmed by a visual inspection. The $\rho_k(t)$ plot also shows that most of the disorder induced by the very hot atom recovers after a long equipartition process. The final $\rho_k(t)$ values of the equilibrated system and nonequilibrium system are almost the same, which indicates that the final structure of the nonequilibrium system is also an fcc crystal structure and not a structure with glassy regions. However, this does not rule out temporary dislocations and atom exchanges.

To see whether any atoms exchange positions during the relaxation around the hot atom, $\langle \Delta r^2(t) \rangle$ was calculated for both the nonequilibrium and the conjugate equilibrium systems. The results are shown in Fig. 5. The value of $\langle \Delta r^2(t) \rangle$ increases from zero to $0.014\sigma^2$ quickly for the conjugate system due to the thermal fluctuations (Debye Waller factor), while it increases to $0.085\sigma^2$ (~6 times larger) in 0.95 reduced time units for the nonequilibrated system and fluctuates several times [following the compressions and rarifications of the structure factor $\rho_k(t)$]. However, unlike the structure factor $\rho_k(t)$ which is the same for both systems after equipartition of the nonequilibrium system, the final $\langle \Delta r^2 \rangle$ for the nonequilibrated system is still approximately three times larger than in the conjugate equilibrium system, or $0.041\sigma^2$. This is consistent with the exchange of atoms during the equilibration phase of the system. Using the simple approximation that atoms exchange positions only with their nearest neighbors, we can estimate that the number of atoms which have switched lattice positions is ~130. Movies of the relaxation confirm that there are a significant number of particle exchanges during equilibration. Thus we observe that local melting does occur during equilibration. After this relaxation regime, the high kinetic energy of the very hot atom equilibrates, and the system relaxes to an fcc structure.

IV. SUMMARY

The reference system propagator algorithm (RESPA) has been shown to be more efficient than standard Verlet methods in simulating the kinds of nonequilibrium systems discussed here. For a 4000 atom LJ solid with one very hot atom ($T_i=1000$), RESPA can use a time step ~6 times larger than velocity Verlet for an MD simulation of the nonequilibrium process, which speeds up the simulation by a factor of ~4.

Even larger speedups in CPU time could be achieved in certain other situations: For example:

(1) In radiation damage very much hotter particles are generated than we simulate here. In this case one can define different time scales for very hot particles, hot particles, somewhat hot particles etc. In this way, more subsets are used in the subdivision of the Liouville propagator. This technique is simple to implement in an MD program: Whenever a hot spot is detected in the system, the program calls the RESPA subroutine, which subdivides the Liouville propagator as appropriate for efficiency. When the system reaches a more homogeneous distribution of velocities, the integrator would revert back to velocity Verlet. This can be useful in simulations of chemically reacting systems in which the reactions are highly exoergic. In this way one can dynamically change the time steps used as the system relaxes.

(2) In previous work,[6] it has been shown that, even for such simple force fields as the Lennard-Jones force field, using r-RESPA with the forces subdivided into short and long range parts defined by a switching function, leads to speedups as large as a factor of five over standard integrators. This type of force breakup can also be introduced into the current simulation for even larger speedups. Systems with long-ranged interactions will show even greater savings in CPU time by further breaking the forces into different regions of pair distances.

[1] *Proc. Int. Conf. On Computer Simulation of Radiation Effects in Solids, Hahn-Meitner Institut, Berlin, Aug. 23–28, 1992, Radiation Effects and Defects in Solids*, edited by J. P. Biersack (Gordon and Breach, New York, 1994), Vols. 130–131.

[2] T. Diaz de la Rubia and M. W. Guinan, Phys. Rev. Lett. **66**, 2766 (1991).

[3] D. W. Brenner, D. H. Robertson, M. L. Elert, and C. T. White, Phys. Rev. Lett. **70**, 2174 (1993).

[4] E. Glikman, I. Kelson, N. V. Doan, and H. Tietze, preprint (1995).

[5] M. E. Tuckerman, G. J. Martyna, and B. J. Berne, J. Chem. Phys. **94**, 6811 (1991).

[6] M. E. Tuckerman, B. J. Berne, and G. J. Martyna, J. Chem. Phys. **97**, 1990 (1992).

[7] R. Zhou and B. J. Berne, J. Chem. Phys. **103**, 9444 (1995).

[8] H. F. Trotter, Proc. Am. Math Soc. **10**, 545 (1959).

[9] S. J. Stuart, R. Zhou, and B. J. Berne, J. Chem. Phys. (in press).

[10] P. Procacci and B. J. Berne, J. Chem. Phys. **101**, 2421 (1994).

[11] D. D. Humphreys, R. A. Friesner, and B. J. Berne, J. Phys. Chem. **98**, 6885 (1994).

[12] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University, Oxford, 1987).

[13] M. Watanabe, and M. Karplus, J. Chem. Phys. **99**, 8063 (1993).

[14] J. P. Hansen and I. R. McDonald, *Theory of Simple Liquids* (Academic, New York, 1976).

[15] Y. Choi and T. Ree, J. Chem. Phys. **12**, 9917 (1993).