Molecular dynamics algorithm for multiple time scales: Systems with disparate masses

Mark E. Tuckerman[a] and Bruce J. Berne
Department of Chemistry, Columbia University, New York, New York 10027

Angelo Rossi
IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598

(Received 30 July 1990; accepted 1 October 1990)

A frequently encountered problem in molecular dynamics is how to treat the long times that are required to simulate condensed systems consisting of mixtures of light and heavy particles. Standard methods require the choice of time step sufficiently small to guarantee stable solution for the low mass component with the consequence that these simulations require a very large number of central processing unit cycles to treat the relaxation of the heavier component. In this note, we present a new method that allows one to use a time step appropriate for the heavy particles. This method uses a similar idea to numerical analytical propogator algorithm, an algorithm we invented to treat high frequency oscillators interacting with low frequency baths and is based on a choice of a reference system for the light particle motions. The method is applied to the case of a liquid containing 864 Lennard-Jones spheres, 824 of these particles having a mass, $M = 100$ and 40 spheres picked at random have a mass $m = 1$. It is shown that molecular dynamics using the new algorithm runs seven to ten times faster than standard methods and this approach as well as suitable generalizations should be very useful for future simulations of quantum and classical condensed matter systems.

I. INTRODUCTION

Consider a system consisting of a mixture of light (mass $= m$) and heavy spheres (mass $= M$). In such systems, there is a disparity in the molecular dynamic time scales. If one wishes to simulate such systems using the standard integrators of molecular dynamics, then the maximum time step that can be used to integrate the equations of motion must be chosen to insure accurate integration of the low mass component with the consequence that a very small time step is needed. When a large disparity in time scales exists, a very large number of central processing unit (CPU) cycles will be required to allow the slow degrees of freedom to fluctuate enough to obtain converged time averages for the whole system.

In this paper, a method for accelerating the simulation of such systems is presented. This method, called RESPA (reference system propogator algorithm), is a variant of the numerical analytical propogator algorithm (NAPA), algorithm that we invented for treating the problem of high frequency oscillators coupled to low frequency oscillators.[1] The RESPA method is based on numerical solutions of the reference system equations. The gist of the method is to define a dynamical reference system for the fast motion and to derive equations of motion for the deviation $\delta(t)$ of the fast coordinates from the reference system coordinates. These deviations are coupled to the equations of motion of the slow coordinates. The fast dynamical system is integrated for n small time steps $n\delta t$ holding the slow coordinates fixed. The time dependence of the reference system is then fed into the coupled equations for $\delta(t)$ and the slow coordinates and the resulting equations are integrated for one large time step $\Delta t = n\delta t$. The initial conditions for each large time step are then chosen so that this deviation $\delta(t)$ is zero with the consequence that the deviation is always kept small. The only approximation in this algorithm springs from the numerical integrator used to integrate the equations of motion of the reference system and the coupled equations. Otherwise, the method is self-correcting and exact. For simplicity, the reference system is taken to be the Hamiltonian of the original system with the slow coordinates held fixed at their values at the beginning of the time step.

Teleman and Jönsson[2] have proposed a multiple time-step (TJMTS) method in which the forces are separated into slow and fast components. This separation yields a set of coupled equations of motion for the slow and fast degrees of freedom. TJMTS uses a small step δt to advance the fast degrees of freedom n steps holding the slow variables fixed. The slow degrees of freedom are then updated using a time step $\Delta t = n\delta t$. This method does not correct for the errors incurred in the approximate factorization of the equations of motion, a fact which shows up in poor energy conservation. This is well illustrated when we compare the results of RESPA, TJMTS, and velocity Verlet[3] (using a small time step). Swindoll and Haile[4] have proposed a more accurate multiple time-step method than Teleman and Jönsson, but their method requires high-order spatial derivatives of the potential and is therefore more computationally intensive than RESPA.

For simplicity, we apply this new method to the simulation of a mixture of Lennard-Jones spheres consisting of 824 heavy spheres of mass $M = 100$ and 40 light spheres of mass $m = 1$. For Lennard-Jones (LJ) spheres, the two time scales are $\Delta t_f = \sqrt{m\sigma_1^2/\epsilon_1}$ and $\Delta t_s = \sqrt{M\sigma_2^2/\epsilon_2}$, where σ_i and ϵ_i are the Lennard-Jones parameters for component i. For il-

[a] Ph. D. student in the Department of Physics, Columbia University.

lustrative purposes, we take $\sigma_1 = \sigma_2$ and $\epsilon_1 = \epsilon_2$. The fluid is simulated at the triple point ($\rho\sigma^3 = 0.86$ and $\hat{T} = 0.67$). For this system, the fast motion is ten times faster than the slow motion. Of course if the σ's or ϵ's are different, one can produce even greater disparities in time scale. We compare the CPU times required to generate a given amount of self-diffusion and a given accuracy for the following algorithms:

(i) straightforward molecular dynamics (MD) using the velocity Verlet integrator with a time step chosen to produce the given accuracy;

(ii) MD using RESPA with time step adjusted to give the given accuracy;

(iii) the TJ (Ref. 2) multiple time-step method (MTS) using identical time steps as in RESPA.

We find that straightforward velocity Verlet integration takes seven times longer than RESPA to generate the same mean-square displacement for the given accuracy for the system defined above. For this same system, TJMTS compared favorably in time, but is three orders of magnitude less accurate in energy conservation.The poor energy conservation in the TJMTS calculation is due to a systematic drift in the energy. In implementing TJMTS, one must rescale the velocities periodically in order to avoid this, although this will not yield true dynamics. It is worth noting that for a fixed number of light particles, the acceleration increases with the number of host heavy particles.

The RESPA and MTS methods may also be based on the Gear predictor–corrector algorithm. In fact, numerous molecular dynamics packages (e.g., the MUMOD program of Teleman and Jönsson[2]) employ this integrator as the method of choice. The use of second-order algorithm is, of course, equivalent to velocity Verlet. When a higher-order predictor–corrector procedure is used in the presence of a large mass disparity, then it becomes necessary to iterate the corrector step in order to insure accurate integration in the higher-order time derivatives. For example, if a third-order predictor–corrector algorithm is used, we find that three iterations of the corrector step are necessary to yield acceptable results. Significant degeneration of accuracy occurs when the number of iterations is decreased, a fact which should be considered when using packages which integrate by the Gear algorithm.

With this method, there are many problems that are now feasible. Diffusion of light particles in polymer solutions, solid state diffusion of light impurity atoms are just two kinds of problems that can be approached in this way. Clearly RESPA can be combined with NAPA to treat problems involving light molecules with stiff vibrations. In addition, it may be possible to treat the time-scale disparity due to a very low moment of inertia with concomitant fast rotations and slow translations in water by similar methods.

II. METHOD

To illustrate the method, consider a set of differential equations of the form

$$\ddot{x} = \frac{1}{m} F_x(x,y),$$
$$\ddot{y} = \frac{1}{M} F_y(x,y), \tag{2.1}$$

where $m \ll M$, x and y represent, respectively, the positions of the light and heavy particles, and F_x and F_y represent the forces of these fast and slow degrees of freedom. If F_x and F_y are of the same order, x will change much more rapidly than y.

We must solve this set of equations subject to the initial conditions $\{x(0),\dot{x}(0),y(0),\dot{y}(0)\}$. To proceed, we define a "reference system" equation of motion

$$\ddot{x}_0 = \frac{1}{m} F_x(x_0,\bar{y}) \tag{2.2}$$

which must be solved subject to the initial conditions

$$x_0(0) = x(0),$$
$$\dot{x}_0(0) = \dot{x}(0), \tag{2.3}$$

while keeping the position of the slow coordinate fixed at some value $\bar{y}$ [e.g., $\bar{y} = y(0)$ the value at the beginning of the interval]. The solution of the reference system equations of motion (2.2) denoted

$$x_0(t) = x_0[t;x(0),\dot{x}_0(0),\bar{y})]. \tag{2.4}$$

The true position of the light system deviates from the reference system position and can be expressed as

$$x(t) = x_0(t) + \delta(t). \tag{2.5}$$

Substitution of this into Eq. (2.1) and elimination of $\ddot{x}_0$ using Eq. (2.2) then results in the set of equations

$$\ddot{\delta} = \frac{1}{m}[F_x(x_0 + \delta,y) - F_x(x_0,\bar{y})],$$
$$\ddot{y} = \frac{1}{M} F_y(x_0 + \delta,y). \tag{2.6}$$

To solve Eq. (2.6) subject to the initial conditions [which follow from Eq. (2.3)]

$$\delta(0) = 0,$$
$$\dot{\delta}(0) = 0, \tag{2.7}$$

we propose the following scheme:

(1) numerically integrate Eq. (2.2) for a sequence of n small time steps $\delta t(\Delta t = n\delta t)$ with the background of heavy particles fixed at $\bar{y}$, thus generating $x_0(t)$ for $0 \leqslant t \leqslant \Delta t$ [the choice of $\bar{y} = y(0)$ the value at the beginning of the interval lends itself easily to integration by the velocity Verlet algorithm];

(2) introduce the numerical solution for $x_0(t)$ into Eq. (2.6);

(3) solve Eq. (2.6) subject to the initial conditions [Eq. (2.7)] for one time step Δt using a suitable integrator to obtain $\delta(\Delta t)$, $\dot{\delta}(\Delta t)$, $y(\Delta t)$, and $\dot{y}(\Delta t)$;

(4) calculate

$$x(\Delta t) = x_0(\Delta t) + \delta(\Delta t),$$
$$\dot{x}(\Delta t) = \dot{x}_0(\Delta t) + \dot{\delta}(\Delta t). \tag{2.8}$$

This process is repeated using $x(\Delta t)$, $\dot{x}(\Delta t)$, $y(\Delta t)$, and $\dot{y}(\Delta t)$ as initial conditions. In general, at each step, the output is used as the initial conditions for the next step.The advantage of this method lies in the resetting of the initial conditions on $\delta(t)$ and $\dot{\delta}(t)$ to 0 at every step. Since δ and $\dot{\delta}$ never deviate much from 0 in the given step, the force term $F_x(x_0 + \delta,y) - F_x(x_0,\bar{y})$ in Eq. (2.6) is prevented from be-

coming too large, thus allowing the use of a larger time step in the numerical integration than could be used in the standard integration schemes. It is worth noting that, in contrast to Eqs. (2.2) and (2.6), the TJMTS method[2] uses the approximate set of equations

$$\ddot{x}_0 = \frac{1}{m} F_x(x_0, \bar{y}),$$
$$\ddot{y} = \frac{1}{M} F_y(x_0, y) \tag{2.9}$$

and is thus not self-correcting.

The integration of Eq. (2.6) can be done by any integrator suitable for equations with explicit time dependence.[5] The Runge–Kutta (or predictor–corrector) method are such integrators. In particular, the velocity Verlet[3,5] integrator can be adapted straightforwardly for Eq. (2.6).

The major reason that RESPA saves so much time is that during a small time step δt Eq. (2.2) is integrated keeping the background of heavy particles constant. Thus it is not necessary to recompute the forces between the heavy particles. These have to be updated only once every large time step Δt when Eq. (2.6) is integrated. If there are many more heavy particles than light particles, this saving in force computations can be dramatic. For a mixture containing N_1 light particles and N_2 heavy particles interacting with central forces, one often uses spherical bookkeeping methods. Let z_{ij} denote the average number of particles of type j that interact with a particle of type i. The number of forces that must be calculated during the n time steps δt, during which the slow particles kept fixed are then $nN_1(z_{12} + z_{11}/2)$ and the number that must be calculated for the integration of Eq. (2.6) is $N_2(z_{21} + z_{22}/2)$. Thus the total CPU time for one large time step using RESPA, assuming that the calculation of the forces is the time-consuming part of the calculation, is proportional to $[nN_1(z_{12} + z_{11}/2) + N_2(z_{21} + z_{22}/2)]$. On the other hand, if the calculation is done by standard integrators to compute n steps of length δt moving all of the particles during each cycle, the total number of forces that must be calculated in $n[N_1(z_{12} + z_{11}/2) + N_2(z_{21} + z_{22}/2)]$. This leads to the prediction that the ratio of CPU times for the usual methods to that of RESPA will be

$$r = \frac{n[N_1(z_{12} + z_{11}/2) + N_2(z_{21} + z_{22}/2)]}{[nN_1(z_{12} + z_{11}/2) + N_2(z_{21} + z_{22}/2)]}. \tag{2.10}$$

Equation (2.10) allows us to predict a number of things. We see that

$$\lim_{N_2/N_1 \to \infty} r = n. \tag{2.11}$$

Thus RESPA will be at best n times faster than straightforward methods. For the liquid simulated in this paper $n = 10$, but for fluids in which the light mass particle has a much smaller σ, n can be much larger than 10. The optimal limit $r = n$ can be achieved only for infinitely dilute solutions where the solute is light. For finite concentrations, $r \leqslant n$ and the gain will be smaller, but still nonnegligible. For the case considered here, we predict $r \approx 7$. Furthermore, for this case, r decreases from its theoretical upper limit as the mole fraction of the light species increases at fixed system volume. It reaches a theoretical minimum of $r = 1$, which is exactly equivalent to the standard algorithms. Finally, since the z_{ij}'s all vary in approximately the same way with density, we expect that r will be relatively insensitive to density variations.

III. RESULTS

For simplicity, we apply this new method to the simulation of a mixture of Lennard-Jones spheres consisting of 824 heavy spheres of mass $M = 100$ and 40 light spheres of mass $m = 1$. For Lennard-Jones (LJ) spheres, the two time scales are $\Delta t_f = \sqrt{m\sigma_1^2/\epsilon_1}$ and $\Delta t_s = \sqrt{M\sigma_2^2/\epsilon_2}$, where σ_i and ϵ_i are the Lennard-Jones parameters for component i. For illustrative purposes, we take $\sigma_1 = \sigma_2$ and $\epsilon_1 = \epsilon_2$. The fluid is simulated at the triple point ($\rho\sigma^3 = 0.86$ and $\hat{T} = 0.67$). The time steps used are $\delta t = 2 \times 10^{-3}$ and $\Delta t = 2 \times 10^{-2}$ in reduced units $[(M\sigma^2/\epsilon)^{1/2}]$. To simulate the same real times 50 K time steps were used in the Verlet simulation and 5 K time steps were used in RESPA and TJMTS simulations after equilibration.

To compare the accuracy and effectiveness of the different methods, we compute the following quantities:

(1) The unnormalized velocity autocorrelation function $C_v(t)$ of the light particles defined by

$$C_v(t) = \left\langle \frac{1}{N_m} \sum_{i=1}^{N_m} \mathbf{v}_i(0) \cdot \mathbf{v}_i(t) \right\rangle, \tag{3.1}$$

where N_m is the number of light particles and the sum is taken over all the light particles in the system.

(2) The mean-square displacement (MSD) $\Delta r^2(t)$ defined by

$$\Delta r^2(t) = \left\langle \frac{1}{N_m} \sum_{i=1}^{N_m} [\mathbf{r}_i(t) - \mathbf{r}_i(0)]^2 \right\rangle. \tag{3.2}$$

(3) The average deviation of the energy from its initial value defined to be

$$\Delta \hat{E} = \frac{1}{T} \int_0^T dt \left| \frac{E(t) - E(0)}{E(0)} \right|, \tag{3.3}$$

where T is the total time of the run.

In Fig. 1(a), the velocity autocorrelation functions for the three integration methods are plotted. From the plot, it can be seen that while the velocity correlation functions using RESPA and the velocity Verlet integrators agree, the one calculated using TJMTS differs dramatically from these. Note the initial value. By definition, $C_v(0) = 3kT/m$, a value which is well reproduced by RESPA and velocity Verlet, but which is in error in the TJMTS scheme. There are also significant differences in the short time behavior. In Fig. 2, we plot the mean-square displacements, the inset showing the short time behavior. Interestingly, the errors in $C_v(t)$ from TJMTS cancel when the function is integrated over time and thus do not show up in the long time behavior of the msds. The self-diffusion coefficient of the light particles determined from the slopes of the long time limit of the msds are $D_{\text{Verlet}} = 2.0007 \times 10^{-2}$, $D_{\text{RESPA}} = 1.997 \times 10^{-2}$, and $D_{\text{TJMTS}} = 2.003 \times 10^{-2}$. These coefficients are expressed in dimensionless units and must be multiplied by the factor $\sigma\epsilon/M$ to express them in cm/s^2. The agreement among the three methods is remarkable good. Also shown in Fig. 2 is

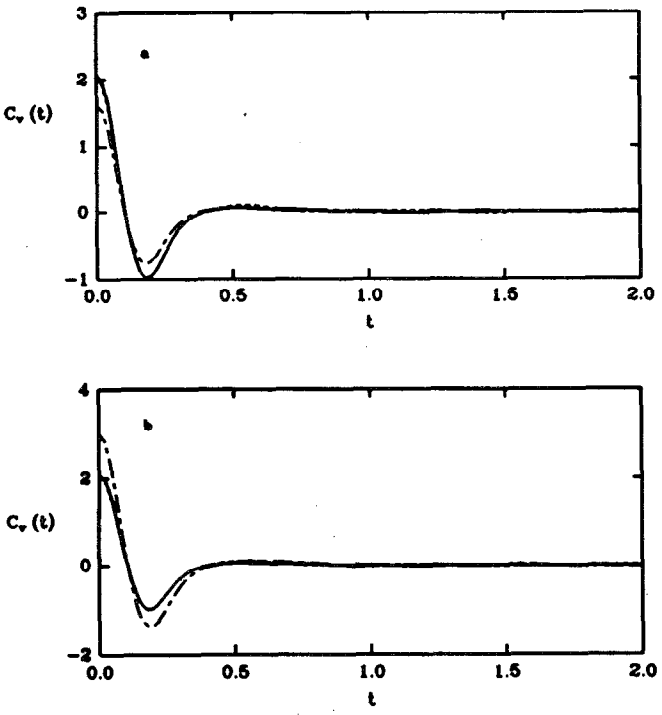

FIG. 1. (a) The unnormalized velocity correlation functions defined in Eq. (3.1) as a function of time for the light particles in the Lennard-Jones mixture for the light particles for RESPA (solid line), velocity Verlet (dashed line), and for TJMTS (dash–dot line). (b) The same as (a), but using a different initial configuration.

the mean-square displacement of the light particles in a fixed configuration of the heavy particles, which demonstrates that in the fixed configuration, the light particles get trapped. The MSD of the light particles in this fixed matrix agrees with the full simulations using RESPA, etc., for short times, but, as expected, diffusion of the light particles can take place only if the heavy particles can move to allow them to get out of their traps. Energy conservation measured by Eq. (3.3) for the runs described above was $\Delta\hat{E}_{\mathrm{RESPA}} = 2 \times 10^{-6}$, $\Delta\hat{E}_{\mathrm{Verlet}} = 2 \times 10^{-6}$, and $\Delta\hat{E}_{\mathrm{TJMTS}} = 2 \times 10^{-3}$, which demonstrates that TJMTS is significantly less accurate with respect to energy conservation than RESPA.

It is interesting to compare the velocity autocorrelation functions and mean-square displacements above with those for the heavy particles. The unnormalized correlation function for the heavy particles computed by RESPA and TJMTS is shown in Fig. 3(a) and the corresponding MSD in Fig. 3(b). The differences between RESPA and TJMTS in the light particle dynamics do not show up for the heavy particles. The reason for this is the RESPA and TJMTS treat the heavy particle dynamics in the same way and the light particles do not significantly influence the dynamics of the heavy particles. In particular, there is substantially more efficient backscattering of a light particle from its cage of heavy particles than a heavy particle from a cage of light particles. This can be seen by noting that the light particle velocity autocorrelation function has a much deeper negative region than the heavy particle one. The diffusion coefficients calculated from the long time dependence of the MSD for RESPA and TJMTS also agree, giving a value of $D = 1.72 \times 10^{-2}$. The diffusion coefficients for light and heavy particles are quite similar (a result expected from the Stokes–Einstein relation $D = kT/2\pi\eta\sigma$). Note also that the decay time of the heavy particle correlation function is essentially ten times longer than for the light particles in accordance with the hundredfold mass disparity.

It should be mentioned that the TJMTS results for the light particles are extremely sensitive to initial conditions.

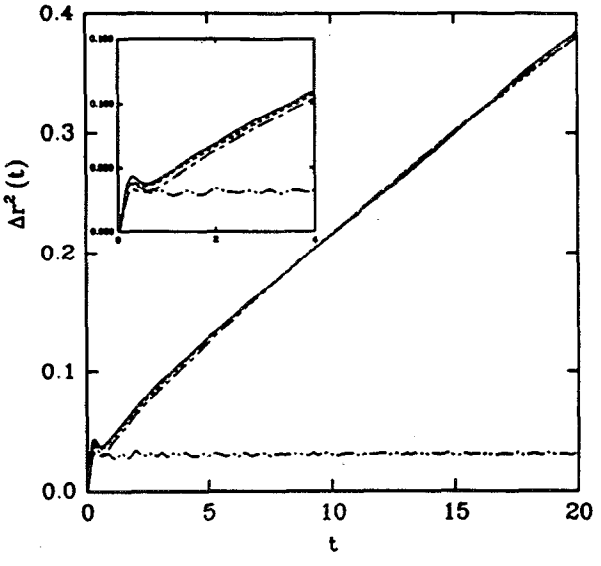

FIG. 2. The mean-square displacents defined in Eq. (3.2) as a function of time for the light particles in the Lennard-Jones mixture for the light particles for RESPA (solid line), velocity Verlet (dashed line), TJMTS (dash–dot line), and for motion of the light particles in a fixed heavy particle matrix with the same initial conditions as in Fig. 1(a) (dash–double dot line).

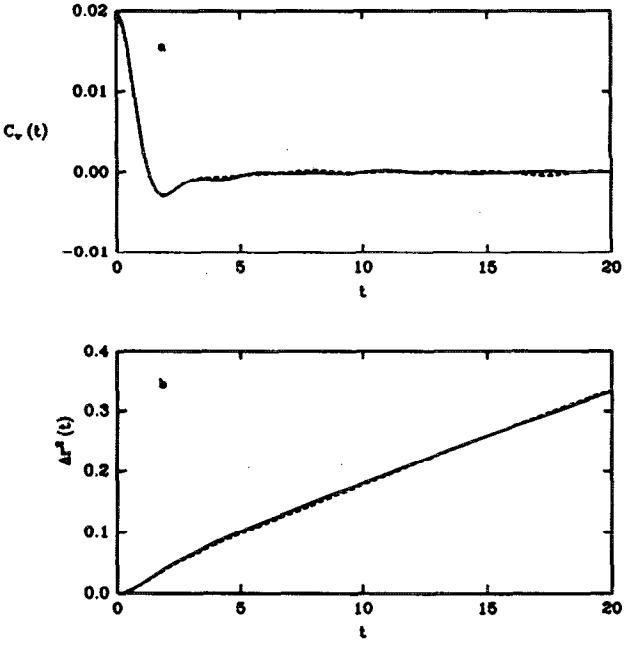

FIG. 3. (a) The velocity autocorrelation function for the heavy particles as computed by RESPA (solid line) and by TJMTS (dashed line). (b) The mean square displacement for the heavy particles as computed by RESPA (solid line) and TJMTS (dashed line).

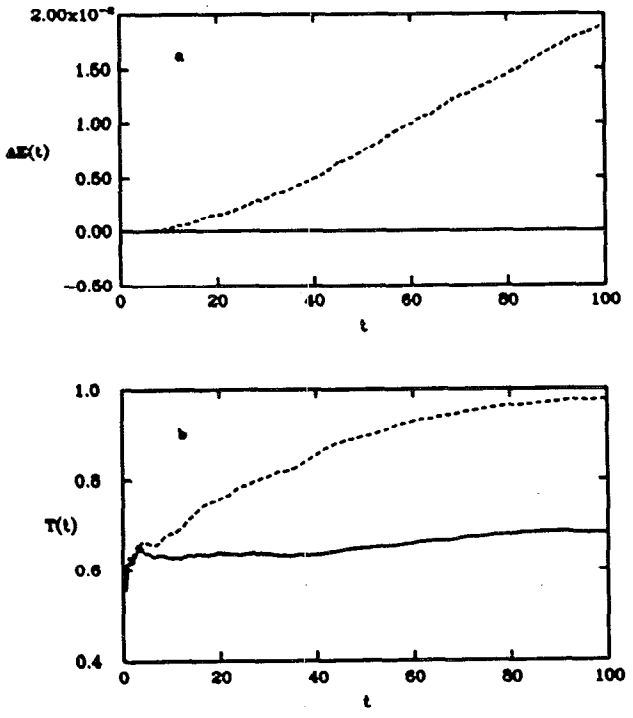

FIG. 4. (a) The instantaneous deviation in the energy from the initial value, normalized by the initial value for RESPA (solid line) and TJMTS (dashed line). (b) The cumulative average temperature of the light particles for RESPA (solid line) and TJMTS (dashed line).

To see this, we carried out an additional set of simulations using a different initial configuration. We plot the resulting velocity autocorrelation functions from these simulations in Fig. 1(b). This time, we see that TJMTS gives an initial value which is too large. In addition, energy conservation measured by Eq. (3.3) for these runs yielded $\Delta\hat{E}_{\mathrm{RESPA}} = 1.5\times10^{-6}$, $\hat{E}_{\mathrm{Verlet}} = 1.6\times10^{-6}$, and $\Delta\hat{E}_{\mathrm{TJMTS}} = 8\times10^{-3}$. The large value of $\Delta\hat{E}_{\mathrm{TJMTS}}$ is due to a systematic drift in the energy during run. To see this, we plot $\Delta E(t) \equiv [E(t) - E(0)]/E(0)$ vs t for RESPA and TJMTS in Fig. 4(a). Here we see that TJMTS has a significant drift in the energy. In Fig. 4(b), we plot $T(t)$, the cumulative average of the temperature of the light particles corresponding to the results in Fig. 4(a). Since there are only 40 light particles, one expects large fluctuations about the equilibrium temperature as can be seen in the RESPA curve, however, the cumulative average does converge to the correct value. The TJMTS curve does not converge, but drifts to a value too large as expected from the initial value of the velocity autocorrelation function. Because dynamic quantities were sought, TJMTS was done with no velocity rescaling.

IV. CONCLUSION

In this paper we have presented a method which accelerates molecular dynamics simulations of systems consisting of light and heavy particles. This method, called RESPA, an extension of the NAPA method is based on a set of exact equations which greatly reduces the number of force calculations that are required for the simulation. Since it is the computation of the forces that dominates the CPU time required to simulate systems, this reduction in the number of pair forces that have to be evaluated leads to a saving in CPU time. The ratio of the time required for the simulation using the RESPA method to that for a standard method is given by r in Eq. (2.10). For solutions very dilute in the light particles, $r = n$, where n is the number of time steps for which the fast coordinates are integrated keeping the slow coordinates fixed in the reference system. For the Lennard-Jones mixture in which there is a light particle in a large bath of heavy particles $n = \sqrt{M/m}$, a number that can be large. The method presented here is much more accurate than the TJMTS method which is based on an approximate set of equations and leads to poor energy conservation.

It is a simple matter to generalize the RESPA method to treat the problem of simulating systems with both long and short range forces. This will be presented in a forthcoming paper.

[1] M. Tuckerman, G. Martyna, and B. J. Berne, J. Chem. Phys. **93**, 1287 (1990).
[2] Olle Teleman and Bo Jönsson, J. Comp. Chem. **7**, 58 (1986).
[3] W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, J. Chem. Phys. **76**, 637 (1982).
[4] R. D. Swindoll and J. M. Haile, J. Comp. Phys. **53**, 289 (1984).
[5] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University, Oxford, 1989).