MOLECULAR PHYSICS, 1996, VOL. 87, NO. 5, 1117–1157

Explicit reversible integrators for extended systems dynamics

By GLENN J. MARTYNA
Department of Chemistry, Indiana University, Bloomington, IN 47405-4001, USA

MARK E. TUCKERMAN, DOUGLAS J. TOBIAS and MICHAEL L. KLEIN
Department of Chemistry, University of Pennsylvania, Philadelphia, PA 19104-6323, USA

(*Received 4 May 1995; revised version accepted 4 September 1995*)

Explicit reversible integrators, suitable for use in large-scale computer simulations, are derived for extended systems generating the canonical and isothermal–isobaric ensembles. The new methods are compared with the standard implicit (iterative) integrators on some illustrative example problems. In addition, modification of the proposed algorithms for multiple time step integration is outlined.

1. Introduction

Recently, explicit reversible multiple time step integrators have been developed to handle efficiently problems involving stiff vibrations, disparate masses, and long-range forces that occur in large-scale molecular dynamics (MD) calculations [1]. The methods are based on the Liouville operator formulation of Hamiltonian mechanics. However, it is often useful to perform dynamical calculations using so-called extended system schemes that generate statistical mechanical ensembles other than the traditional microcanonical ensemble. For example, it is possible to generate both the canonical and the isothermal–isobaric ensembles via continuous dynamics [2–7]. Although non-Hamiltonian in nature, the equations of motion possess the fundamental symmetry of time reversibility. Fortunately, the Liouville operator formalism can be extended to treat these more complicated non-Hamiltonian cases.

In this article, explicit reversible integrators for systems yielding the canonical [6] and isobaric–isothermal ensembles [7] are developed. The resulting algorithms are straightforward to use and can be modified readily to accommodate the type of reversible multiple time step approaches found to be effective in microcanonical simulations [1]. In addition, since reversible integrators are required by the hybrid Monte Carlo algorithm/MD scheme [8], the new algorithms allow extended phase space methods to be used in conjunction with this technique. The reversible integrators, and their multiple time step generalizations, are tested on a variety of model and 'real' systems and the results compared with the standard implicit (iterative) integration schemes based on the usual velocity–Verlet scheme.

2. Equations of motion

The equations of motion that generate the canonical and isobaric–isothermal ensembles are first reviewed briefly.

2.1. *Canonical ensemble*

Martyna, Tuckerman and Klein (MTK), following Nosé [3] and Hoover [4], employed a chain of Nosé–Hoover thermostats to drive a dynamical system to generate canonically distributed positions $\mathbf{r}_i$, and momenta $\mathbf{p}_i$. The equations of motion MTK proposed are as follows [6]:

$$\begin{aligned}
\dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i}, \\
\dot{\mathbf{p}}_i &= \mathbf{F}_i - \mathbf{p}_i \frac{p_{\xi_i}}{Q_i}, \\
\dot{\xi}_i &= \frac{p_{\xi_i}}{Q_i}, \\
\dot{p}_{\xi_1} &= \left[\sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{m_i} - N_f kT\right] - p_{\xi_1} \frac{p_{\xi_2}}{Q_2}, \\
\dot{p}_{\xi_j} &= \left[\frac{p_{\xi_{j-1}}^2}{Q_{j-1}} - kT\right] - p_{\xi_j} \frac{p_{\xi_{j+1}}}{Q_{j+1}}, \\
\dot{p}_{\xi_M} &= \left[\frac{p_{\xi_{M-1}}^2}{Q_{M-1}} - kT\right].
\end{aligned} \tag{1}$$

There are M thermostats ξ_j, with masses Q_j, and momenta p_{ξ_j}. Here, N is the number of particles, N_f is the number of degrees of freedom (equal to dN, where d is the system dimension, if there are no constraints), and $\mathbf{F}_i = -\nabla_i \phi(\mathbf{r})$ is the force derived from the potential ϕ, which is a function of the N-particle position vector $\mathbf{r}$.

The conserved quantity for the MTK dynamics is

$$H' = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + \sum_{i=1}^{M} \frac{p_{\xi_i}^2}{2Q_i} + \phi(\mathbf{r}) + N_f kT\xi_1 + kT \sum_{i=2}^{M} \xi_i, \tag{2}$$

and the Jacobian [9] is

$$\begin{aligned}
\frac{\mathrm{d}J(t)}{\mathrm{d}t} &= -J(t)\left[\sum_{i=1}^{M}\left(\frac{\mathrm{d}\dot{\xi}_i}{\mathrm{d}\xi_i} + \frac{\mathrm{d}\dot{p}_{\xi_i}}{\mathrm{d}p_{\xi_i}}\right) + \sum_{i=1}^{N}\left(\nabla_{\mathbf{p}_i}\dot{\mathbf{p}}_i + \nabla_{\mathbf{r}_i}\dot{\mathbf{r}}_i\right)\right], \\
J(t) &= \exp\left[N_f \xi_1 + \sum_{i=2}^{M} \xi_i\right].
\end{aligned} \tag{3}$$

The Jacobian is the weight associated with the phase space volume and is unity for systems that obey Liouville's theorem [9]. It represents the transformation to a set of variables $\{s_1 = N_f s_1 p_{\xi_1}, \dot{s}_i = s_i p_{\xi_i}\}$ or, equivalently, $\{\log s_1 = N_f \xi_1, \log s_i = \xi_i\}$ for which the Jacobian $J = 1$. If the system studied is ergodic, the dynamics gives rise to the canonical (NVT) ensemble, i.e., the partition function is proportional to

$$\int \mathrm{d}\mathbf{p}_\xi \,\mathrm{d}\xi \,\mathrm{d}\mathbf{p} \,\mathrm{d}\mathbf{r} \exp\left[N_f \xi_1 + \sum_{i=2}^{M} \xi_i\right] \delta(E - H') \propto \int \mathrm{d}\mathbf{p}_\xi \,\mathrm{d}\mathbf{p} \,\mathrm{d}\mathbf{r} \exp\left[-\beta H''\right], \tag{4}$$

where

$$H'' = \sum_{i=1}^{N} \frac{\mathbf{p}_i^2}{2m_i} + \sum_{i=1}^{M} \frac{p_{\xi_i}^2}{2Q_i} + \phi(\mathbf{r}). \tag{5}$$

2.2. *Isothermal–isobaric ensemble (isotropic cell fluctuations)*

In order to produce the isothermal–isobaric ensemble, the volume V of the simulation cell must be permitted to undergo only isotropic fluctuations. Following Anderson [2] and Hoover [4], the equations of motion are [7]:

$$\begin{aligned}
\dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \frac{p_\varepsilon}{W}\mathbf{r}_i, \\
\dot{\mathbf{p}}_i &= \mathbf{F}_i - \left(1 + \frac{d}{N_f}\right)\frac{p_\varepsilon}{W}\mathbf{p}_i - \frac{p_\xi}{Q}\mathbf{p}_i, \\
\dot{V} &= \frac{dVp_\varepsilon}{W}, \\
\dot{p}_\varepsilon &= dV(P_{\text{int}} - P_{\text{ext}}) + \frac{d}{N_f}\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i} - \frac{p_\xi}{Q}p_\varepsilon, \\
\dot{\xi} &= \frac{p_\xi}{Q}, \\
\dot{p}_\xi &= \sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i} + \frac{p_\varepsilon^2}{W} - (N_f + 1)kT,
\end{aligned} \tag{6}$$

where P_{ext} is the external/applied pressure,

$$P_{\text{int}} = \frac{1}{dV}\left[\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i} + \sum_{i=1}^{N}\mathbf{r}_i \cdot \mathbf{F}_i - (dV)\frac{\partial\phi(\mathbf{r}, V)}{\partial V}\right] \tag{7}$$

is the internal pressure, p_ε is the momentum associated with the logarithm of the volume, and W is the mass of the 'barostat'. Note that, in general, the particles and barostat can be coupled to separate chains of thermostats although, for simplicity, they have been coupled to the same single thermostat here.

The conserved quantity is

$$H' = \sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{2m_i} + \frac{p_\varepsilon^2}{2W} + \frac{p_\xi^2}{2Q} + \phi(\mathbf{r}, V) + (N_f + 1)kT\xi + P_{\text{ext}}V, \tag{8}$$

and the Jacobian of the dynamics is

$$J = \exp[(N_f + 1)\xi]. \tag{9}$$

The isothermal–isobaric partition function,

$$\Delta \propto \int \mathrm{d}p_\xi \, \mathrm{d}p_\varepsilon \, \mathrm{d}V \int_{D(V)} \mathrm{d}\mathbf{p} \, \mathrm{d}\mathbf{r} \exp\left(-\frac{H''}{kT}\right), \tag{10}$$

is therefore produced, where

$$H'' = \sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{2m_i} + \frac{p_\varepsilon^2}{2W} + \frac{p_\xi^2}{2Q} + \phi(\mathbf{r}, V) + P_{\text{ext}}V, \tag{11}$$

and $D(V)$ is the domain defined by the volume [7].

2.3. *Isothermal–isobaric ensemble (full cell fluctuations)*

If full fluctuations of the simulation cell size and shape are permitted then, following Parrinello and Rahman [5], the isothermal–isobaric (NPT) ensemble with partition function

$$\Delta \propto \int d\vec{\mathbf{h}} \exp[-\beta P_{\text{ext}} \det[\vec{\mathbf{h}}]] Q(\vec{\mathbf{h}}) \det[\vec{\mathbf{h}}]^{1-d} \tag{12}$$

is generated (to within a constant) by the following equations of motion [7]:

$$\begin{aligned}
\dot{\mathbf{r}}_i &= \frac{\mathbf{p}_i}{m_i} + \frac{\vec{\mathbf{p}}_g}{W_g}\mathbf{r}_i, \\
\dot{\mathbf{p}}_i &= \mathbf{F}_i - \frac{\vec{\mathbf{p}}_g}{W_g}\mathbf{p}_i - \left(\frac{1}{N_f}\right)\frac{\mathrm{Tr}[\vec{\mathbf{p}}_g]}{W_g}\mathbf{p}_i - \frac{p_\xi}{Q}\mathbf{p}_i, \\
\dot{\vec{\mathbf{h}}} &= \frac{\vec{\mathbf{p}}_g \vec{\mathbf{h}}}{W_g}, \\
\dot{\vec{\mathbf{p}}}_g &= V(\vec{\mathbf{P}}_{\text{int}} - \vec{\mathbf{I}} P_{\text{ext}}) + \left[\frac{1}{N_f}\sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i}\right]\vec{\mathbf{I}} - \frac{p_\xi}{Q}\vec{\mathbf{p}}_g, \\
\dot{\xi} &= \frac{p_\xi}{Q}, \\
\dot{p}_\xi &= \sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{m_i} + \frac{1}{W_g}\mathrm{Tr}[\vec{\mathbf{p}}_g^{\,t}\vec{\mathbf{p}}_g] - (N_f + d^2)kT,
\end{aligned} \tag{13}$$

where $\vec{\mathbf{h}}$ is the cell matrix ($V = \det[\vec{\mathbf{h}}]$), $\vec{\mathbf{I}}$ is the identity matrix, $\mathrm{Tr}[\vec{\mathbf{p}}_g^{\,t}\vec{\mathbf{p}}_g]$ is the sum of the squares of all the elements of $\vec{\mathbf{p}}_g$, the cell variable momentum matrix, and the particles and cell variables have been coupled to a single thermostat. The pressure tensor is defined as

$$\begin{aligned}
(P_{\text{int}})_{\alpha\beta} &= \frac{1}{V}\left[\sum_{i=1}^{N}\frac{(\mathbf{p}_i)_\alpha(\mathbf{p}_i)_\beta}{m_i} + (\mathbf{F}_i)_\alpha(\mathbf{r}_i)_\beta - (\vec{\phi}'\vec{\mathbf{h}}^{\,t})_{\alpha\beta}\right], \\
(\phi')_{\alpha\beta} &= \frac{\partial\phi(\mathbf{r}, \vec{\mathbf{h}})}{\partial(h)_{\alpha\beta}}.
\end{aligned} \tag{14}$$

Equations (13) have the conserved quantity

$$H' = \sum_{i=1}^{N}\frac{\mathbf{p}_i^2}{2m_i} + \frac{1}{2W_g}\mathrm{Tr}[\vec{\mathbf{p}}_g^{\,t}\vec{\mathbf{p}}_g] + \frac{p_\xi^2}{2Q} + \phi(\mathbf{r}, \vec{\mathbf{h}}) + P_{\text{ext}}\det[\vec{\mathbf{h}}] + (N_f + d^2)kT\xi, \tag{15}$$

and Jacobian

$$J = \det[\vec{\mathbf{h}}]^{1-d}\exp[(N_f + d^2)\xi], \tag{16}$$

which lead to the desired NPT ensemble [7].

2.4. *Elimination of cell rotations*

The equations of motion for the case of a flexible simulation cell were derived using the full matrix $\vec{\mathbf{h}}$ of Cartesian cell parameters. The cell can therefore, in general, rotate in space [10]. This motion can make data analysis difficult and should be eliminated.

The origin of the rotational motion of the cell lies in the pressure tensor $P_{\alpha\beta}$ (see

equation (14)). If the pressure tensor is asymmetric ($P_{\alpha\beta} \neq P_{\beta\alpha}$) at a given instant of time, then there will be a net torque acting on the cell that will cause it to rotate. Cell rotations can be eliminated by using the symmetrized tensor $\bar{P}_{\alpha\beta} = (P_{\alpha\beta} + P_{\beta\alpha})/2$ in the equations of motion and setting the initial total angular momentum of the cell to zero ($\vec{\mathbf{p}}_g = \vec{\mathbf{p}}_g^{\,t}$) [7]. This approach is formally implemented by constraining $\vec{\mathbf{g}} = \vec{\mathbf{g}}^{\,t}$, where $\vec{\mathbf{p}}_g = W\vec{\mathbf{g}}$.

2.5. *Choice of mass for the extended variables*

It is generally useful to thermostat the particles and the barostats independently using Nosé–Hoover chains. It has been shown elsewhere [3, 6] that the masses of the particle thermostats should be

$$Q_{\mathrm{p}_1} = N_{\mathrm{f}} kT/\omega_{\mathrm{p}}^2,$$
$$Q_{\mathrm{p}_i} = kT/\omega_{\mathrm{p}}^2,$$

where ω_{p} is the frequency at which the particle thermostats fluctuate. Similarly, the masses of the barostat/cell parameter thermostats should be:

$$Q_{\mathrm{b}_1} = d^2 kT/\omega_{\mathrm{b}}^2,$$
$$Q_{\mathrm{b}_i} = kT/\omega_{\mathrm{b}}^2.$$

Finally, the masses of the barostat/cell parameters themselves [3, 11] should be

$$W = (N_{\mathrm{f}} + d)\, kT/\omega_{\mathrm{b}}^2,$$
$$W_{\mathrm{g}} = (N_{\mathrm{f}} + d)\, kT/d\omega_{\mathrm{b}}^2.$$

3. Iterative velocity–Verlet based integrators

The standard velocity–Verlet integration [12] is based on the equations

$$x(\Delta t) = x(0) + \dot{x}(0)\,\Delta t + \ddot{x}(0)\frac{\Delta t^2}{2} + \mathcal{O}(\Delta t^3),$$

$$\dot{x}(\Delta t) = \dot{x}(0) + [\ddot{x}(0) + \ddot{x}(\Delta t)]\frac{\Delta t}{2} + \mathcal{O}(\Delta t^3). \tag{17}$$

If the second time derivative directly depends on the first derivative, then $\dot{x}(\Delta t)$ must, in general, be determined iteratively, thus sacrificing reversibility and forming an implicit method. Iterative velocity–Verlet integration of the equations of motion presented in the previous section has been described elsewhere [7].

4. Explicit reversible integrators

The purpose of this paper is to develop schemes for integrating the equations of motion presented in section 2 with explicit reversible integrators derived from the appropriate classical propagators using operator factorization techniques [13, 1]. This approach yields manifestly reversible integrators that weight the phase space correctly [13, 1]. We mention again that the phase space volume is conserved only for systems that obey Liouville's theorem, and the equations of motion given in section 2 do not necessarily have this property. Explicit reversible integrators are now presented that integrate the equations of motion for the different ensembles in order of increasing

complexity: *NVE*, *NVT*, *NPT* with isotropic simulation cell fluctuations, and *NPT* with full simulation cell fluctuations. It will be shown that the new integrators can be modified easily to accommodate reversible multiple time step methods useful for studying systems involving stiff vibrations, long-range forces, and other problems involving a separation of time scales [1].

4.1. *The evolution operator*

A system of coupled, first order differential equations can be evolved from time $t = 0$ to time t by applying the evolution operator

$$\begin{aligned} \Gamma(t) &= \exp(iLt)\,\Gamma(0), \\ iL &= \dot{\Gamma}\cdot\nabla_{\Gamma}, \end{aligned} \tag{18}$$

where iL is the Liouville operator and Γ is the multidimensional vector of independent variables (coordinates and velocities). In general, the action of the evolution operator on the coordinates cannot be performed analytically. Therefore, a short-time approximation to the true operator, accurate at time $\Delta t = t/P$, is applied P times in succession to evolve the system:

$$\Gamma(t) = \prod_{i=1}^{P}\left(\prod_{s}\exp(iL_s\Delta t)\right)\Gamma(0), \tag{19}$$

which, for an nth order factorization, carries an overall error t^n/P^{n-1}.

4.2. *Microcanonical ensemble (NVE)*

In the *NVE* ensemble, the dynamics is Hamiltonian, and the Liouville operator is written as

$$iL = \sum_{i=1}^{N}\mathbf{v}_i\cdot\nabla_{\mathbf{r}_i} + \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i}. \tag{20}$$

A short-time approximation to the evolution operator can be generated via the Trotter formula [14]

$$\exp(iL\Delta t) = \exp\left(iL_1\frac{\Delta t}{2}\right)\exp(iL_2\Delta t)\exp\left(iL_1\frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^3), \tag{21}$$

with $iL_1 = \sum_{i=1}^{N}[\mathbf{F}_i(\mathbf{r})/m_i]\cdot\nabla_{\mathbf{v}_i}$ and $iL_2 = \sum_{i=1}^{N}\mathbf{v}_i\cdot\nabla_{\mathbf{r}_i}$.

As outlined earlier, the approximate evolution operator can be used to generate the positions and velocities at time Δt:

$$\begin{aligned} \mathbf{r}(\Delta t) &= \mathbf{r}(0) + \Delta t\mathbf{v}(0) + \frac{\Delta t^2}{2m}\mathbf{F}[\mathbf{r}(0)], \\ \mathbf{v}(\Delta t) &= \mathbf{v}(0) + \frac{\Delta t}{2m}(\mathbf{F}[\mathbf{r}(0)] + \mathbf{F}[\mathbf{r}(\Delta t)]), \end{aligned} \tag{22}$$

where the identity $\exp[a(\partial/\partial g(x))]\,x = g^{-1}[g(x)+a]$ has been used. The result is just the familiar velocity–Verlet integration scheme (section 3) [12], derived in an unfamiliar but powerful way [13, 1].

Another feature of the operator formalism is the ease with which a product of operators in a factorized expression such as equation (21) can be translated directly into computer code. The operators can be applied sequentially, thus bypassing the need to calculate analytically the phase space vector $\Gamma(\Delta t)$ in terms of the initial conditions $\Gamma(0)$. This will be referred to as the direct translation technique. For example, the three operators that appear in equation (21) can be translated into three sets of instructions which update first the velocities, next the positions and finally the velocities. A pseudocode for performing these operations would appear as:

$$\begin{aligned} \mathbf{v} &= \mathbf{v} + \frac{\Delta t}{2m} * \mathbf{F} \\ \mathbf{r} &= \mathbf{r} + \Delta t * \mathbf{v} \\ &\text{CALL} \quad \text{FORCE} \\ \mathbf{v} &= \mathbf{v} + \frac{\Delta t}{2m} * \mathbf{F} \end{aligned} \tag{23}$$

where loops over the number of atoms are assumed. Here, the closed form expression for $\Gamma(\Delta t)$ is simple (cf. equation (22)). In the more complex cases discussed below, such expressions rapidly become cumbersome, and the direct translation technique is the preferable approach.

4.3. *Canonical (NVT) ensemble*

An integration scheme for the *NVT* ensemble can be formulated also using the approach described above. The Liouville operator for the equations of motion, equation (1), is

$$\begin{aligned} iL = &\sum_{i=1}^{N} \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} + \sum_{i=1}^{N} \left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i} \right] \cdot \nabla_{\mathbf{v}_i} \\ &- \sum_{i=1}^{N} v_{\xi_1} \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} + \sum_{i=1}^{M} v_{\xi_i} \frac{\partial}{\partial \xi_i} + \sum_{i=1}^{M-1} (G_i - v_{\xi_i} v_{\xi_{i+1}}) \frac{\partial}{\partial v_{\xi_i}} + G_M \frac{\partial}{\partial v_{\xi_M}}, \end{aligned} \tag{24}$$

where a chain of Nosé–Hoover thermostats of length M has been coupled to an N particle system [6] and

$$\begin{aligned} G_1 &= \frac{1}{Q_1} \left(\sum_{i=1}^{N} m_i \mathbf{v}_i^2 - N_f kT \right) \\ G_i &= \frac{1}{Q_i} (Q_{i-1} v_{\xi_{i-1}}^2 - kT) \quad i > 1. \end{aligned} \tag{25}$$

Using a simple generalization of the Trotter formula, the evolution operator can be written as

$$\begin{aligned} \exp(iL\Delta t) = &\exp\left(iL_{\mathrm{NHC}} \frac{\Delta t}{2} \right) \exp\left(iL_1 \frac{\Delta t}{2} \right) \exp(iL_2 \Delta t) \exp\left(iL_1 \frac{\Delta t}{2} \right) \exp\left(iL_{\mathrm{NHC}} \frac{\Delta t}{2} \right) \\ &+ \mathcal{O}(\Delta t^3), \end{aligned} \tag{26}$$

where L_1 and L_2 retain their previous definitions, and $iL_{\mathrm{NHC}} = iL - iL_1 - iL_2$ consists of the rest of terms in iL. The Nosé–Hoover chain (NHC) part of the evolution operator, $\exp(iL_{\mathrm{NHC}}\Delta t/2)$, is simplified as follows:

$$\exp\left(iL_{\mathrm{NHC}}\frac{\Delta t}{2}\right) = \prod_{i=1}^{n_c}\exp\left(iL_{\mathrm{NHC}}\frac{\Delta t}{2n_c}\right) \tag{27}$$

$$\begin{aligned}\exp\left(iL_{\mathrm{NHC}}\frac{\Delta t}{2n_c}\right) = {} & \exp\left(\frac{\Delta t}{4n_c}G_M\frac{\partial}{\partial v_{\xi_M}}\right)\exp\left(-\frac{\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\\ & \times\exp\left(\frac{\Delta t}{4n_c}G_{M-1}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\exp\left(-\frac{\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\\ & \times\exp\left(-\frac{\Delta t}{2n_c}\sum_{i=1}^{N}v_{\xi_1}\mathbf{v}_i\cdot\nabla\mathbf{v}_i\right)\exp\left(\frac{\Delta t}{2n_c}\sum_{i=1}^{M}v_{\xi_i}\frac{\partial}{\partial\xi_i}\right)\\ & \times\ldots\exp\left(-\frac{\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\exp\left(\frac{\Delta t}{4n_c}G_{M-1}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\\ & \times\exp\left(-\frac{\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\exp\left(\frac{\Delta t}{4n_c}G_M\frac{\partial}{\partial v_{\xi_M}}\right),\end{aligned} \tag{28}$$

where a multiple time step, $n_c > 1$, approach has been used. For typical simulations, n_c can be taken to be one. However, if the frequency associated with the Nosé–Hoover chain is high ($Q = NkT/\omega^2$), n_c must be taken rather large to generate accurate trajectories. Substantial reduction can be obtained by using a higher order algorithm to apply the NHC part of the evolution operator, e.g.,

$$\exp\left(iL_{\mathrm{NHC}}\frac{\Delta t}{2}\right) = \prod_{i=1}^{n_c}\left[\prod_{j=1}^{n_{ys}}\exp\left(iL_{\mathrm{NHC}}\frac{w_j\Delta t}{2n_c}\right)\right], \tag{29}$$

where the w_j are chosen such that when $\exp(iL_{\mathrm{NHC}}\Delta t/2n_c)$ is approximated as in equation (28), the error is $\mathcal{O}(\Delta t/2n_c)^5$ [15, 16]. The values of the $\{n_{ys}, w_j\}$ are $\{n_{ys} = 3, w_1 = w_3 = 1/(2-2^{1/3}), w_2 = 1-2w_1\}$ [15] or, alternatively, $\{n_{ys} = 5, w_1 = w_2 = w_4 = w_5 = 1/(4-4^{1/3}), w_3 = 1-4w_1\}$ [16]. Higher order methods involving more time steps are also possible [15, 16].

In the multiple time step procedure (equation (27)), the seemingly unnecessary factorization

$$\begin{aligned}&\exp\left\{t'[-v_{\xi_k}v_{\xi_{k-1}} + G_{k-1}]\frac{\partial}{\partial v_{\xi_{k-1}}}\right\}\\ &\quad\sim\exp\left(-\frac{t'}{2}v_{\xi_k}v_{\xi_{k-1}}\frac{\partial}{\partial v_{\xi_{k-1}}}\right)\exp\left(t'G_{k-1}\frac{\partial}{\partial v_{\xi_{k-1}}}\right)\exp\left(-\frac{t'}{2}v_{\xi_k}v_{\xi_{k-1}}\frac{\partial}{\partial v_{\xi_{k-1}}}\right)\end{aligned} \tag{30}$$

has been employed. The action of the unfactorized operator on $v_{\xi_{k-1}}$ gives

$$v_{\xi_{k-1}} \to v_{\xi_{k-1}}\exp(-v_{\xi_k}t') + t'G_{k-1}\exp\left(-v_{\xi_k}\frac{t'}{2}\right)\left[\frac{\sinh\left(\frac{t'}{2}v_{\xi_k}\right)}{\frac{t'}{2}v_{\xi_k}}\right], \tag{31}$$

while the factorized operator yields

$$v_{\xi_{k-1}} \to v_{\xi_{k-1}} \exp(-v_{\xi_k} t') + t' G_{k-1} \exp\left(-v_{\xi_k} \frac{t'}{2}\right). \tag{32}$$

The potentially singular hyberbolic sine function that appears in the unfactorized result, equation (31), can be expanded in a Maclaurin series to arbitrarily high order without loss of generality. This is equivalent to building progressively higher order Trotter formula solutions [17] to the operator. In fact, no degradation of accuracy was found to result from the use of the factorized version (i.e., the first term in the Maclaurin series expansion of the hyperbolic sine).

The approach described above appears to be rather complicated. However, it is actually straightforward and computationally inexpensive to implement. One first applies the operator $\exp(iL_{\mathrm{NHC}}\Delta t/2)$ to update the $\{\xi, \mathbf{v}_\xi, \mathbf{v}\}$ (the operator alters only these variables). Next, one uses the updated velocities as input to the usual velocity–Verlet step, equation (22), and then applies $\exp(iL_{\mathrm{NHC}}\Delta t/2)$ to the output of this velocity–Verlet step. The procedure can be summarized as follows:

$$\begin{aligned}
\mathbf{r}(\Delta t) &= \mathbf{r}_{vV}\left[\Delta t; \mathbf{r}(0), \mathbf{v}\left(\frac{\Delta t}{2}\right)\right] \\
\mathbf{v}(\Delta t) &= \mathbf{v}_{\mathrm{NHC}}\left[\frac{\Delta t}{2}; \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right)\right] \\
\xi(\Delta t) &= \xi_{\mathrm{NHC}}\left[\frac{\Delta t}{2}; \xi\left(\frac{\Delta t}{2}\right), \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right)\right] \\
\mathbf{v}_\xi(\Delta t) &= \mathbf{v}_{\xi_{\mathrm{NHC}}}\left[\frac{\Delta t}{2}; \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right)\right],
\end{aligned} \tag{33}$$

where

$$\begin{aligned}
\mathbf{v}\left(\frac{\Delta t}{2}\right) &= \mathbf{v}_{\mathrm{NHC}}\left[\frac{\Delta t}{2}; \mathbf{v}(0), \mathbf{v}_\xi(0)\right] \\
\mathbf{v}'(\Delta t) &= \mathbf{v}_{vV}\left[\Delta t; \mathbf{r}(0), \mathbf{v}\left(\frac{\Delta t}{2}\right)\right] \\
\xi\left(\frac{\Delta t}{2}\right) &= \xi_{\mathrm{NHC}}\left[\frac{\Delta t}{2}; \xi(0), \mathbf{v}(0), \mathbf{v}_\xi(0)\right] \\
\mathbf{v}_\xi\left(\frac{\Delta t}{2}\right) &= \mathbf{v}_{\xi_{\mathrm{NHC}}}\left[\frac{\Delta t}{2}; \mathbf{v}(0), \mathbf{v}_\xi(0)\right],
\end{aligned} \tag{34}$$

and $\{\mathbf{r}_{vV}(t'; \mathbf{r}', \mathbf{v}'), \mathbf{v}_{vV}(t'; \mathbf{r}', \mathbf{v}')\}$ represents the output of a single velocity Verlet step of time increment t' using the initial conditions $\{\mathbf{r}', \mathbf{v}'\}$. A similar structure is used to indicate the action of the NHC part of the evolution operator.

The direct translation technique can be used to convert the operator $\exp(iL_{\mathrm{NHC}} w_j \Delta t/2n_c)$ into a set of instructions analogous to equation (27):

CALL ATOMIC_KE(AKIN)

SCALE = 1.0

DO $k = 1, n_c$

DO $j = 1, n_{ys}$

$\Delta t_s = w_j \Delta t / n_c$

$G_M = (Q_{M-1} v_{\xi_{M-1}}^2 - kT)/Q_M$

$v_{\xi_M} = v_{\xi_M} + \frac{\Delta t_s}{4} * G_M$

$v_{\xi_{M-1}} = v_{\xi_{M-1}} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_M}\right)$

$G_{M-1} = (Q_{M-2} v_{\xi_{M-2}}^2 - kT)/Q_{M-1}$

$v_{\xi_{M-1}} = v_{\xi_{M-1}} + \frac{\Delta t_s}{4} * G_{M-1}$

$v_{\xi_{M-1}} = v_{\xi_{M-1}} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_M}\right)$

...

$v_{\xi_1} = v_{\xi_1} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_2}\right)$

$G_1 = (\mathrm{AKIN} - N_f kT)/Q_{M-1}$

$v_{\xi_1} = v_{\xi_1} + \frac{\Delta t_s}{4} * G_1$

$v_{\xi_1} = v_{\xi_1} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_2}\right)$

$\mathrm{SCALE} = \mathrm{SCALE} * \exp\left(-\frac{\Delta t_s}{2} v_{xi_1}\right)$

$\mathrm{AKIN} = \mathrm{AKIN} * \exp(-\Delta t_s v_{xi_1})$

DO $i = 1, \mathrm{M}$

$\xi_i = \xi_i + \frac{\Delta t_s}{2} * v_{\xi_i}$

ENDDO

$v_{\xi_1} = v_{\xi_1} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_2}\right)$

$G_1 = (\mathrm{AKIN} - N_f kT)/Q_{M-1}$

$v_{\xi_1} = v_{\xi_1} + \frac{\Delta t_s}{4} * G_1$

$v_{\xi_1} = v_{\xi_1} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_2}\right)$

...

$$v_{\xi_{M-1}} = v_{\xi_{M-1}} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_M}\right)$$

$$G_{M-1} = (Q_{M-2} v_{\xi M-2}^2 - kT)/Q_{M-1}$$

$$v_{\xi_{M-1}} = v_{\xi_{M-1}} + \frac{\Delta t_s}{4} * G_{M-1}$$

$$v_{\xi_{M-1}} = v_{\xi_{M-1}} * \exp\left(-\frac{\Delta t_s}{8} v_{\xi_M}\right)$$

$$G_M = (Q_{M-1} v_{\xi_{M-1}}^2 - kT)/Q_M$$

$$v_{\xi_M} = v_{\xi_M} + \frac{\Delta t_s}{4} * G_M$$

ENDDO

ENDDO

$$\mathbf{v} = \mathbf{v} * \text{SCALE} \tag{35}$$

Note, a time-saving feature has been employed. The effect of the operator $\exp(iL_{\text{NHC}} w_j \Delta t/2n_c)$ is to scale the particle velocities by the factor $\exp(-v_{\xi_1} w_j \Delta t/2n_c)$. The only coupling of the particle velocities to the thermostat variables occurs through the total atomic kinetic energy (AKIN), which appears in the force on the first thermostat G_1. Therefore, a velocity scaling factor can be accumulated and applied to the velocities at the end of the procedure. In addition, the total particle kinetic energy can be evolved by multiplying by the factor $\exp(-v_{\xi_1} w_j \Delta t/2n_c)$ at each step in the iteration. The entire propagator may be implemented by performing the procedure defined in equation (35) before and after performing the procedure defined in equation (23). A fifty line Fortran code based on this algorithm is presented in appendix A.

The reversible *NVT* integration method is not altered significantly if an arbitrary set of constraints is placed on the particle degrees of freedom. The operator $\exp(iL_{\text{NHC}} \Delta t/2n_c)$ acts by scaling the particle velocities by factors of $\exp(-v_{\xi_1} w_j \Delta t/2n_c)$ at each step in the NHC multiple time step procedure ($\exp[ax(\partial/\partial x)]\, x = x \exp[a]$). This scaling does not effect a given constraint if it is initially satisfied ($\mathrm{d}\sigma_k/\mathrm{d}t = \sum_i \mathbf{v}_i \cdot \nabla_{\mathbf{r}_i} \sigma_k = 0$). Therefore, the new integration algorithm can be made consistent with a set of constraints by adding the iterative Shake/Rattle algorithm to the velocity–Verlet step in the usual way [18, 19]. Alternatively, the equations of motion generated by Gauss' principle of least constraint can be integrated reversibly in some cases by methods similar to those described earlier [20]. Note that, in a system with constraints and multiple chains of thermostats on the particle degrees of freedom (for example, all atoms of type X thermostatted with one chain and all atoms of type Y with another), atoms involved in a common constraint must be thermostatted by the same chain, independent of integration algorithm. If this is not the case, the number of degrees of freedom (N_f) associated with each individual chain becomes a complicated function of the particle positions.

4.4. *Isothermal–isobaric (NPT) ensemble: isotropic cell fluctuations*

The Liouville operator for equations of motion, equation (6), is

$$iL = iL_{\mathrm{NHC}} - \left(1+\frac{d}{N_{\mathrm{f}}}\right)\sum_{i=1}^{N} v_\varepsilon \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} + [G_\varepsilon - v_\varepsilon v_\xi]\frac{\partial}{\partial v_\varepsilon}$$

$$+ v_\varepsilon \frac{\partial}{\partial \varepsilon} + \sum_{i=1}^{N} [\mathbf{v}_i + v_\varepsilon \mathbf{r}_i]\cdot \nabla_{\mathbf{r}_i} + \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i}\right]\cdot \nabla_{\mathbf{v}_i}, \tag{36}$$

where $\mathbf{v}_i = \mathbf{p}_i/m_i \neq \dot{\mathbf{r}}_i$,

$$G_\varepsilon = \frac{1}{W}\left[\left(1+\frac{d}{N_{\mathrm{f}}}\right)\sum_{i=1}^{N} m_i \mathbf{v}_i^2 + \sum_{i=1}^{N} \mathbf{r}_i \cdot \mathbf{F}_i(\mathbf{r}) - \mathrm{d}V\frac{\partial \phi(\mathbf{r}, V)}{\partial V} - dP_{\mathrm{ext}} V\right],$$

$$\varepsilon = \frac{1}{d}\log V, \tag{37}$$

and iL_{NHC} retains its previous definition except that

$$G_1 = \frac{1}{Q}\left[\sum_{i=1}^{N} m_i \mathbf{v}_i^2 + W v_\varepsilon^2 - (N_{\mathrm{f}}+1)kT\right]. \tag{38}$$

The equations of motion can be integrated using the approximate evolution operator,

$$\exp(iL\Delta t) = \exp\left(iL_{\mathrm{NHCP}}\frac{\Delta t}{2}\right)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp(iL_2\,\Delta t)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp\left(iL_{\mathrm{NHCP}}\frac{\Delta t}{2}\right)$$

$$+ \mathcal{O}(\Delta t^3), \tag{39}$$

where

$$iL_{\mathrm{NHCP}} = iL_{\mathrm{NHC}} + iL_P,$$

$$iL_P = -\left(1+\frac{d}{N_{\mathrm{f}}}\right)\sum_{i=1}^{N} v_\varepsilon \mathbf{v}_i \cdot \nabla_{\mathbf{v}_i} + [G_\varepsilon - v_\varepsilon v_{\xi_1}]\frac{\partial}{\partial v_\varepsilon},$$

$$iL_1 = \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i}\right]\cdot \nabla_{\mathbf{v}_i},$$

$$iL_2 = \sum_{i=1}^{N} [\mathbf{v}_i + v_\varepsilon \mathbf{r}_i]\cdot \nabla_{\mathbf{r}_i} + v_\varepsilon \frac{\partial}{\partial \varepsilon}. \tag{40}$$

This integrator can be implemented readily by analogy with the results of the previous section. The operator, $\exp(iL_{\mathrm{NHCP}}\Delta t/2)$, is decomposed in the manner of

equation (29), and the operator $\exp(iL_{\mathrm{NHCP}} w_j \Delta t/2n_c)$ is factorized similarly to equation (28):

$$\exp\left(iL_{\mathrm{NHCP}}\frac{w_j\Delta t}{2n_c}\right) = \exp\left(\frac{w_j\Delta t}{4n_c}G_M\frac{\partial}{\partial v_{\xi_M}}\right)\exp\left(-\frac{w_j\Delta t}{8n_c}\mathbf{v}_{\xi_M}\mathbf{v}_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)$$

$$\times\exp\left(\frac{w_j\Delta t}{4n_c}G_{M-1}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\ldots\right)$$

$$\times\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_1}v_\varepsilon\frac{\partial}{\partial v_\varepsilon}\right)\exp\left(\frac{w_j\Delta t}{4n_c}G_\varepsilon\frac{\partial}{\partial v_\varepsilon}\right)\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_1}v_\varepsilon\frac{\partial}{\partial v_\varepsilon}\right)$$

$$\times\exp\left(-\frac{w_j\Delta t}{2n_c}\left[v_{\xi_1}+\left(1+\frac{d}{N_f}\right)v_\varepsilon\right]\mathbf{v}_i\cdot\nabla\mathbf{v}_i\right)\exp\left(\frac{w_j\Delta t}{2n_c}\sum_{i=1}^{M}v_{\xi_i}\frac{\partial}{\partial\xi_i}\right)$$

$$\times\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_1}v_\varepsilon\frac{\partial}{\partial v_\varepsilon}\right)\exp\left(\frac{w_j\Delta t}{4n_c}G_\varepsilon\frac{\partial}{\partial v_\varepsilon}\right)\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_1}v_\varepsilon\frac{\partial}{\partial v_\varepsilon}\right)$$

$$\times\ldots\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\exp\left(\frac{w_j\Delta t}{4n_c}G_{M-1}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)$$

$$\times\exp\left(-\frac{w_j\Delta t}{8n_c}v_{\xi_M}v_{\xi_{M-1}}\frac{\partial}{\partial v_{\xi_{M-1}}}\right)\exp\left(\frac{w_j\Delta t}{4n_c}G_M\frac{\partial}{\partial v_{\xi_M}}\right). \tag{41}$$

The direct translation of this operator into computer code can be carried out by analogy with equation (35). The full propagator, equation (41), is applied to the full phase space by first acting with the operator $\exp(iL_{\mathrm{NHCP}}\Delta t/2)$ to update $\{\xi, \mathbf{v}_\xi, v_\varepsilon, \mathbf{v}\}$, then performing a modified velocity–Verlet (MVV) step

$$\mathbf{r}_i(\Delta t) = \mathbf{r}_i(0)\exp\left[\Delta t v_\varepsilon(0)\right]$$

$$+\exp\left[\frac{\Delta t}{2}v_\varepsilon(0)\right]\left(\mathbf{v}_i(0)\Delta t+\frac{\Delta t^2\mathbf{F}_i(\mathbf{r})}{2m_i}\right)\frac{\sinh\left[\frac{\Delta t}{2}v_\varepsilon(0)\right]}{\frac{\Delta t}{2}v_\varepsilon(0)}$$

$$\varepsilon(\Delta t) = \varepsilon(0)+\Delta t v_\varepsilon(0)$$

$$\mathbf{v}_i(\Delta t) = \mathbf{v_i(0)}+\frac{\Delta t}{2m}\left[\mathbf{F}_i(0)+\mathbf{F}_i(\Delta t)\right] \tag{42}$$

to update $\{\varepsilon, v_\varepsilon, \mathbf{v}, \mathbf{r}\}$, and finally, again acting with the operator $\exp(iL_{\mathrm{NHCP}}\Delta t/2)$ this time to update the output of the modified velocity–Verlet procedure. As stated previously, the potentially singular term, $\sinh(x)/x$, can be expanded in a Maclaurin series to arbitrarily high order (e.g., eighth in practice).

The explicit integration method, can be written as

$$\mathbf{r}(\Delta t) = \mathbf{r}_{\mathrm{MVV}}\left[\Delta t; \mathbf{r}(0), \varepsilon(0), \mathbf{v}\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right)\right]$$

$$\mathbf{v}(\Delta t) = \mathbf{v}_{\mathrm{NHCP}}\left[\frac{\Delta t}{2}; \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right)\right]$$

$$\varepsilon(\Delta t) = \varepsilon_{\mathrm{MVV}}\left[\Delta t; \mathbf{r}(0), \varepsilon(0), \mathbf{v}\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right)\right]$$

$$v_\varepsilon(\Delta t) = v_{\varepsilon_{\mathrm{NHCP}}}\left[\frac{\Delta t}{2}; \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right)\right]$$

$$\xi(\Delta t) = \xi_{\mathrm{NHC}}\left[\frac{\Delta t}{2}; \xi\left(\frac{\Delta t}{2}\right), \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right),\right]$$

$$\mathbf{v}_\xi(\Delta t) = \mathbf{v}_{\xi_{\mathrm{NHCP}}}\left[\frac{\Delta t}{2}; \mathbf{v}'(\Delta t), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right)\right], \tag{43}$$

where

$$\mathbf{v}\left(\frac{\Delta t}{2}\right) = \mathbf{v}_{\mathrm{NHCP}}\left[\frac{\Delta t}{2}; \mathbf{v}(0), \mathbf{v}_\xi(0), v_\varepsilon(0)\right]$$

$$\mathbf{v}'(\Delta t) = \mathbf{v}_{\mathrm{MVV}}\left[\Delta t; \mathbf{r}(0), \varepsilon(0), \mathbf{v}\left(\frac{\Delta t}{2}\right), \mathbf{v}_\xi\left(\frac{\Delta t}{2}\right), v_\varepsilon\left(\frac{\Delta t}{2}\right)\right]$$

$$v_\varepsilon\left(\frac{\Delta t}{2}\right) = v_{\varepsilon_{\mathrm{NHCP}}}\left[\frac{\Delta t}{2}; \mathbf{v}(0), \mathbf{v}_\xi(0), v_\varepsilon(0)\right]$$

$$\xi\left(\frac{\Delta t}{2}\right) = \xi_{\mathrm{NHCP}}\left[\frac{\Delta t}{2}; \xi(0), \mathbf{v}(0), \mathbf{v}_\xi(0), v_\varepsilon(0)\right]$$

$$\mathbf{v}_\xi\left(\frac{\Delta t}{2}\right) = \mathbf{v}_{\xi_{\mathrm{NHCP}}}\left[\frac{\Delta t}{2}; \mathbf{v}(0), \mathbf{v}_\xi(0), v_\varepsilon(0)\right]. \tag{44}$$

The notation is analogous to that of the previous section, equation (33). The label MVV refers to the modified velocity–Verlet step equation (42). Equation (43) can be bypassed using the direct translation technique discussed in the previous sections. In appendix B a short Fortran code is given that implements the isotropic constant pressure integrator. The method and the code can be modified easily to place an independent chain of thermostats on the barostat (ε).

4.5. *Isothermal–isobaric (NPT) ensemble: full cell fluctuations*

The equations of motion, equation (13), have the associated Liouville operator,

$$\begin{aligned} iL = iL_{\mathrm{NHC}} &- \sum_{i=1}^{N}\left[\left(\vec{\mathbf{v}}_{\mathrm{g}} + \frac{\mathrm{Tr}[\vec{\mathbf{v}}_{\mathrm{g}}]}{N_{\mathrm{f}}}\vec{\mathbf{I}}\right)\mathbf{v}_i\right]\cdot\nabla_{\mathbf{v}_i} \\ &+ \sum_{\alpha\beta}[(\vec{\mathbf{G}}_{\mathrm{g}})_{\alpha\beta} - (\vec{\mathbf{v}}_{\mathrm{g}})_{\alpha\beta}v_{\xi_1}]\frac{\partial}{\partial(\vec{\mathbf{v}}_{\mathrm{g}})_{\alpha\beta}} \\ &+ \sum_{\alpha\beta}(\vec{\mathbf{v}}_{\mathrm{g}}\vec{\mathbf{h}})_{\alpha\beta}\frac{\partial}{\partial(\vec{\mathbf{h}})_{\alpha\beta}} + \sum_{i=1}^{N}[\mathbf{v}_i + \vec{\mathbf{v}}_{\mathrm{g}}\mathbf{r}_i]\cdot\nabla_{\mathbf{r}_i} + \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i}, \end{aligned} \tag{45}$$

where $\mathbf{v}_i = \mathbf{p}_i/m_i \neq \dot{\mathbf{r}}_i$,

$$(\vec{\vec{G}}_g)_{\alpha\beta} = \frac{1}{W_g}\sum_{i=1}^{N} m_i(\mathbf{v}_i)_\alpha(\mathbf{v}_i)_\beta + \frac{1}{W_g}\left[\left(\frac{1}{N_f}\sum_{i=1}^{N} m_i\mathbf{v}_i^2 - P_{ext}V\right)\delta_{\alpha\beta} + \sum_{i=1}^{N}(\mathbf{F}_i)_\alpha(\mathbf{r}_i)_\beta - (\vec{\vec{\phi}}'\vec{\vec{h}}^t)_{\alpha\beta}\right], \tag{46}$$

and iL_{NHC} retains its previous definition except that

$$G_i = \frac{1}{Q_1}\left[\sum_{i=1}^{N} m_i\mathbf{v}_i^2 + W_g \operatorname{Tr}[\vec{\vec{v}}_g^t\vec{\vec{v}}_g] - (N_f + d^2)kT\right]. \tag{47}$$

The equations of motion can be integrated using the approximate evolution operator,

$$\exp(iL\Delta t) = \exp\left(iL_{NHCP}\frac{\Delta t}{2}\right)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp(iL_2\Delta t)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp\left(iL_{NHCP}\frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^3), \tag{48}$$

where

$$\begin{aligned} iL_{NHCP} &= iL_{NHC} + iL_P, \\ iL_P &= \sum_{\alpha\beta}[(\vec{\vec{G}}_g)_{\alpha\beta} - (\vec{\vec{v}}_g)_{\alpha\beta}v_{\xi_1}]\frac{\partial}{\partial(\vec{\vec{v}}_g)_{\alpha\beta}} - \sum_{i=1}^{N}\left[\left(\vec{\vec{v}}_g + \frac{\operatorname{Tr}[\vec{\vec{v}}_g]}{N_f}\vec{\vec{I}}\right)\mathbf{v}_i\right]\cdot\nabla_{\mathbf{v}_i}, \\ iL_1 &= \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i}, \\ iL_2 &= \sum_{i=1}^{N}[\mathbf{v}_i + \vec{\vec{v}}_g\mathbf{r}_i]\cdot\nabla_{\mathbf{r}_i} + \sum_{\alpha\beta}(\vec{\vec{v}}_g\vec{\vec{h}})_{\alpha\beta}\frac{\partial}{\partial(\vec{\vec{h}})_{\alpha\beta}}. \end{aligned} \tag{49}$$

The procedure used to apply this evolution operator is not very different from that used in the isotropic method. However, a matrix ($\vec{\vec{v}}_g$) appears in the operators that generate the particle positions $\mathbf{r}_i$ rather than a single variable (v_ε), and a matrix of cell parameters ($\vec{\vec{h}}$) is introduced rather than a scalar variable (ε). Fortunately, $\vec{\vec{v}}_g$ is constrained to be a symmetric matrix (see section 4.4). Therefore, the modified velocity–Verlet part of the procedure becomes

$$\begin{aligned} \mathbf{r}_i(\Delta t) &= \vec{\vec{c}}_g^t(0)\left\{\vec{\vec{I}}_e\left(\frac{\Delta t}{2}\right)\vec{\vec{c}}_g(0)\mathbf{r}_i(0) + \Delta t\,\vec{\vec{I}}_s\left(\frac{\Delta t}{2}\right)\vec{\vec{c}}_g(0)\mathbf{v}_i\left(\frac{\Delta t}{2}\right)\right\} \\ \vec{\vec{h}}(\Delta t) &= \vec{\vec{c}}_g^t(0)\left\{\vec{\vec{I}}_e\left(\frac{\Delta t}{2}\right)\vec{\vec{c}}_g(0)\vec{\vec{h}}(0)\right\} \\ \mathbf{v}_i\left(\frac{\Delta t}{2}\right) &= \mathbf{v}(0) + \frac{\Delta t}{2m_i}[\mathbf{F}_i(0)] \\ \mathbf{v}_i(\Delta t) &= \mathbf{v}_i(0) + \frac{\Delta t}{2m_i}[\mathbf{F}_i(0) + \mathbf{F}_i(\Delta t)], \end{aligned} \tag{50}$$

where

$$\left[\vec{\mathbf{I}}_{\mathrm{e}}\left(\frac{\Delta t}{2}\right)\right]_{\alpha\beta} = \exp\left(\lambda_\alpha \Delta t\right)\delta_{\alpha\beta}$$

$$\left[\vec{\mathbf{I}}_{\mathrm{s}}\left(\frac{\Delta t}{2}\right)\right]_{\alpha\beta} = \exp\left(\lambda_\alpha \frac{\Delta t}{2}\right)\frac{\sinh\left(\lambda_\alpha \dfrac{\Delta t}{2}\right)}{\lambda_\alpha \dfrac{\Delta t}{2}}\delta_{\alpha\beta}, \tag{51}$$

the λ are the eigenvalues of $\vec{\mathbf{v}}_{\mathrm{g}}(\Delta t/2)$, and $\vec{\mathbf{c}}_{\mathrm{g}}(0)$ is the associated matrix of eigenvectors ($\vec{\mathbf{c}}_{\mathrm{g}}^{\mathrm{t}}\vec{\mathbf{v}}_{\mathrm{g}}\vec{\mathbf{c}}_{\mathrm{g}} = \vec{\lambda}$). (Note, for 3×3 matrices the eigenvalues and eigenvectors can be determined analytically.)

A similar scheme is used to apply the operator

$$\hat{O} = \exp\left\{-t'\sum_{i=1}^{N}\left[\left(\vec{\mathbf{v}}_{\mathrm{g}} + \left[\frac{\mathrm{Tr}\left(\vec{\mathbf{v}}_{\mathrm{g}}\right)}{N_{\mathrm{f}}} + v_\xi\right]\vec{\mathbf{I}}\right)\mathbf{v}_i\right]\cdot\nabla_{\mathbf{v}_i}\right\} \tag{52}$$

that appears in the multiple time step factorization of $\exp\left(iL_{\mathrm{NHCP}}t\right)$. That is,

$$\mathbf{v}_i(t') = \vec{\mathbf{c}}_{\mathrm{g}}^{\mathrm{t}}(0)\vec{\mathbf{I}}_{\mathrm{e}}(0)\vec{\mathbf{c}}_{\mathrm{g}}(0)\mathbf{v}_i(0), \tag{53}$$

where

$$[\vec{\mathbf{I}}_{\mathrm{e}}(0)]_{\alpha\beta} = \exp\left(-\lambda_\alpha t'\right)\delta_{\alpha\beta}, \tag{54}$$

the λ are the eigenvalues of the matrix $(\vec{\mathbf{v}}_{\mathrm{g}}(0) + [\mathrm{Tr}[\vec{\mathbf{v}}_{\mathrm{g}}(0)]/N_{\mathrm{f}} + v_\xi(0)]\vec{\mathbf{I}})$, and $\vec{\mathbf{c}}_{\mathrm{g}}(0)$ is the associated matrix of eigenvectors. Again, the factors of $\sinh(x)/x$ can be expanded in a Maclaurin series to arbitrarily high order without loss of generality. Note that accumulation of the velocity scaling factor, as in equation (35), cannot be used in conjunction with the application of the operator $\exp\left(iL_{\mathrm{NHCP}}\Delta t/2\right)$ because the velocities are both rotated and scaled (see appendix C).

As before, the use of the direct translation technique is recommended over closed-form expressions for the phase space vector. Fortran code to apply this integrator is presented in appendix C. Note that the isotropic part (i.e., the volume) is calculated in exactly the same way as in the Andersen–Hoover method.

It is often useful to introduce a set of constraints on the particle degrees of freedom into isothermal–isobaric calculations [10]. This is commonly accomplished through the use of one of two methods [10]. The first scheme divides the system into small groups of atoms that share common constraints and couples the barostat (v_ε or $\vec{\mathbf{v}}_{\mathrm{g}}$) only to the centre-of-mass of each such group, and a centre-of-mass pressure virial replaces the full atomic result in the equations of motion. It can be shown that the ensemble average pressure generated by the centre-of-mass virial at constant volume and temperature is given correctly, provided the size of the groups is small in comparison with the size of the simulation cell. If, however, the groups are not rigid bodies, the isobaric ensemble defined by centre-of-mass volume scaling differs from the standard definition. The numerical implementation of the centre-of-mass *NPT* method can be carried out straightforwardly. The centres-of-mass of the groups are evolved by reversible *NPT* integration (isotropic or flexible), while the relative coordinates are evolved by reversible *NVT* integration in conjunction with Shake/Rattle algorithms.

The second *NPT* scheme is formulated by coupling all the atoms in the system to the barostat. The constraints then make a non-trivial contribution to the pressure virial because $V_{\mathrm{c}} = -\sum_i \lambda_i \sigma_i$ is a part of the total potential energy and contributes to the virial. Numerical integration of this latter *NPT* method requires the iteration of a modified procedure, Shake/Rattle plus Roll (appendix D), through the first/second

application of the $\exp(iL_{\mathrm{NHCP}}\Delta t/2)$ operator (see equation (39)). The integration scheme and the Roll algorithm are described in detail in appendix D. Alternatively, the Shake/Rattle plus Roll procedure can be replaced in some cases by a reversible Gaussian dynamics algorithm [20] to form a fully reversible explicit integrator. In contrast to the *NVT* method, the volume and the atoms involved in all sets of constraints formally must be coupled to the same thermostat. This restriction is generally relaxed as the volume is assumed to be approximately dynamically decoupled (large W).

5. Multiple time step integration

It is relatively straightforward to modify the preceding integrators to perform the type of multiple time step integration useful for treating problems involving long-range forces, high frequency vibrations and general problems involving separation of time scales. These integrators, called reference system propagator algorithms (RESPAs; to indicate explicit reversibility, the designation r-RESPA has also been used in the literature [21]) [1], are now defined.

5.1. *NVE-RESPA*

For completeness, the *NVE*-RESPA algorithm [1] is reviewed first. In *NVE*-RESPA, the evolution operator is broken up into several parts

$$\exp(iL\Delta t) = \exp\left(iL_3\frac{\Delta t}{2}\right)\left[\exp\left(iL_1\frac{\delta t}{2}\right)\exp(iL_2\,\delta t)\right.$$
$$\left.\times\exp\left(iL_1\frac{\delta t}{2}\right)\right]^n\exp\left(iL_3\frac{\Delta t}{2}\right)+\mathcal{O}(\Delta t^3), \tag{55}$$

where

$$iL_1 = \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i},$$
$$iL_2 = \sum_{i=1}^{N}\mathbf{v}_i\cdot\nabla_{\mathbf{r}_i},$$
$$iL_3 = \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})-\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i}, \tag{56}$$

and $\delta t = \Delta t/n$. This factorization naturally gives rise to a process wherein the system is propagated using velocity–Verlet integration at small time step δt for n steps under the influence of an arbitrary refernce force $\mathbf{F}^{\mathrm{ref}}$. The error induced by this approximation is corrected by applying $\exp(iL_3\Delta t/2)$ at both the beginning and end of the large time interval Δt. Thus, the presumably computationally expensive true force $\mathbf{F}$, is evaluated only every n steps (multiple time step integration). In practice, the integrator is applied as follows:

$$\mathbf{r}(\Delta t) = \mathbf{r}_{\mathrm{VV}}^{\mathrm{ref}}\left[\mathbf{r}(0),\mathbf{v}(0)+\frac{\Delta t}{2}\left[\frac{\mathbf{F}(0)-\mathbf{F}^{\mathrm{ref}}(0)}{m}\right];n,\delta t\right]$$
$$\mathbf{v}(\Delta t) = \mathbf{v}_{\mathrm{VV}}^{\mathrm{ref}}\left[\mathbf{r}(0),\mathbf{v}(0)+\frac{\Delta t}{2}\left[\frac{\mathbf{F}(0)-\mathbf{F}^{\mathrm{ref}}(0)}{m}\right];n,\delta t\right]$$
$$+\frac{\Delta t}{2}\left[\frac{\mathbf{F}(\Delta t)-\mathbf{F}^{\mathrm{ref}}(\Delta t)}{m}\right]. \tag{57}$$

That is, the aforementioned n-step velocity–Verlet (VV) integration procedure is performed starting with a velocity initial condition modified by the difference between the true force and the reference force at the beginning of each large step. Also, the velocities obtained from the n-step velocity–Verlet procedure must be modified by the difference between the reference force and the true force evaluated at the end of the large time step.

The direct translation technique can be used to develop a conceptually simple RESPA integration procedure. For example, the operator, equation (55), can be rendered into pseudocode as

$$
\begin{aligned}
&\mathbf{v} = \mathbf{v} + \frac{\Delta t}{2m} * (\mathbf{F} - \mathbf{F}^{\mathrm{ref}}) \\
&\mathrm{DO} \quad i = 1, n \\
&\quad \mathbf{v} = \mathbf{v} + \frac{\delta t}{2m} * \mathbf{F}^{\mathrm{ref}} \\
&\quad \mathbf{r} = \mathbf{r} + \delta t * \mathbf{v} \\
&\quad \mathrm{CALL} \quad \mathrm{REF\_FORCE} \\
&\quad \mathbf{v} = \mathbf{v} + \frac{\delta t}{2m} * \mathbf{F}^{\mathrm{ref}} \\
&\mathrm{ENDDO} \\
&\mathrm{CALL} \quad \mathrm{FORCE} \\
&\mathbf{v} = \mathbf{v} + \frac{\Delta t}{2m} * (\mathbf{F} - \mathbf{F}^{\mathrm{ref}})
\end{aligned}
\tag{58}
$$

or

$$
\begin{aligned}
&\mathrm{DO} \quad i = 1, n \\
&\quad \mathbf{v} = \mathbf{v} + \frac{\delta t}{2m} * \mathbf{F}^{\mathrm{now}} \\
&\quad \mathbf{r} = \mathbf{r} + \delta t * \mathbf{v} \\
&\quad \mathrm{CALL} \quad \mathrm{REF\_FORCE} \\
&\quad \mathbf{F}^{\mathrm{now}} = \mathbf{F}^{\mathrm{ref}} \\
&\quad \mathrm{IF}(i.\mathrm{EQ}.n) \quad \mathrm{THEN} \\
&\quad\quad \mathrm{CALL} \quad \mathrm{FORCE} \\
&\quad\quad \mathbf{F}^{\mathrm{now}} = \mathbf{F}^{\mathrm{ref}} + n * (\mathbf{F} - \mathbf{F}^{\mathrm{ref}}) \\
&\quad \mathrm{ENDIF} \\
&\quad \mathbf{v} = \mathbf{v} + \frac{\delta t}{2m} * \mathbf{F}^{\mathrm{now}} \\
&\mathrm{ENDDO}.
\end{aligned}
\tag{59}
$$

The latter formulation, equation (59), is particularly useful when combining RESPA integration with holonomic constraints. The standard holonomic constraint algorithms, Shake [18] and Rattle [2], can be simply inserted in equation (59) in the usual places (Shake before the calls to the forces, and Rattle before the ENDDO; see code in appendix E).

In order to handle problems involving high frequency vibrations, it is generally sufficient to take $\mathbf{F}^{\mathrm{ref}} = \mathbf{F}^{\mathrm{vib}}$. Thus, the system can be evolved under the action of the

large vibrational forces for a small time step δt and the other softer forces applied only for the large time step Δt [1]. This, of course, saves many evaluations of the softer forces which usually take much more CPU time to evaluate than $\mathbf{F}^{\text{vib}}$.

In problems involving long-range forces, strong short-range interactions are also present. These can be separated out and placed in the reference force. For a system of particles interacting via a pair potential consisting of both a short-range repulsive and a long-range attractive interaction (as, for instance, in a Lennard-Jones fluid), it is useful to write the force on each particle as [1]

$$\begin{aligned}\mathbf{F}_i &= \sum_j \mathbf{f}_{ij}(\mathbf{r}_{ij}) = \sum_j \mathbf{f}_{ij}(\mathbf{r}_{ij})\, S(r_{ij}) + \sum_j \mathbf{f}_{ij}(\mathbf{r}_{ij})\,[1 - S(r_{ij})],\\ \mathbf{F}_i &= \mathbf{F}_i^{\text{short}} + \mathbf{F}_i^{\text{long}}.\end{aligned} \tag{60}$$

If $S(r)$ is chosen to change slowly from one to zero as r increases from $r_c - \lambda$ to r_c, for example [1],

$$\begin{aligned} S(r) &= 1 && r < r_c - \lambda,\\ &= 1 + \gamma^2(2\gamma - 3) && r_c - \lambda \geqslant r \leqslant r_c,\\ &= 0 && r > r_c,\\ \gamma &= \frac{r - (r_c - \lambda)}{\lambda}, \end{aligned} \tag{61}$$

then the short-range repulsive force is separated from the long-range attractive part as desired. Computer time is saved because the computation of the reference force, $\mathbf{F}_i^{\text{ref}} = \mathbf{F}_i^{\text{short}}$, requires fewer interparticle distance evaluations than a computation of the full force (due to the small cutoff distance, r_c). The full force, which includes the long-range component, needs to be evaluated only for the large time step.

5.2. *Extended system RESPA*

The approximate evolution operators that generate integrators for the extended systems methods presented in the previous section are of the form

$$\begin{aligned}\exp(iL\Delta t) = \exp\left(iL_{\text{NHC}x}\frac{\Delta t}{2}\right)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp(\mathrm{i}L_2\,\Delta t)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp\left(iL_{\text{NHC}x}\frac{\Delta t}{2}\right)\\ + \mathcal{O}(\Delta t^3),\end{aligned} \tag{62}$$

where $iL_{\text{NHC}x}$ denotes either iL_{NHC} or iL_{NHCP}. In addition, the inner sequence of three operators in equation (62) leads to a velocity–Verlet or modified velocity–Verlet integration. This realization makes it straightforward to define an appropriate RESPA.

The first variant of RESPA presented here, is useful when the evolution prescribed by the operator $\exp(iL_{\text{NHC}x}\,\Delta t/2)$ is slow compared with the time scale associated with the reference force. It is formed by writing

$$\begin{aligned}\exp(iL\Delta t) = \exp\left(iL_{\text{NHC}x}\frac{\Delta t}{2}\right)\exp\left(iL_3\frac{\Delta t}{2}\right)\left[\exp\left(iL_1\frac{\delta t}{2}\right)\exp(iL_2\,\delta t)\exp\left(iL_1\frac{\delta t}{2}\right)\right]^n\\ \times\exp\left(iL_3\frac{\Delta t}{2}\right)\exp\left(iL_{\text{NHC}x}\frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^3),\end{aligned} \tag{63}$$

and is named XO-RESPA (extended system outside-reference system propagator algorithm). In general, XO-RESPA can be applied in systems that have fast vibrations as the time scale associated with the extended system variables is usually chosen to be

slow compared with these motions (i.e., through the masses Q and W). Two exceptions are in path integral molecular dynamics where it is most efficient to have the extended system variables (the thermostats) evolve on the same time scale as the vibrations of the path integral chain polymer [22] and in NPT simulations of systems with stiff bonds (atomic virial) that give rise to a strong coupling between the system and the baro/thermostat velocities. The XO-RESPA factorization may also be applied to the long-range force problem. However, it may be the case that the extended system variables have been chosen to evolve on a time scale close to that of the short-range forces. Such systems require a different RESPA factorization.

If the motion prescribed by $\exp(iL_{\mathrm{NHC}x}t')$ occurs on the same time scale as that generated by the reference force, then a useful RESPA must include the application of this operator for the small time step δt:

$$\begin{aligned}\exp(iL\Delta t) = &\left[\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\exp\left(iL_3\frac{\Delta t}{2}\right)\exp\left(-iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\right]\\ &\times\left[\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\exp\left(iL_1\frac{\delta t}{2}\right)\exp(iL_2\,\delta t)\exp\left(iL_1\frac{\delta t}{2}\right)\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\right]^n\\ &\times\left[\exp\left(-iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\exp\left(iL_3\frac{\Delta t}{2}\right)\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\right]+\mathcal{O}(\Delta t^3).\end{aligned}\tag{64}$$

The resulting integrator, XI-RESPA (extended system inside-reference system propagator algorithm), is constructed so that when $n = 1$ the original extended system algorithm (equation (62)) is recovered. Also, the operator $\exp(-iL_{\mathrm{NHC}x}\delta t/2)$ is never really applied, as can be seen by rewriting XI-RESPA as

$$\begin{aligned}\exp iL\Delta t = &\left[\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\exp\left(iL_3\frac{\Delta t}{2}\right)\exp\left(iL_1\frac{\delta t}{2}\right)\exp(iL_2\,\delta t)\exp\left(iL_1\frac{\delta t}{2}\right)\right.\\ &\left.\times\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\right]\left[\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\exp\left(iL_1\frac{\delta t}{2}\right)\exp(iL_2\,\delta t)\right.\\ &\left.\times\exp\left(iL_1\frac{\delta t}{2}\right)\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\right]^{n-2}\left[\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\exp\left(iL_1\frac{\delta t}{2}\right)\right.\\ &\left.\times\exp(iL_2\,\delta t)\exp\left(iL_1\frac{\delta t}{2}\right)\exp\left(iL_3\frac{\Delta t}{2}\right)\exp\left(iL_{\mathrm{NHC}x}\frac{\delta t}{2}\right)\right]+\mathcal{O}(\Delta t^3).\end{aligned}\tag{65}$$

More specific information on the formulation of XO-RESPA and XI-RESPA for each type of the extended system method is provided in the following subsections.

5.3. *NVT-RESPA*

In the case of NVT-RESPA, the operators, iL_1, iL_2 and iL_3, retain exactly the same definitions as for the NVE-RESPA case. Therefore, NVT-XO-RESPA is simply NVE-RESPA modified by the application of the extended system operator $\exp(iL_{\mathrm{NHC}}\Delta t/2)$ at the beginning and the end of each large time interval Δt. NVT-XI-RESPA is only slightly different. It can be defined as an n time step, reversible, NVT integration of the reference system wherein the particle velocities are modified by the difference between the true force and reference force (as in NVE-RESPA) after/before the first/final ($2n$th) application of $\exp(iL_{\mathrm{NHC}}\delta t/2)$. A Fortran implementation of NVT-XO-RESPA and NVT-XI-RESPA is provided in appendix E.

5.4. *NPT-RESPA*

The *NPT*-RESPA case is somewhat more difficult to formulate than the *NVT* case. Here, the operator factorizations defined in equations (64) and (65) generate an n-step, modified, velocity–Verlet procedure that depends on the degrees of freedom associated with the cell volume (Andersen–Hoover) and cell shape (Parrinello–Rahman). In addition, under *NPT*-XI-RESPA the last/first $\exp(iL_{\mathrm{NHCP}}\delta t)$ operator in the factorization, equation (65), contains different volume or cell shape related forces from the others in the sequence. For isotropic constant pressure, the operators iL_1, iL_2, and iL_3 are

$$\begin{aligned} iL_1 &= \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i},\\ iL_2 &= \sum_{i=1}^{N}[\mathbf{v}_i+\mathbf{r}_i]\cdot\nabla_{\mathbf{r}_i}+v_\varepsilon\frac{\partial}{\partial\varepsilon},\\ iL_3 &= \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i(\mathbf{r})-\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i}, \end{aligned} \tag{66}$$

and in XI-RESPA the operator iL_{NHCP} always contains the reference force on the logarithm of the volume (the reference virial)

$$G_\varepsilon^{\mathrm{ref}} = \left(1+\frac{d}{N_{\mathrm{f}}}\right)\sum_{i=1}^{N} m_i\mathbf{v}_i^2+\sum_{i=1}^{N}\mathbf{r}_i\cdot\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})-dP_{\mathrm{ext}}^{\mathrm{ref}}V, \tag{67}$$

but the first and last operator in the break up, equation (65), also contain n times the difference virial

$$G_\varepsilon^{\mathrm{eff}} = G_\varepsilon^{\mathrm{ref}}+n\left\{\sum_{i=1}^{N}\mathbf{r}_i\cdot[\mathbf{F}_i-\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})]-d(P_{\mathrm{ext}}-P_{\mathrm{ext}}^{\mathrm{ref}})V\right\}, \tag{68}$$

where $n\delta t = \Delta t$. For fully flexible constant pressure, the required operators are

$$\begin{aligned} iL_1 &= \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i},\\ iL_2 &= \sum_{i=1}^{N}[\mathbf{v}_i+\vec{\mathbf{v}}_{\mathrm{g}}\mathbf{r}_i]\cdot\nabla_{\mathbf{r}_i}+\sum_{\alpha\beta}(\vec{\mathbf{v}}_{\mathrm{g}}\vec{\mathbf{h}})_{\alpha\beta}\frac{\partial}{\partial(\vec{\mathbf{h}})_{\alpha\beta}},\\ iL_3 &= \sum_{i=1}^{N}\left[\frac{\mathbf{F}_i-\mathbf{F}_i^{\mathrm{ref}}(\mathbf{r})}{m_i}\right]\cdot\nabla_{\mathbf{v}_i}, \end{aligned} \tag{69}$$

and in XI-RESPA the operator iL_{NHCP} always contains the force on the unit cell (the reference tensorial virial)

$$\begin{aligned} (\vec{\mathbf{F}}_{\mathrm{g}}^{\mathrm{ref}})_{\alpha\beta} &= \sum_{i=1}^{N} m_i(\mathbf{v}_i^{\mathrm{ref}})_\alpha(\mathbf{v}_i)_\beta\\ &\quad+\sum_{i=1}^{N}(\mathbf{F}_i^{\mathrm{ref}})_\alpha(\mathbf{r}_i)_\beta-P_{\mathrm{ext}}^{\mathrm{ref}}V\delta_{\alpha\beta}, \end{aligned} \tag{70}$$

but the first operator and last operator in the break up, equation (65), also contain n times the difference virial,

$$(\vec{\mathbf{F}}_{\mathrm{g}}^{\mathrm{eff}})_{\alpha\beta} = (\vec{\mathbf{F}}_{\mathrm{g}}^{\mathrm{ref}})_{\alpha\beta}+n\left\{\sum_{i=1}^{N}(\mathbf{F}_i-\mathbf{F}_i^{\mathrm{ref}})_\alpha(\mathbf{r}_i)_\beta-[(P_{\mathrm{ext}}-P_{\mathrm{ext}}^{\mathrm{ref}})V]\delta_{\alpha\beta}\right\}. \tag{71}$$

Despite these complications, *NPT*-XO-REPSA and *NPT*-XI-RESPA are applied in an analogous manner to their *NVT* counterparts.

6. Stability of integrators

Newtonian or Hamiltonian (*NVE*) dynamics obey Liouville's theorem and thus preserve the phase space volume. An integrator or map derived from a (symmetric) Trotter breakup of the evolution operator is also phase space volume preserving. Therefore, the dynamics generated by such an integrator or map will conserve not the true Hamiltonian, but rather a modified, time-step dependent Hamiltonian, $\tilde{H}(\Delta t)$ [23]. This occurs because, in one dimension, phase space volume preservation implies Hamiltonian flow. For example, velocity–Verlet integration of $H = p^2/2m + m\omega^2 x^2/2$ conserves $\tilde{H} = p^2/(2m[1-(\omega\Delta t/2)^2]) + m\omega^2 x^2/2$. This guarantees that for all time t, the instantaneous energy $H(t)$ generated by the map will only fluctuate about $H(0)$ and not drift away, thus preserving the integrity of the energy shell over a trajectory. It also indicates that, in the region of stable evolution ($\omega\Delta t < 2$ for the oscillator), the map can possess closed orbits ($d = 1$); the orbits differ from those of the true Hamiltonian. In addition, the map does not generate the exact dynamics of the modified Hamiltonian $\tilde{H}$, but only conserves it (phase errors accumulate). Nonetheless, any Trotter factorization of the evolution operator will, in one dimension, possess, by construction, a desirable level of accuracy. In higher dimensional problems, velocity–Verlet integration (one class of these maps), has at least been observed empirically to produce long trajectories with well behaved energy conservation in many applications [24]. Other factorizations of more recent origin [1] seem to exhibit this same property.

Extended system methods are not Hamiltonian and always have a greater than one-dimensional phase space. It is, therefore, unclear from the standpoint of analytical theory whether or not integrators which correctly weight the phase space (have the correct J) will possess any special properties relative to the conserved quantity H' (which is not a Hamiltonian). It will be demonstrated empirically (see section 7) that Trotter factorizations of the evolution operator with the form

$$\exp(iL\Delta t) = \exp\left(iL_{\mathrm{NHC}x}\frac{\Delta t}{2}\right)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp(iL_2\Delta t)\exp\left(iL_1\frac{\Delta t}{2}\right)\exp\left(iL_{\mathrm{NHC}x}\frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^3) \tag{72}$$

generate dynamics that conserve H' to good tolerance, while factorizations such as

$$\exp(iL\Delta t) = \exp\left(iL_1\frac{\Delta t}{2}\right)\exp\left(iL_{\mathrm{NHC}x}\frac{\Delta t}{2}\right)\exp(iL_2\Delta t)\exp\left(iL_{\mathrm{NHC}x}\frac{\Delta t}{2}\right)\exp\left(iL_1\frac{\Delta t}{2}\right) + \mathcal{O}(\Delta t^3) \tag{73}$$

behave less satisfactorily. One possible explanation of this observation is that applying the quasi-Hamiltonian portion of the evolution operator,

$$\exp(iL_1\Delta t/2)\exp(iL_2\Delta t)\exp(iL_1\Delta t/2),$$

and modifying the result with a high accuracy, multiple time-step treatment of the extended variables (thermostat, barostats, etc.) preserves some of the features of Hamiltonian flow described above. The iterative velocity–Verlet algorithms which are

used commonly in extended systems simulations [24] are formulated in a similar way. The salient difference between the two types of method, i.e., the reversible algorithms described here and iterative velocity–Verlet, is that a multiple time-step treatment of the extended system degrees of freedom is used in place of an iterative self-consistent scheme.

The observed differences in the ability of the two Trotter factorizations, equations (72) and (73), to conserve H' have important ramifications for the multiple step methods defined in the previous section. In general, XO-REPSA give good energy conservation for all choices of n, the number of inner time steps. That is, given a reference force and a fixed value of Δt, the improvement in energy conservation will saturate at large n, as is observed in *NVE*-RESPA (propagation using only the reference force has a finite level of accuracy). The XI-RESPA algorithm, however, will be shown (numerically) to behave differently (see section 7). Energy conservation improves with increasing n, and then degrades. A more accurate reference force (F_{ref} a better description of F) increases the critical value of n. This phenomenon is observed because as n increases, XI-RESPA approaches the less desirable integrator, equation (73) (i.e., $\delta t \to 0$ but Δt finite). Despite theses difficulties, some level of XI-RESPA is often desirable or necessary. Path integral MD simulations [22], for instance, utilize high frequency thermostats to promote the efficient sampling of phase space. *NVT*-XI-RESPA is therefore employed to generate accurate trajectories. Similarly, *NPT*-XI-RESPA integration is required to produce the correct equilibrium volume distribution in systems which contain molecules with stiff intramolecular bonds. Basically, XI-RESPA must be used with a number n of inner RESPA steps commensurate with the quality of the reference force.

7. Results

Applications of the integrators developed in the previous sections to instructive model problems and more realistic atomic as well as molecular systems are presented in this section.

7.1 *Model problems*

In order to illustrate some of the general properties of the reversible integrators, a one-dimensional harmonic oscillator ($m = 1, \omega = 1, kT = 1, Q = 1, Q_i = 1, M = 2$, $x(0) = 1, v(0) = 1, \Delta t = 0{\cdot}03$) is studied under canonical dynamics (*NVT*). Although this is a simple one-dimensional problem, it nonetheless contains sufficient complexity to be of pedagogical value. In addition, very long trajectories (many oscillator 'periods') can be generated, allowing one to determine unambiguously the relative efficiency of various algorithms.

First, the ability the multiple time step Yoshida/Suzuki method [15, 16] defined in equation 29 to reduce the computational overhead associated with the application of the Nosé–Hoover chain (NHC) operator $\exp(iL_{\text{NHC}}\,\Delta t/2)$ is demonstrated. In figure 1, the deviation of the conserved quantity from its initial value, $[E(t)-E(0)]/[E(0)]$, is plotted versus time as a function of the number of inner steps n_c used in the NHC multiple time step procedure, equation (29). Yoshida/Suzuki intetgration, denoted as $n_c \times n_{ys}$ for a n_{ys} step algorithm, is observed to reduce the total number of steps ($n_c\, n_{ys}$) necessary to achieve converged results relative to the simple algorithm (n_c equal length steps, $n_{ys} = 1$) by a factor of from seven to fourteen. (Note, the baseline on the 48×1 result is not flat; it becomes flat at 96×1.)

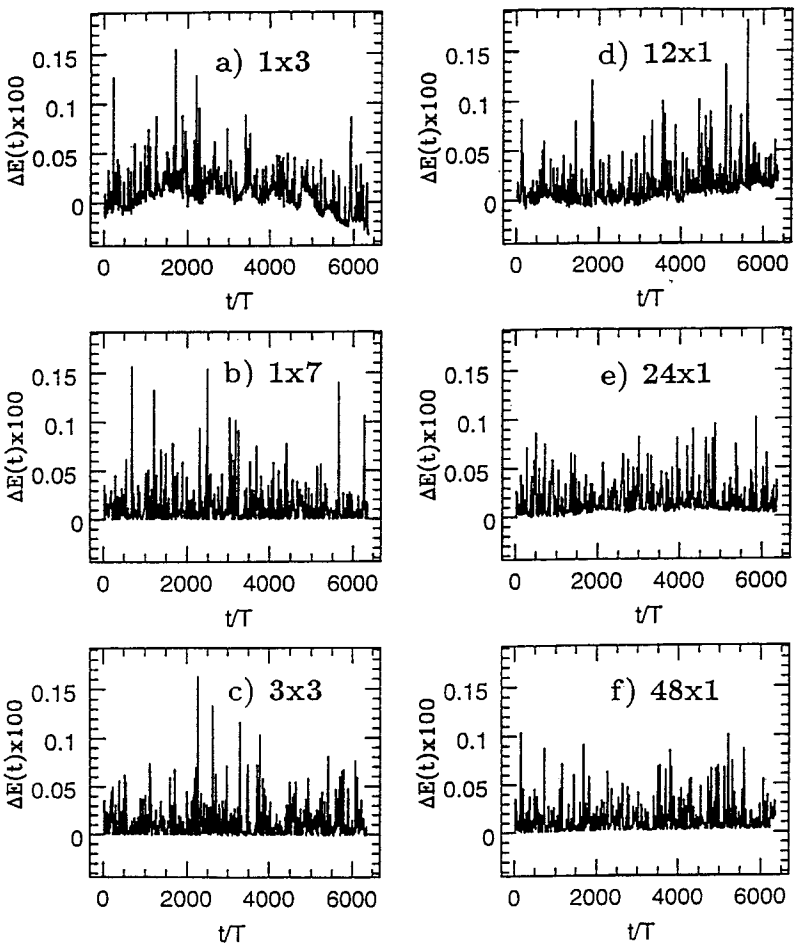

Figure 1. The deviation of the conserved quantity from its initial value, $\Delta E(t) = [E(t) - E(0)]/[E(0)]$, for a harmonic oscillator undergoing Nosé–Hoover chain dynamics ($m = 1, \omega = 1$, $Q_k = 1$), plotted versus time as a function of the number of inner steps n_c used in the NHC multiple time step procedure, equation (29). Yoshida/Suzuki integration is denoted by $n_c \times n_{ys}$ for a total of $n_c \times n_{ys}$ inner steps.

The new reversible *NVT* integrator has been tested against the standard iterative method described in reference [7]. The quantity,

$$\Delta E(\Delta t) = \left(\frac{1}{N}\right) \sum_{k=1}^{N} \left| \frac{E(k\Delta t) - E(0)}{E(0)} \right|, \tag{74}$$

the average deviation of the conserved quantity, is used to assess the accuracy of the trajectories. In figure 2, results for the two *NVT* methods, again for the harmonic oscillator, are presented as a function of time step Δt, for two different values of the NHC parameter $Q = kT\tau^2$. The *NVE* results are shown as a reference. As the NHC time scale τ decreases ($\tau > 1$), the energy conservation of the iterative method degrades while that of the reversible method remains the same. Therefore, when $\tau = 1/\omega = 1$, the reversible method achieves the same relative energy conservation as the iterative method with about twice the same step.

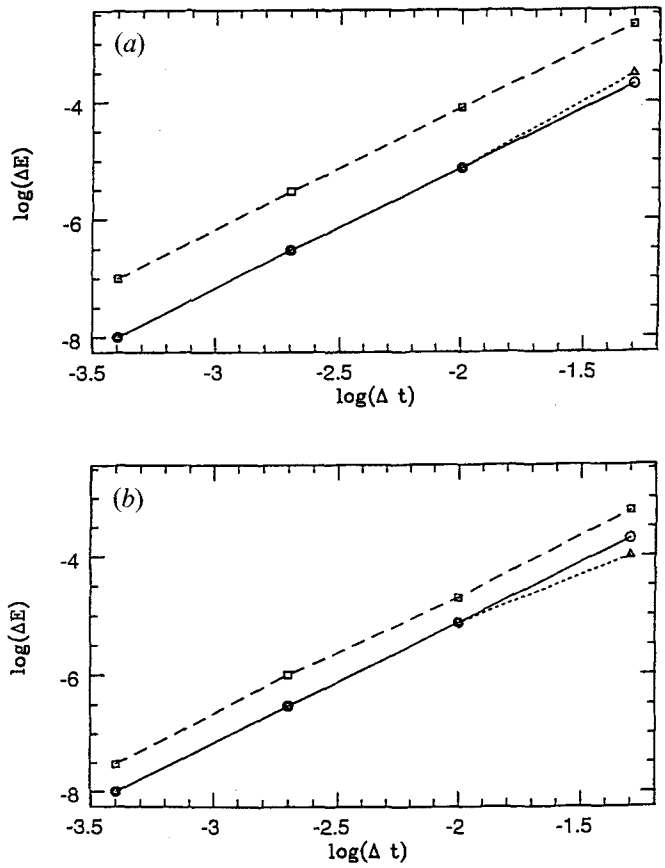

Figure 2. The deviation of the conserved quantity versus time step for a harmonic oscillator undergoing Nosé–Hoover chain dynamics ($m = 1, \omega = 1$) for two different values of the Nosé–Hoover chain parameter Q_k. The squares are the results of iterative velocity–Verlet, the triangles are the results of the new explicit method, and the circles are the results of an *NVE* calculation performed using velocity–Verlet. (*a*) $Q_k = 1$; (*b*) $Q_k = 10$.

The alternative breakup of the Liouville operator, equation (73), has also been investigated. In figure 3, the deviation of the conserved quantity as a function of time, generated by the alternative scheme, is presented. As stated in the previous section, the method is not nearly as well behaved as the recommended formulation (see figure 1).

NVT-XI-RESPA (extended system inside-reference system propagation algorithm) integration of the oscillator system under *NVT* dynamics ($Q = 1$) has been examined in order to illustrate the care that is necessary in the use of the algorithm. The trivial reference force $F_{\text{ref}} = -\lambda\omega^2 x$ is employed for this purpose. In figure 4, the average deviation of the conserved quantity is presented versus the number n of inner time steps used to integrate the reference system. Standard *NVE*-RESPA is included as a benchmark. As discussed in the previous section, *NVT*-XI-RESPA begins to approach the less stable breakup, equation (73), in the large n limit. Hence, there is a critical value of n above which the stability of the algorithm degrades. The critical value of n increases as a reference system becomes more accurate (here, as $\lambda \to 1$). This is in contrast to the *NVE*-RESPA and *NVT*-XO-RESPA methods, which give good

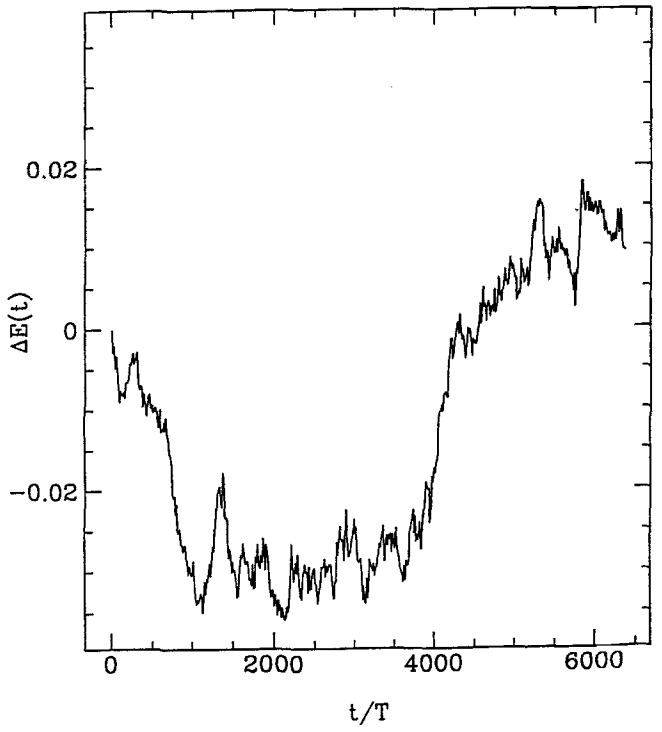

Figure 3. The deviation of the conserved quantity from its initial value, $\Delta E(t) = [E(t) - E(0)]/[E(0)]$, for a harmonic oscillator undergoing Nosé–Hoover chain dynamics ($m = 1, \omega = 1, Q_k = 1$) using decomposition equation (73).

energy conservation for all values of n (i.e., the average deviation of the conserved quantity reaches a plateau). The critical value of n in XI-RESPA is within the plateau region of *NVE*-RESPA. It is, therefore, a useful algorithm. Again, however, n must not be taken larger than warranted by the reference force. The results described above are not peculiar to this simple model and have been observed in many different systems.

7.2. *Lennard-Jones fluid*

A somewhat more realistic system, Lennard-Jones argon ($\varepsilon = 119{\cdot}8$ K, $\sigma = 3{\cdot}405$ Å, $T^* = 0{\cdot}75$, $\rho^* = 0{\cdot}8$, $N = 864$, $R_{\text{cut}} = 3\sigma$), has been studied under *NVE*, and both 'massive' *NVT* and *NPT* dynamics. In a 'massive' calculation [7, 25], an independent NHC is placed on every degree of freedom (including the volume) for a total of $3N$ chains in *NVT* dynamics or $3N+1$ chains in *NPT* dynamics. This represents a stringent test of the reversible method. The extended system time scale parameter used in the calculations was taken to be $\tau = 2500$ fs for both the barostat and thermostats.

In figure 5(*a*), the average deviation of the conserved quantity as a function of time step is presented. The behaviour of the conserved quantity is essentially the same for *NVE*, *NVT* and *NPT* dynamics. The iterative integrators give similar results (not shown). However, the reversible formalism does allow multiple time step integration to be formally developed (a more *ad hoc* iterative scheme has been presented elsewhere [22]). Accordingly, the results of long-range forces XI-RESPA calculations are

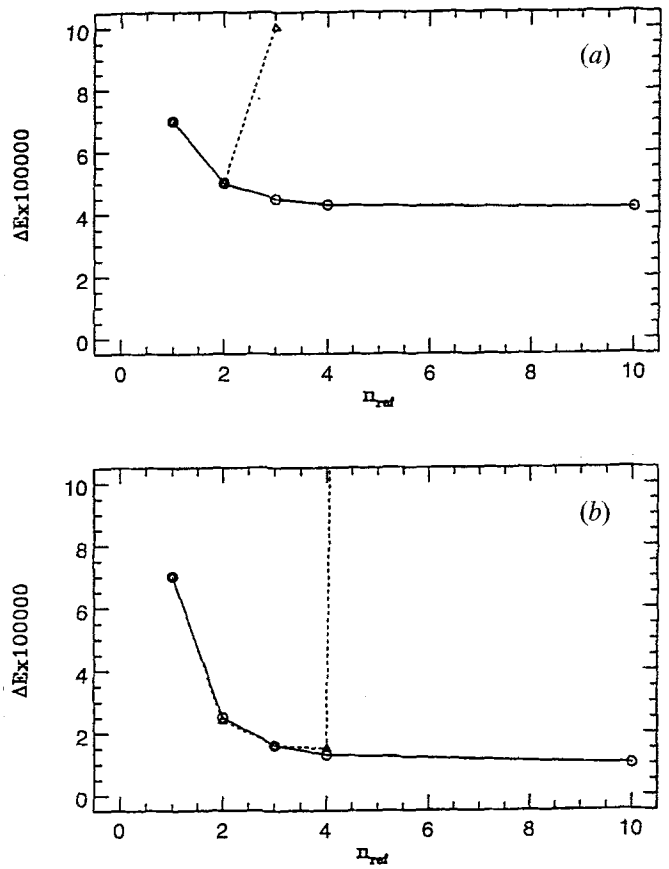

Figure 4. The average deviation of the conserved quantity for a harmonic oscillator undergoing Nosé–Hoover chain ($m = 1, \omega = 1, Q_k = 1$), and Newtonian dynamics, plotted versus the number of inner time steps n_{ref} used to integrate the reference system ($F_{ref} = -\lambda\omega^2 x, \Delta t = 0{\cdot}01$). The circles are Newtonian dynamics, and the triangles are Nosé–Hoover dynamics. (*a*) $\lambda = 0{\cdot}5$; (*b*) $\lambda = 0{\cdot}9$.

presented in figure 5(*b*) for $r_c = 2\sigma, \lambda = 0{\cdot}3\sigma, \delta t = 25$ fs, and n_{ref}, the number of multiple time steps. The energy conservation is essentially unchanged with n_{ref} ranging from 1 to 4.

7.3 *Liquid all-atom butane*

A flexible all-atom butane model [26] has been examined in the liquid phase ($T = 267$ K, $V = 159$ Å^3/molecule, $N = 64$ molecules) under *NVE*, and both 'massive' *NVT* and *NPT* dynamics. The average deviation of the conserved quantity ΔE at the typical time step $\Delta t = 0{\cdot}5$ fs is the same for the three types of dynamics and is independent of integrator. Table 1 shows ΔE from a series of butane simulations using reversible *NVT* and Andersen–Hoover *NPT* integration with no RESPA ($n = 1$), and *NVT*-XO-RESPA and *NPT*-XI-RESPA ($n > 1$). The number n of multiple time steps was chosen to give a constant value for the inner time step $\delta t = 0{\cdot}25$ fs in all of the RESPA calculations. The data in table 1 show that the reversible multiple time step method, *NVT*-XO-RESPA, can be employed to integrate the system to the same level of accuracy as the small time step (0·5 fs) using a 3 fs time step and $n = 12$ RESPA

Table 1. Average deviation of the conserved quantity ΔE from liquid all-atom butane simulations with reversible *NVT* and Andersen–Hoover *NPT* integration with no RESPA ($n = 1$), and *NVT*-XO-RESPA and *NPT*-XI-RESPA ($n > 1$).

		$10^4 \Delta E$	
Δt (fs)	n	*NVT*	*NPT*
0·25	1	0·426	0·344
0·5	1	1·616	1·624
1·0	4	0·473	0·489
2·0	8	0·898	0·739
3·0	12	1·753	1·606
4·0	16	3·838	3·322

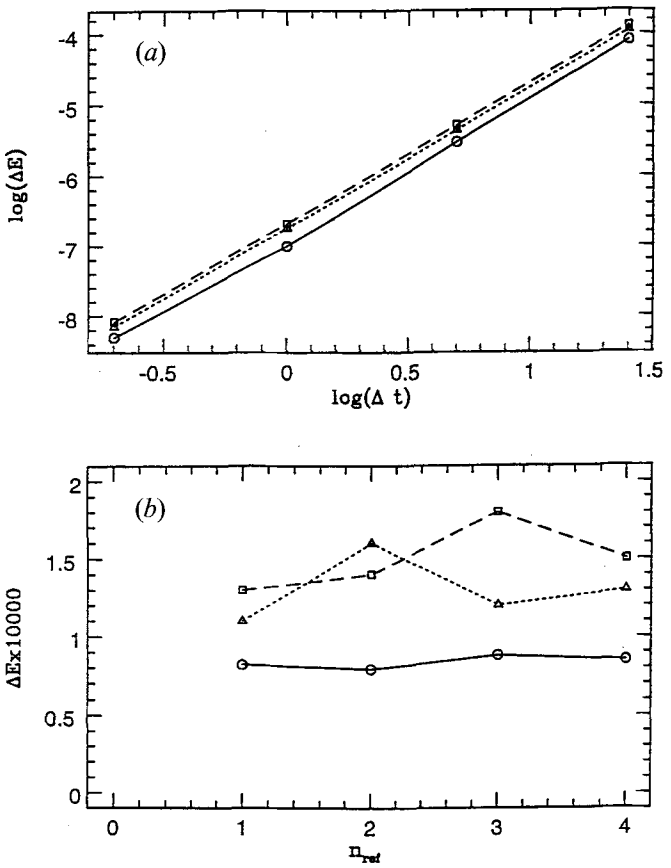

Figure 5. The average deviation of the conserved quantity for a Lennard-Jones system ($\varepsilon = 119{\cdot}8$ K, $\sigma = 3{\cdot}405$ Å, $T^* = 0{\cdot}75$, $\rho^* = 0{\cdot}8$, $N = 864$, $R_{cut} = 3\sigma$) undergoing *NVE* (circles), *NVT* (triangles), and *NPT* (squares) dynamics: (*a*) plotted versus the time step (no RESPA); (*b*) plotted versus the number n_{ref} of RESPA steps under RESPA with $R_{cut}^{(ref)} = 2\sigma, \Delta t = (n_{ref}\, 25)$ fs.

Table 2. Average deviation of the conserved quantity ΔE from liquid pseudoatom tetradecane simulations with Andersen–Hoover *NPT* integration with no RESPA ($n_{\text{lrf}} = n_{\text{intra}} = 1$), and *NPT*-XI-RESPA ($n_{\text{lrf}} = 2, n_{\text{intra}} > 1$).

Δt (fs)	n_{lrf}	n_{intra}	$10^4 \Delta E$
0·5	1	1	0·153
1·0	1	1	0·641
2·0	2	2	0·161
4·0	2	4	0·169
6·0	2	6	0·234
8·0	2	8	0·372
10·0	2	10	0·549
12·0	2	12	0·697
14·0	2	14	0·962

steps (all intramolecular interactions in the reference force but no intermolecular terms). Similar results have been reported elsewhere for *NVE*-RESPA [21]. The Andersen–Hoover equations can be integrated at the same level of accuracy by using *NPT*-XI-RESPA (see table 1). Note that, in constant pressure simulations, *NPT*-XO-RESPA is inappropriate as *NPT*-XI-RESPA integration with a sufficient number of NHC multiple time steps $\{n_c = 3, n_{ys} = 3\}$ is required to give the correct average volume (the volume generated by the small time step calculations).

7.4. *Solid cholesterol acetate*

Cholesterol acetate is a molecular solid of space group P111 at the state point $\{T = 123$ K, $P_{\text{ext}} = 0\}$ [27]. As in the butane example above, *NVT*-XO-RESPA integration allows a time step of 3 fs ($n = 10$) to be used without degradation of energy conservation in simulations of an all-atom model [28] of cholesterol acetate. Reversible Parrinello–Rahman–Hoover *NPT* simulations under *NPT*-XI-RESPA were found to perform similarly well. Both the *NVT*-XO-RESPA and *NPT*-XI-RESPA methods gave results in agreement with all facets of the corresponding small time step calculations. In contrast, *NPT*-XO-RESPA is not sufficiently accurate to yield the correct average volume (i.e., the volume generated by the standard method (no RESPA) using a small time step of 0·5 fs).

7.5. *Liquid pseudoatom tetradecane*

A flexible pseudoatom model [29] of tetradecane has been simulated in the liquid phase ($T = 323$ K, $V = 444$ Å^3/molecule, $N = 48$ molecules) to illustrate the utility of using a double RESPA factorization. Here, the reference force consists of all the intramolecular contributions $\mathbf{F}_{\text{vib}}$ as well as the short-ranged part $\mathbf{F}_{\text{short}}$ of the Lennard-Jones potential ($r_c = 1{\cdot}8\sigma, \lambda = 0{\cdot}25\sigma$). The $\mathbf{F}_{\text{vib}}$ are evaluated at each inner time step δt, the $\mathbf{F}_{\text{short}}$ each n_{intra} inner time steps, and the full force each $n_{\text{lrf}} \times n_{\text{intra}}$ inner time steps. Table 2 shows ΔE from a series of tetradecane simulations using reversible Andersen–Hoover *NPT* integration with no RESPA ($n_{\text{lrf}} = n_{\text{intra}} = 1$) and *NPT*-XI-RESPA ($n_{\text{lrf}} = 2, n_{\text{intra}} > 1$). The number of multiple time steps, $n_{\text{lrf}} \times n_{\text{intra}}$, was chosen to give a constant value for the inner time step $\delta t = 0{\cdot}5$ fs in all of the

RESPA calculations. The data in table 2 show that the reversible multiple time step method, *NPT*-XI-RESPA, can be employed to integrate the equations of motion to the same level of accuracy as the typical, small time step (1 fs) using a time step as large as 12 fs with $n_{\text{lrf}} \times n_{\text{intra}} = 2 \times 12$ RESPA steps.

8. Conclusion

New reversible integrators for extended system dynamics have been derived and applied to both realistic and model problems. The performance of the new methods was found to be as good as, or better than, the standard iterative schemes [24]. The most significant advantage of the reversible integrators is that their multiple time step generalizations (RESPA), which can deliver large savings in CPU time, are easy to define and implement. In addition, the new schemes can be used in conjunction with other methods, such as hybrid Monte Carlo [8], that require reversible evolution.

The research described herein was supported by the National Science Foundation under CHE-92-23546 and the National Institutes of Health under R01 GM40712. G.J.M. acknowledges startup funds from Indiana University and the Petroleum Research Fund (ACS-PRF 28968-G6), and D.J.T. support from the National Institutes of Health (F32 GM14463). The authors thank Chris Mundy for a critical reading of the manuscript and Preston Moore for help with the code.

Appendix A

In this appendix, Fortran code that implements the explicit method for integrating systems in the *NVT* ensemble is presented. A system of N particles is assumed to be coupled to a single Nosé–Hoover chain. Continuation cards have been eliminated for clarity.

```
      SUBROUTINE INTEGRATE
C THIS ROUTINE INTEGRATES THE SYSTEM FROM t=0 TO t=DT
C HERE
C N = THE NUMBER OF PARTICLES
C X,Y,Z = THE PARTICLE POSITIONS
C FX,FY,FZ = THE COMPONENTS OF THE PARTICLE FORCES
C MASS = THE PARTICLE MASSES
C NNOS = M OF THE TEXT
C NNOS1 = M + 1 OF THE TEXT
C XLOGS = ξ OF THE TEXT
C VLOGS = v_ξ OF THE TEXT
C GLOGS = G_i OF THE TEXT
C QMASS = Q_i OF TEXT
C GNKT = N_f kT OF THE TEXT
C GKT = kT OF THE TEXT
C NYOSH = n_ys OF THE TEXT
C NRESN = n_c OF THE TEXT
C DT = Δt OF THE TEXT
C DT2 = Δt/2
C DT22 = Δt^2/2
C WDTI = wΔt/n_c OF THE TEXT
C WDTI2 = wΔt/2n_c ETC.
C
C UPDATE THE PARTICLE VELOCITIES,
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
```

```
      IF(NNOS.GT.0)CALL NHCINT
      DO I = 1,N
         VX(I) = VX(I) + DT2*FX(I)/MASS(I)
         VY(I) = VY(I) + DT2*FY(I)/MASS(I)
         VZ(I) = VZ(I) + DT2*FZ(I)/MASS(I)
      ENDDO
C UPDATE THE PARTICLE POSITIONS
      DO I = 1,N
         X(I) = X(I) + VX(I)*DT
         Y(I) = Y(I) + VY(I)*DT
         Z(I) = Z(I) + VZ(I)*DT
      ENDDO
C APPLY CONSTRAINTS
      CALL SHAKE
C GET THE NEW FORCE
      CALL GETFORCE
C UPDATE THE PARTICLE VELOCITIES
      DO I = 1,N
         VX(I) = VX(I) + DT2*FX(I)/MASS(I)
         VY(I) = VY(I) + DT2*FY(I)/MASS(I)
         VZ(I) = VZ(I) + DT2*FZ(I)/MASS(I)
      ENDDO
C APPLY THE CONSTRAINTS
      CALL RATTLE
C UPDATE THE PARTICLE VELOCITIES,
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
      IF(NNOS.GT.0)CALL NHCINT
      RETURN
      END
C
C
      SUBROUTINE NHCINT
C THIS ROUTINE DOES THE NOSE-HOOVER PART OF THE
C INTEGRATION FROM t=0 TO t=DT/2
C GET THE TOTAL KINETIC ENERGY
      SCALE = 1.D0
      CALL GETKINP(MASS,VX,VY,VZ,AKIN)
C UPDATE THE FORCES
      GLOGS(1) = (AKIN - GNKT)/QMASS(1)
C START THE MULTIPLE TIME STEP PROCEDURE
      DO 15 IRESN = 1,NRESN
         DO 10 IYOSH = 1,NYOSH
C UPDATE THE THERMOSTAT VELOCITIES
            VLOGS(NNOS) = VLOGS(NNOS) + GLOGS(NNOS)*WDTI4(IYOSH)
            DO INOS = 1,NNOS-1
               AA = EXP( -WDTI8(IYOSH)*VLOGS(NNOS1-INOS) )
               VLOGS(NNOS-INOS) = VLOGS(NNOS-INOS)*AA*AA
                    + WDTI4(IYOSH)*GLOGS(NNOS-INOS)*AA
            ENDDO
C UPDATE THE PARTICLE VELOCITIES
            AA = EXP( -WDTI2(IYOSH)*VLOGS(1) )
            SCALE = SCALE*AA
C UPDATE THE FORCES
            GLOGS(1) = (SCALE*SCALE*AKIN - GNKT)/QMASS(1)
C UPDATE THE THERMOSTAT POSITIONS
            DO INOS = 1,NNOS
               XLOGS(INOS) = XLOGS(INOS) + VLOGS(INOS)*WDTI2(IYOSH)
            ENDDO
```

```
C UPDATE THE THERMOSTAT VELOCITIES
            DO INOS = 1,NNOS-1
               AA = EXP( -WDTI8(IYOSH)*VLOGS(INOS+1) )
               VLOGS(INOS) = VLOGS(INOS)*AA*AA
                           + WDTI4(IYOSH)*GLOGS(INOS)*AA
               GLOGS(INOS+1) = (QMASS(INOS)*VLOGS(INOS)*VLOGS(INOS)
                             -GKT)/QMASS(INOS+1)
            ENDDO
            VLOGS(NNOS) = VLOGS(NNOS) + GLOGS(NNOS)*WDTI4(IYOSH)
 10      CONTINUE
 15   CONTINUE
C UPDATE THE PARTICLE VELOCITIES
      DO I = 1,N
         VX(I) = VX(I)*SCALE
         VY(I) = VY(I)*SCALE
         VZ(I) = VZ(I)*SCALE
      ENDDO
      RETURN
      END
```

Appendix B

In this appendix, Fortran code that implements the explicit method for integrating systems undergoing Andersen–Hoover-*NPT* dynamics (isotropic simulation cell fluctuations) is presented. A system of N particles and the volume is assumed to be coupled to a single Nosé–Hoover chain. Continuation cards have been eliminated for clarity. (See the comments in the code presented in appendix A for a complete definition of the variables.)

```
      SUBROUTINE INTEGRATE
C THIS ROUTINE INTEGRATES THE SYSTEM FROM t=0 TO t=DT
C HERE
C XLOGV = ε OF THE TEXT
C VLOGV = v_ε OF THE TEXT
C GLOGV = G_ε OF THE TEXT
C VMASS = W_ε OF THE TEXT
C ODNF = 1 + d/N_f WHERE d = 3
C GN1KT = (N_f + 1)kT
C PINT = is the internal Pressure minus velocity-dependent part
C PEXT = is the external Pressure
      PARAMETER(E2=1.D0/6.D0,E4=E2/20.D0,
                E6=E4/42.D0,E8=E6/72.D0)
C
C UPDATE THE PARTICLE VELOCITIES, dLOG(V)/dt,
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
      CALL NHCPISOINT
      DO I = 1,N
         VX(I) = VX(I) + DT2*FX(I)/MASS(I)
         VY(I) = VY(I) + DT2*FY(I)/MASS(I)
         VZ(I) = VZ(I) + DT2*FZ(I)/MASS(I)
      ENDDO
C UPDATE THE PARTICLE POSITIONS
      AA = EXP(DT2*VLOGV)
```

```
            AA2 = AA*AA
            ARG2 = (VLOGV*DT2)*(VLOGV*DT2)
            POLY = (((E8*ARG2+E6)*ARG2+E4)*ARG2+E2)*ARG2+1.D0
            BB = AA*POLY*DT
            DO I = 1,N
            X(I) = X(I)*AA2 + VX(I)*BB
            Y(I) = Y(I)*AA2 + VY(I)*BB
            Z(I) = Z(I)*AA2 + VZ(I)*BB
         ENDDO
C UPDATE LOG(V)
         XLOGV = XLOGV + VLOGV*DT
C GET THE NEW FORCE
         CALL GETFORCE
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
         DO I = 1,N
            VX(I) = VX(I) + DT2*FX(I)/MASS(I)
            VY(I) = VY(I) + DT2*FY(I)/MASS(I)
            VZ(I) = VZ(I) + DT2*FZ(I)/MASS(I)
         ENDDO
         CALL NHCPISOINT
         RETURN
         END
C
C
         SUBROUTINE NHCPISOINT
C THIS ROUTINE DOES THE
C ANDERSEN-HOOVER P/NOSE-HOOVER CHAIN
C PART OF THE INTEGRATION FROM t=0 TO t=DT/2
C GET THE TOTAL KINETIC ENERGY
         SCALE = 1.D0
         CALL GETKINP(MASS,VX,VY,VZ,AKIN)
C UPDATE THE FORCES
         GLOGS(1) = (AKIN + VMASS*VLOGV*VLOGV - GN1KT)/QMASS(1)
         GLOGV = (ODNF*AKIN+3.D0*(PINT-PEXT)*VOL)/VMASS
C START THE MULTIPLE TIME STEP PROCEDURE
         DO 15 IRESN = 1,NRESN
            DO 10 IYOSH = 1,NYOSH
C UPDATE THE THERMOSTAT VELOCITIES
               VLOGS(NNOS) = VLOGS(NNOS) + GLOGS(NNOS)*WDTI4(IYOSH)
               DO INOS = 1,NNOS-1
                  AA = EXP( -WDTI8(IYOSH)*VLOGS(NNOS1-INOS) )
                  VLOGS(NNOS-INOS) = VLOGS(NNOS-INOS)*AA*AA
                        + WDTI4(IYOSH)*GLOGS(NNOS-INOS)*AA
               ENDDO
C UPDATE dLOG(V)/dt
               AA = EXP( -WDTI8(IYOSH)*VLOGS(1) )
               VLOGV = VLOGV*AA*AA + WDTI4(IYOSH)*GLOGV*AA
C UPDATE THE PARTICLE VELOCITIES
               AA = EXP(-WDTI2(IYOSH)*(VLOGS(1)+ODNF*VLOGV))
               SCALE = SCALE*AA
               AKIN = AKIN*AA*AA
               GLOGV = (ODNF*AKIN+3.D0*(PINT-PEXT)*VOL)/VMASS
C UPDATE THE THERMOSTAT POSITIONS
               DO INOS = 1,NNOS
                  XLOGS(INOS) = XLOGS(INOS) + VLOGS(INOS)*WDTI2(IYOSH)
               ENDDO
C UPDATE dLOG(V)/dt
               AA = EXP( -WDTI8(IYOSH)*VLOGS(1) )
```

```
            VLOGV = VLOGV*AA*AA + WDTI4(IYOSH)*GLOGV*AA
C UPDATE THE FORCES
            GLOGS(1) = (AKIN + VMASS*VLOGV*VLOGV
                        - GN1KT)/QMASS(1)
C UPDATE THE THERMOSTAT VELOCITIES
            DO INOS = 1,NNOS-1
              AA = EXP( -WDTI8(IYOSH)*VLOGS(INOS+1) )
              VLOGS(INOS) = VLOGS(INOS)*AA*AA
                          + WDTI4(IYOSH)*GLOGS(INOS)*AA
              GLOGS(INOS+1) = (QMASS(INOS)*VLOGS(INOS)*VLOGS(INOS)
                              -GKT)/QMASS(INOS+1)
            ENDDO
            VLOGS(NNOS) = VLOGS(NNOS) + GLOGS(NNOS)*WDTI4(IYOSH)
 10       CONTINUE
 15     CONTINUE
C UPDATE THE PARTICLE VELOCITIES
        DO I = 1,N
          VX(I) = VX(I)*SCALE
          VY(I) = VY(I)*SCALE
          VZ(I) = VZ(I)*SCALE
        ENDDO
        RETURN
        END
```

Appendix C

In this appendix, Fortran code that implements the explicit method for integrating systems undergoing Parrinello–Rahman–Hoover *NPT* dynamics is presented. A system of N particles and the box variables is assumed to be coupled to a single Nosé–Hoover chain. Continuation cards have been eliminated for clarity. (See the comments in the code presented in appendices A and B for a complete definition of the variables.)

```
        SUBROUTINE INTEGRATE
C THIS ROUTINE INTEGRATES THE SYSTEM FROM t=0 TO t=DT
C HERE
C BOX = h⃡ OF THE TEXT
C VBOXG = P⃡_g/W_g OF THE TEXT
C VIEG = EIGENVALUES OF VBOXG
C VIEGV = EIGENVECTORS OF VBOXG
C GBOXG = G⃡_g OF THE TEXT
C PINT = THE INTERNAL PRESSURE TENSOR(MINUS VELOCITY-DEPENDENT PART)
C PEXT = THE EXTERNAL PRESSURE TENSOR = I*PEXT
C AKIN(J,I) = Σ_{k=1}^{N} m_i(v_k)_j(v_k)_i
C ONF = 1/N_f
C ONFM = (1/N_f)I⃡
C
C UPDATE THE PARTICLE VELOCITIES, BOX VELOCITIES,
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
        CALL NHCPFULLINT
        DO I = 1,N
          VX(I) = VX(I) + DT2*FX(I)/MASS(I)
          VY(I) = VY(I) + DT2*FY(I)/MASS(I)
```

```
         VZ(I) = VZ(I) + DT2*FZ(I)/MASS(I)
      ENDDO
C UPDATE THE PARTICLE POSITIONS
      CALL DIAG(VTEMP,VEIG,VEIGV)
      DO I = 1,3
         AA = EXP(DT2*VEIG(I))
         AA2(I) = AA*AA
         ARG2 = (VEIG(I)*DT2)*(VEIG(I)*DT2)
         POLY = (((E8*ARG2+E6)*ARG2+E4)*ARG2+E2)*ARG2+1.D0
         BB(I) = AA*POLY*DT
      ENDDO
      DO I = 1,N
         U1 = X(I)*VEIGV(1,1)+ Y(I)*VEIGV(2,1)+ Z(I)*VEIGV(3,1)
         U2 = X(I)*VEIGV(1,2)+ Y(I)*VEIGV(2,2)+ Z(I)*VEIGV(3,2)
         U3 = X(I)*VEIGV(1,3)+ Y(I)*VEIGV(2,3)+ Z(I)*VEIGV(3,3)
         UV1 = VX(I)*VEIGV(1,1)+VY(I)*VEIGV(2,1)+VZ(I)*VEIGV(3,1)
         UV2 = VX(I)*VEIGV(1,2)+VY(I)*VEIGV(2,2)+VZ(I)*VEIGV(3,2)
         UV3 = VX(I)*VEIGV(1,3)+VY(I)*VEIGV(2,3)+VZ(I)*VEIGV(3,3)
         U1 = U1*AA2(1) + UV1*BB(1)
         U2 = U2*AA2(2) + UV2*BB(2)
         U3 = U3*AA2(3) + UV3*BB(3)
         X(I) = U1*VEIGV(1,1)+ U2*VEIGV(1,2)+ U3*VEIGV(1,3)
         Y(I) = U1*VEIGV(2,1)+ U2*VEIGV(2,2)+ U3*VEIGV(2,3)
         Z(I) = U1*VEIGV(3,1)+ U2*VEIGV(3,2)+ U3*VEIGV(3,3)
      ENDDO
C UPDATE THE BOX
C           UBOX(J,I) = VEIGV(K,J)*BOX(K,I) (SUM OVER K)
      CALL MATMUL(UBOX,VEIGV,BOX,1)
      DO I = 1,3
         DO J = 1,3
            UBOX(J,I) = UBOX(J,I)*AA2(J)
         ENDDO
      ENDDO
C           BOX(J,I) = VEIGV(J,K)*UBOX(K,I) (SUM OVER K)
      CALL MATMUL(BOX,VEIGV,UBOX,0)
C GET THE NEW FORCE
      CALL GETFORCE
C UPDATE THE PARTICLE VELOCITIES, BOX VELOCITIES
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
      DO I = 1,N
         VX(I) = VX(I) + DT2*FX(I)/MASS(I)
         VY(I) = VY(I) + DT2*FY(I)/MASS(I)
         VZ(I) = VZ(I) + DT2*FZ(I)/MASS(I)
      ENDDO
      CALL NHCPFULLINT
      RETURN
      END
C
C
      SUBROUTINE NHCPFULLINT
C THIS ROUTINE DOES THE
C PARRINELLO-RAHMAN-HOOVER/NOSE-HOOVER CHAIN
C PART OF THE INTEGRATION FROM t=0 TO t=DT/2
C GET THE TOTAL KINETIC ENERGY
      CALL GETKINP(MASS,VX,VY,VZ,AKIN,AKINTOT)
C GET BOX KINETIC ENERGY
      CALL GETKINB(BMASS,VBOXG,AKINB)
```

```
C UPDATE THE FORCES
      GLOGS(1) = (AKINTOT + AKINB - GND2KT)/QMASS(1)
      DO I = 1,9
         GBOXG(I,1) = (ONFM(I,1)*AKINTOT+AKIN(I,1)
              + VOL*(PINT(I,1)-PEXT(I,1)))/BMASS
      ENDDO
C START THE MULTIPLE TIME STEP PROCEDURE
      DO 15 IRESN = 1,NRESN
         DO 10 IYOSH = 1,NYOSH
C UPDATE THE THERMOSTAT VELOCITIES
            VLOGS(NNOS) = VLOGS(NNOS) + GLOGS(NNOS)*WDTI4(IYOSH)
            DO INOS = 1,NNOS-1
               AA = EXP( -WDTI8(IYOSH)*VLOGS(NNOS1-INOS) )
               VLOGS(NNOS-INOS) = VLOGS(NNOS-INOS)*AA*AA
                    + WDTI4(IYOSH)*GLOGS(NNOS-INOS)*AA
            ENDDO
C UPDATE THE BOX VELOCITIES
            AA = EXP( -WDTI8(IYOSH)*VLOGS(1) )
            DO I = 1,9
               VBOXG(I,1) = VBOXG(I,1)*AA*AA + WDTI4(IYOSH)*GBOXG(I,1)*AA
            ENDDO
C UPDATE THE THERMOSTAT POSITIONS
            DO INOS = 1,NNOS
               XLOGS(INOS) = XLOGS(INOS) + VLOGS(INOS)*WDTI2(IYOSH)
            ENDDO
C UPDATE THE PARTICLE VELOCITIES
            TRVG = ONF*(VBOXG(1,1)+VBOXG(2,2)+VBOXG(3,3))
            DO J = 1,9
               VTEMP(J,1) = VBOXG(J,1) + TRVG + VXLOGS(1)
            ENDDO
            CALL DIAG(VTEMP,VEIG,VEIGV)
            SC1 = EXP(-VEIG(1)*WDTI2(IYOSH))
            SC2 = EXP(-VEIG(2)*WDTI2(IYOSH))
            SC3 = EXP(-VEIG(3)*WDTI2(IYOSH))
            DO I = 1,N
               UV1 = VX(I)*VEIGV(1,1)+VY(I)*VEIGV(2,1)+VZ(I)*VEIGV(3,1)
               UV2 = VX(I)*VEIGV(1,2)+VY(I)*VEIGV(2,2)+VZ(I)*VEIGV(3,2)
               UV3 = VX(I)*VEIGV(1,3)+VY(I)*VEIGV(2,3)+VZ(I)*VEIGV(3,3)
               UV1 = UV1*SC1
               UV2 = UV2*SC2
               UV3 = UV3*SC3
               VX(I) = UV1*VEIGV(1,1)+ UV2*VEIGV(1,2)+ UV3*VEIGV(1,3)
               VY(I) = UV1*VEIGV(2,1)+ UV2*VEIGV(2,2)+ UV3*VEIGV(2,3)
               VZ(I) = UV1*VEIGV(3,1)+ UV2*VEIGV(3,2)+ UV3*VEIGV(3,3)
            ENDDO
C GET THE TOTAL KINETIC ENERGY
            CALL GETKINP(VX,VY,VZ,AKIN,AKINTOT)
C UPDATE THE FORCES
            DO I = 1,9
               GBOXG(I,1) = (ONFM(I,1)*AKINTOT+AKIN(I,1)
                    +VOL*(PINT(I,1)-PEXT(I,1)))/BMASS
            ENDDO
C UPDATE THE BOX VELOCITIES
            AA = EXP( -WDTI8(IYOSH)*VLOGS(1) )
            DO I = 1,9
               VBOXG(I,1) = VBOXG(I,1)*AA*AA + WDTI4(IYOSH)*GBOXG(I,1)*AA
            ENDDO
C UPDATE THE FORCES
```

```
          CALL GETKINB(BMASS,VBOXG,AKINB)
          GLOGS(1) = (AKINTOT + AKINB - GND2KT)/QMASS(1)
C UPDATE THE THERMOSTAT VELOCITIES
          DO INOS = 1,NNOS-1
            AA = EXP( -WDTI8(IYOSH)*VLOGS(INOS+1) )
            VLOGS(INOS) = VLOGS(INOS)*AA*AA
                        + WDTI4(IYOSH)*GLOGS(INOS)*AA
            GLOGS(INOS+1) = (QMASS(INOS)*VLOGS(INOS)*VLOGS(INOS)
                          -GKT)/QMASS(INOS+1)
          ENDDO
          VLOGS(NNOS) = VLOGS(NNOS) + GLOGS(NNOS)*WDTI4(IYOSH)
 10     CONTINUE
 15   CONTINUE
      RETURN
      END
```

Appendix D

The standard Shake and Rattle constraint method is designed to work in conjunction with the velocity–Verlet integration algorithm (*NVE*) [18]. It also can be used in conjunction with the explicit *NVT* integration scheme as described in section 4.3. However, if constraints are desired in *NPT* simulations, the Shake and Rattle algorithms must be modified. The modification is given the sobriquet 'Roll' for reasons that will be come clear in the derivation.

In the following derivation, it is assumed, as in the *NVT* case, that common thermostats are assigned to variables involved in common sets of constraints. It is not assumed that a single thermostat is assigned to the volume and all the constrained degrees of freedom. The explicit *NPT* integration algorithm can be thought of as evolving the system from $t = 0$ to $t = \Delta t/2$ through the action of $\exp(iL_2\,\Delta t)\exp(iL_1\,\Delta t/2)\exp(iL_{\mathrm{NHCP}}\,\Delta t/2)$ on the initial conditions ($t = 0$) and then from $t = \Delta t/2$ to $t = \Delta t$ through the action of $\exp(iL_{\mathrm{NHCP}}\,\Delta t/2)\exp(iL_1\,\Delta t/2)$ on the conditions generated at $t = \Delta t/2$. The particle positions at $t = \Delta t$ and velocities at $t = \Delta t/2$ can be written in the form

$$\begin{aligned}
\mathbf{r}_i(\Delta t) &= \vec{\mathbf{R}}_x(\lambda,0)\,\mathbf{r}_i(0) \\
&\quad + \vec{\mathbf{R}}_v(\lambda,0)\left(\Delta t\mathbf{v}_i^{(\mathrm{NHCP})}(\lambda,0) + \frac{\Delta t^2}{2m}\left[\mathbf{F}_i(0) + \sum_k \lambda_k \mathbf{F}_{c_i}^{(k)}(0)\right]\right) \\
\mathbf{v}_i\left(\frac{\Delta t}{2}\right) &= \mathbf{v}_i^{(\mathrm{NHCP})}(\lambda,0) + \frac{\Delta t}{2m}\left[\mathbf{F}_i(0) + \sum_k \lambda_k \mathbf{F}_{c_i}^{(k)}(0)\right],
\end{aligned} \tag{D 1}$$

where $\mathbf{v}^{\mathrm{NHCP}}(\lambda,0)$ indicates the action of $\exp(iL_{\mathrm{NHCP}}\,\Delta t/2)$ on $\mathbf{v}(0)$, λ are the Lagrange multipliers and $\lambda_k \mathbf{F}_{c_i}^{(k)}(0)$ are the constraint forces ($\mathbf{F}_{c_i}^{(k)}(0) = -\nabla_{\mathbf{r}_i(0)}\sigma_k(\mathbf{r}(0))$). The tensors $\vec{\mathbf{R}}_v(\lambda,0)$ and $\vec{\mathbf{R}}_x(\lambda,0)$ (see equation (51)) revert to scalars under isotropic Andersen–Hoover dynamics. The dependence of $\mathbf{v}^{(\mathrm{NHCP})}(\lambda,0)$ on the multipliers is induced by the operator $\exp(iL_{\mathrm{NHCP}}\,\Delta t/2)$, which depends directly on the multipliers through the position-dependent part of the pressure tensor

$$(P_{\mathrm{int}}^{(r)})_{\alpha\beta} = \sum_i \left[\mathbf{F}_i + \sum_k \lambda_k \mathbf{F}_{c_i}^{(k)}\right]_\alpha (\mathbf{r}_i)_\beta. \tag{D 2}$$

The velocities at time $t = \Delta t$ can be written exactly as

$$\mathbf{v}_i(\Delta t) = \ddot{\mathbf{R}}_v(\lambda, \Delta t)\, S_i(\lambda, \Delta t)\left[\mathbf{v}_i\left(\frac{\Delta t}{2}\right) + \frac{\Delta t}{2m}\left[\mathbf{F}_i(\Delta t) + \sum_k \lambda_k \mathbf{F}_{c_i}^{(k)}(\Delta t)\right]\right]$$

$$\ddot{\mathbf{v}}_g(\Delta t) = \ddot{\mathbf{v}}_g^{(0)}(\Delta t) + \frac{\Delta t}{2W_g} S_V(\lambda, \Delta t) \sum_k \lambda_k (\mathbf{F}_{c_i}^{(k)}(\Delta t))_\alpha (\mathbf{r}_i(\Delta t))_\beta. \tag{D 3}$$

Here, the tensor $\ddot{\mathbf{R}}_v(\lambda, \Delta t)$, and the scalars $S_i(\lambda, \Delta t)$ and $S_V(\lambda, \Delta t)$ are generated by the action of the evolution operator $\exp(iL_{\mathrm{NHCP}}\Delta t/2)$ on the particle and cell velocities obtained by applying $\exp(iL_1 \Delta t/2)$ to the conditions at $t = \Delta t/2$.

The Shake/Roll procedure is designed to determine the Lagrange multipliers such that the $\mathbf{r}(\Delta t)$ satisfy the constraints $\sigma_k(\mathbf{r}(\Delta t)) = 0$, for all k, to within a desired tolerance. In order to accomplish this, the initial values of the entire phase space $\{\ddot{\mathbf{h}}(0), \ddot{\mathbf{v}}_g(0), \xi(0), \mathbf{v}_{\xi_i}(0), \mathbf{v}(0), \mathbf{r}(0)\}$ are first stored. The evolution operator

$$\exp(iL_2 \Delta t)\exp(iL_1 \Delta t/2)\exp(iL_{\mathrm{NHCP}}\Delta t/2) \tag{D 4}$$

is then applied to these initial conditions to take the system to time $t = \Delta t/2$. The multipliers required in equations (D 1) and (D 2) are taken from the Rattle/Roll procedure of the previous time step. Next, the particle positions, velocities, and position dependent part of the pressure tensor (i.e., the multipliers) are refined iteratively by assuming that the $\{\mathbf{v}_{\mathrm{NHCP}}(\lambda, 0), \ddot{\mathbf{R}}_v(\lambda, 0), \ddot{\mathbf{R}}_x(\lambda, 0)\}$ are approximately independent of the multipliers, the constraints are approximately independent of each other and that it is sufficient to solve a linearized equation for the increment to each of the multipliers. For the *NPT* integration scheme outlined were, equation (D 1), the result is

$$\delta\lambda_k = \frac{-2\sigma_k(\Delta t)}{\Delta t^2 \sum_i m_i^{-1}[\ddot{\mathbf{R}}_v(\lambda, 0)\, \mathbf{F}_{c_i}^{(k)}(0)] \cdot \mathbf{F}_{c_i}^{(k)}(\Delta t)},$$

$$\mathbf{v}^{(\mathrm{new})}\left(\frac{\Delta t}{2}\right) = \mathbf{v}^{(\mathrm{old})}\left(\frac{\Delta t}{2}\right) + \frac{\Delta t}{2m}\delta\lambda_k \mathbf{F}_c^{(k)}(0),$$

$$\mathbf{r}^{(\mathrm{new})}(\Delta t) = \mathbf{r}^{(\mathrm{old})}(\Delta t) + \frac{\Delta t^2}{2m}\ddot{\mathbf{R}}_v(\lambda, 0)\, \delta\lambda_k \mathbf{F}_c^{(k)}(0),$$

$$(P_{\mathrm{int}}^{(r)}(0))_{\alpha\beta}^{(\mathrm{new})} = (P_{\mathrm{int}}^{(r)}(0))_{\alpha\beta}^{(\mathrm{old})} + \frac{1}{V(0)}\delta\lambda_k \sum_i (\mathbf{F}_{c_i}^{(k)}(0))_\alpha (\mathbf{r}_i(0))_\beta,$$

$$\lambda_k^{(\mathrm{new})} = \lambda_k^{(\mathrm{old})} + \delta\lambda_k, \tag{D 5}$$

where $\mathbf{F}_c^{(k)}(\Delta t)$ is evaluated using $\mathbf{r}^{(\mathrm{old})}(\Delta t)$. The only difference between this procedure and the standard Shake method is the insertion of the rotation matrix/tensor $\ddot{\mathbf{R}}_v(\lambda, 0)$, and hence 'roll'. After iterating Shake/Roll, equation (D 5), to convergence by cycling through the constraints a number of times, holding $\{\mathbf{v}_{\mathrm{NHCP}}(\lambda, 0), \ddot{\mathbf{R}}_v(\lambda, 0), \ddot{\mathbf{R}}_x(\lambda, 0)\}$ fixed, new $\{\mathbf{v}_{\mathrm{NHCP}}(\lambda, 0), \ddot{\mathbf{R}}_v(\lambda, 0), \ddot{\mathbf{R}}_x(\lambda, 0)\}$ are generated by applying $\exp(iL_2 \Delta t)\exp(iL_1 \Delta t/2)\exp(iL_{\mathrm{NHCP}}\Delta t/2)$ to the initial conditions. This time, however, the refined multipliers (i.e., the refined constraint forces and position dependent part of the pressure tensor) are inserted in the evolution operator. The Shake/Roll procedure is then applied to the new input. This process of generating $\{\mathbf{v}_{\mathrm{NHCP}}(\lambda, 0), \ddot{\mathbf{R}}_v(\lambda, 0), \ddot{\mathbf{R}}_x(\lambda, 0)\}$, followed by Shake/Roll refinement of the multipliers is continued until final convergence. Here, final convergence means that the action of the operator

$$\exp(iL_2 \Delta t)\exp(iL_1 \Delta t/2)\exp(iL_{\mathrm{NHCP}}\Delta t/2)$$

on the initial conditions results in $\mathbf{r}(\Delta t)$ that satisfy the constraints to the desired tolerance. Although, in principle, it may take many cycles through the algorithm to attain final convergence, in practice, the approximations are sufficiently good that usually only two iterations are necessary. Note, the implementation of the method is simplified if the multipliers are stored. This poses no particular problems on modern computers because the number of multipliers scales linearly with the number of particles.

In contrast to Shake/Roll, the Rattle/Roll scheme is designed to determine the Lagrange multipliers such that the first time derivative of each constraint is zero,

$$\dot{\sigma}_k(\Delta t) = \sum_i [\mathbf{v}_i(\Delta t) + \vec{\mathbf{v}}_g(\Delta t)\,\mathbf{r}_i(\Delta t)] \cdot \nabla_{\mathbf{r}_i(\Delta t)}\, \sigma_k(\mathbf{r}(\Delta t)) = 0, \tag{D 6}$$

within a given tolerance. As in Shake/Roll, the initial conditions $\{\vec{\mathbf{v}}_g(\Delta t/2), \xi(\Delta t/2), \mathbf{v}_{\xi_i}(\Delta t/2), \mathbf{v}(\Delta t/2)\}$ are first stored. The multipliers from the Shake/Roll procedure are then used as input to $\exp(iL_{\mathrm{NHCP}}\Delta t/2)\exp(iL_1\Delta t/2)$ and the initial conditions evolved to $t = \Delta t$. This produces a first guess to $\{\vec{\mathbf{R}}_v(\lambda, \Delta t), S_i(\lambda, \Delta t), S_V(\lambda, \Delta t)\}$. The Rattle/Roll procedure iteratively refines the particle and cell velocities assuming that the $\{\vec{\mathbf{R}}_v(\lambda, \Delta t)\, S_i(\lambda, \Delta t), S_V(\lambda, \Delta t)\}$ are constant, the constraints are independent of each other, and that it is sufficient to solve a linearized equation for the increment to each of the multipliers:

$$\delta\lambda_k = \frac{-2\dot{\sigma}_k(\Delta t)}{\Delta t \sum_i m_i^{-1} [\vec{\mathbf{R}}_v(\lambda, \Delta t)\, S_i(\lambda, \Delta t)\, \mathbf{F}_{c_i}(\Delta t)] \cdot \mathbf{F}_{c_i}(\Delta t)},$$

$$\mathbf{v}_i^{(\mathrm{new})}(\Delta t) = \mathbf{v}_i^{(\mathrm{old})}(\Delta t) + \frac{\Delta t}{2m}\delta\lambda_k\, \vec{\mathbf{R}}_v(\lambda, \Delta t)\, S_i(\lambda, \Delta t)\, \mathbf{F}_{c_i}^{(k)}(\Delta t),$$

$$(P_{\mathrm{int}}^{(\mathrm{r})}(\Delta t))_{\alpha\beta}^{(\mathrm{new})} = (P_{\mathrm{int}}^{(\mathrm{r})}(\Delta t))_{\alpha\beta}^{(\mathrm{old})} + \frac{1}{V(\Delta t)}\delta\lambda_k (\mathbf{F}_{c_i}^{(k)}(\Delta t))_\alpha (\mathbf{r}_i(\Delta t))_\beta,$$

$$\vec{\mathbf{v}}_g^{(\mathrm{new})}(\Delta t) = \vec{\mathbf{v}}_g^{(\mathrm{old})}(\Delta t) + \frac{\Delta t}{2W_g} S_V(\lambda, \Delta t)\, \delta\lambda_k \sum_i (\mathbf{F}_{c_i}^{(k)}(\Delta t))_\alpha (\mathbf{r}_i(\Delta t))_\beta,$$

$$\lambda_k^{(\mathrm{new})} = \lambda_k^{(\mathrm{old})} + \delta\lambda_k, \tag{D 7}$$

where the contribution of the pressure tensor to $\delta\lambda_k$ has been neglected as W_g is generally large. After iterating Rattle/Roll, equation (D 7), to convergence by cycling through the constraints a number of times, holding $\{\vec{\mathbf{R}}_v(\lambda, \Delta t), S_i(\lambda, \Delta t), S_V(\lambda, \Delta t)\}$ fixed, new $\{\vec{\mathbf{R}}_v(\lambda, \Delta t), S_i(\lambda, \Delta t), S_V(\lambda, \Delta t)\}$ are generated using the refined constraint forces and position dependent part of the pressure tensor in

$$\exp(iL_{\mathrm{NHCP}}\Delta t/2)\exp(iL_1\Delta t/2).$$

The Rattle/Roll refinement is then repeated. Note, these subsequent Rattle/Roll calculations require very few iterations to converge as the multipliers are already fairly accurate. The full algorithm, which consists of generating the $\{\vec{\mathbf{R}}_v(\lambda, \Delta t), S_i(\lambda, \Delta t)\, S_V(\lambda, \Delta t)\}$ followed by a Rattle/Roll refinement, converges rapidly (in about three iterations). As in Shake/Roll, convergence is defined to mean that the action of the operator

$$\exp(iL_{\mathrm{NHCP}}\Delta t/2)\exp(iL_1\Delta t/2)$$

on the initial conditions generates $\{\mathbf{v}(\Delta t), \vec{\mathbf{v}}_g(\Delta t)\}$ that satisfy the condition on the time derivatives of the constraints to within the desired tolerance.

Appendix E

In this appendix, Fortran code that implements *NVT*-XO-RESPA and *NVT*-XI-RESPA is presented. A system of N particles is assumed to be coupled to a single Nosé–Hoover chain. Continuation cards have been eliminated for clarity. See the comments in the code presented in the previous appendices for a complete definition of the variables.

```
      SUBROUTINE INTEGRATE
C THIS ROUTINE INTEGRATES THE SYSTEM FROM t=0 TO t=DT
C USE NVT-XO-RESPA OR NVT-XI-RESPA
C HERE
C NRESP = THE NUMBER OF INNER TIME STEPS
C DT = Δt OF THE TEXT
C DTI = δt OF THE TEXT
C
C UPDATE THE PARTICLE VELOCITIES,
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
      IF(XORESP.EQ.1)CALL NHCINT(DT2)
      DO 10 IRES = 1,NRESP
        IF(XIRESP.EQ.1)CALL NHCINT(DTI2)
        IF(IRES.EQ.1)THEN
          DO I = 1,N
            VX(I) = VX(I) + DT2*(FX(I)-FXREF(I))/MASS(I)
            VY(I) = VY(I) + DT2*(FY(I)-FYREF(I))/MASS(I)
            VZ(I) = VZ(I) + DT2*(FZ(I)-FZREF(I))/MASS(I)
          ENDDO
        ENDIF
        DO I = 1,N
          VX(I) = VX(I) + DTI2*FXREF(I)/MASS(I)
          VY(I) = VY(I) + DTI2*FYREF(I)/MASS(I)
          VZ(I) = VZ(I) + DTI2*FZREF(I)/MASS(I)
        ENDDO
C UPDATE THE PARTICLE POSITIONS
        DO I = 1,N
          X(I) = X(I) + VX(I)*DTI
          Y(I) = Y(I) + VY(I)*DTI
          Z(I) = Z(I) + VZ(I)*DTI
        ENDDO
C APPLY CONSTRAINTS
        CALL SHAKE
C GET THE NEW FORCES
        IF(IRES.NE.NRES)CALL GETFREF
        IF(IRES.EQ.NRES)CALL GETFREFANDF
C UPDATE THE PARTICLE VELOCITIES
        DO I = 1,N
          VX(I) = VX(I) + DTI2*FXREF(I)/MASS(I)
          VY(I) = VY(I) + DTI2*FYREF(I)/MASS(I)
          VZ(I) = VZ(I) + DTI2*FZREF(I)/MASS(I)
        ENDDO
        IF(IRES.EQ.NRESP)THEN
          DO I = 1,N
            VX(I) = VX(I) + DT2*(FX(I)-FXREF(I))/MASS(I)
            VY(I) = VY(I) + DT2*(FY(I)-FYREF(I))/MASS(I)
            VZ(I) = VZ(I) + DT2*(FZ(I)-FZREF(I))/MASS(I)
```

```
          ENDDO
        ENDIF
C APPLY THE CONSTRAINTS
        CALL RATTLE
        IF(XIRESP.EQ.1)CALL NHCINT(DTI2)
  10  CONTINUE
C UPDATE THE PARTICLE VELOCITIES,
C THERMOSTAT VELOCITIES AND THERMOSTAT POSITIONS
      IF(XORESP.EQ.1)CALL NHCINT(DT2)
      RETURN
      END
```

References

[1] TUCKERMAN, M., MARTYNA, G. J., and BERNE, B. J., 1992, *J. chem. Phys.*, **97**, 1990.
[2] ANDERSEN, H. C., 1980, *J. chem. Phys.*, **72**, 2384.
[3] NOSÉ, S., 1984, *J. chem. Phys.*, **81**, 511.
[4] HOOVER, W. G., 1985, *Phys. Rev.* A, **31**, 1695.
[5] PARRINELLO, M., and RAHMAN, A., 1980, *Phys. Rev. Lett.*, **45**, 1196.
[6] MARTYNA, G. J., TUCKERMAN, M. E., and KLEIN, M. L., 1992, *J. chem. Phys.*, **97**, 2635.
[7] MARTYNA, G. J., TOBIAS, D. J., and KLEIN, M. L., 1994, *J. chem. Phys.*, **101**, 4177.
[8] DUANE, S., KENNEDY, A. D., PENDLETON, B. J., and ROWETH, D., 1987, *Phys. Lett.* B, **195**, 216.
[9] ARNOLD, V. I., 1978, *Mathematical Methods of Classical Mechanics* (New York: Springer-Verlag).
[10] NOSÉ, S., and KLEIN, M. L., 1983, *Molec. Phys.*, **50**, 1055.
[11] MELCHIONNA, S., CICCOTTI, G., and HOLIAN, B. L., 1993, *Molec. Phys.*, **78**, 533.
[12] SWOPE, W. C., ANDERSEN, H. C., BERENS, P. H., and WILSON, K. R., 1982, *J. chem. Phys.*, **76**, 637.
[13] CREUTZ, M., and GOKSCH, A., 1989, *Phys. Rev. Lett.*, **63**, 9.
[14] TROTTER, H. F., 1959, *Proc. Amer. math. Soc.*, **10**, 545.
[15] YOSHIDA, H., 1990, *Phys. Lett.* A, **150**, 262.
[16] SUZUKI, M., 1991, *J. math. Phys.*, **32**, 400.
[17] RAEDT, H. D., and RAEDT, B. D., 1983, *Phys. Rev.* A, **28**, 3575.
[18] RYCKAERT, J. P., CICCOTTI, G., and BERENDSEN, H. J. C., 1977, *J. comput. Phys.*, **23**, 327.
[19] ANDERSEN, H. C., 1983, *J. comput. Phys.*, **52**, 24.
[20] TUCKERMAN, M., and PARRINELLO, M., 1994, *J. chem. Phys.*, **101**, 1302.
[21] WATANABE, M., and KARPLUS, M., 1993, *J. chem. Phys.*, **99**, 8063.
[22] TUCKERMAN, M. E., BERNE, B. J., MARTYNA, G. J., and KLEIN, M. L., 1993, *J. chem. Phys.*, **99**, 2796.
[23] AUERBACH, S., and FRIEDMAN, A., 1991, *J. comput. Phys.*, **93**, 189.
[24] ALLEN, M. P., and TILDESLEY, D. J., 1989, *Computer Simulation of Liquids* (Oxford: Clarendon Press).
[25] TOBIAS, D. J., MARTYNA, G. J., and KLEIN, M. L., 1993, *J. phys. Chem.*, **97**, 12959.
[26] TOBIAS, D. J., TU, K., and KLEIN, M. L., 1996, *J. phys. Chem.*, in press.
[27] WEBER, H. P., CRAVEN, B. M., SAWZIK, P., and MCMULLAN, R. K., 1991, *Acta crystallogr.* B, **47**, 116.
[28] TU, K., TOBIAS, D. J., and KLEIN, M. L., unpublished.
[29] SIEPMANN, J. L., KARABORNI, S., and SMIT, B., 1993, *Nature (Lond.)*, **365**, 330.