JCTC Journal of Chemical Theory and Computation
ARTICLE
pubs.acs.org/JCTC

Recursive Taylor Series Expansion Method for Rigid-Body Molecular Dynamics

Alexey V. Akimov* and Anatoly B. Kolomeisky

Department of Chemistry, Rice University, Houston, Texas 77005-1892, United States

Supporting Information

ABSTRACT: Molecular dynamics computer simulation methods are very important for understanding mechanisms of chemical, physical, and biological processes. The reliability of molecular dynamics simulations strongly depends on the integration schemes used in the simulations. In this work, we developed new rigid body integration schemes for molecular dynamics simulations. Our approach is based on a numerically exact solution to the free rigid body problem, which is used in the classical propagator splitting scheme. We use the Taylor series expansion of rotational dynamical variables in conjunction with the recursive solution for higher order derivatives of these variables. Such an approach is computationally very efficient, robust, and easy to implement, and it does not employ Jacobi elliptic functions, while still providing the numerically exact solution of the free rigid body problem. Our studies showed that the new integration methods have long-time stability and accuracy properties which are comparable to those of existing symplectic integrators. The extension to the case of a canonical ensemble is also developed, allowing one to perform simulations at constant temperatures.

1. INTRODUCTION

Molecular dynamics (MD) methods are popular theoretical tools for studying systems of various kinds, ranging from proteins,[1−4] membranes,[5−7] and ion channels[8,9] to self-assembled monolayers,[10−13] interfacial systems,[14−17] crystalline objects,[18,19] and biological[20] and artificial[21−23] nanomachines. They rely on the classical mechanical description of the system temporal evolution as well as on the all-atomic description of the inter- and intramolecular potentials. This however implies that many interactions should be computed explicitly at every step of the MD algorithm. On the other hand, many degrees of freedom of the molecular complex under consideration might not be important for its dynamics and other properties. Therefore, in order to avoid unnecessary expensive calculations, one may combine several atoms in rigid fragments and describe the motion of such fragments in contrast to the motion of all individual atoms. Such an approach is known as rigid body molecular dynamics (RBMD). Thus, RBMD neglects some internal degrees of freedom while significantly accelerating calculations of dynamics properties of involved molecules.

In the RBMD method, the motion of rigid bodies comprising the systems is computed explicitly, including both translational and rotational (orientation) degrees of freedom. The evolution of the translational variables (position of the central of mass and the translational momentum) is usually calculated in the same way as in the all-atomic MD method using Verlet-like schemes. The most challenging part of the computation of dynamics of the rigid bodies is related to a solution for rotational variables (e.g., angular momentum and the attitude matrix).

Several RBMD methods have been developed recently.[24−34] Most of them were constructed in such a way that they have important geometric properties of corresponding evolution operations (mapping), such as symplecticity, time reversibility, or both. The properties mentioned are usually the consequence of the propagator construction, based on Trotter decomposition[35,36] of the corresponding Liouville operator. It should be noted that although such a scheme always generates the time reversible mapping, it is not necessary that the mapping always be symplectic.

Using the Trotter decomposition technique, the rigid body dynamics problem is usually decomposed into two parts—the torque-free (or just free) rigid body (FRB) problem and the rest of problem, which includes the effect of forces and torques. The solution of the latter part is usually the same for most of the methods. The diversity of RBMD methods is due to various approaches to solve the FRB problem. This may be achieved by five rotations,[24] four rotations and the Rodrigues formula,[25,30] rotations in quaternion space,[25,28] as well as use of the fact that this problem is analytically solvable in terms of Jacobi elliptic functions.[26,27,33,34] It should be noted that there are some other techniques[31] that do not split the FRB problem from the part which includes the effect of the forces and torques.

The analytic solution of the FRB problem, utilizing the Jacobi elliptic functions, is obviously one which should be chosen due to its exact nature. However, its implementation is quite elaborate because of the necessity to consider many special cases, which may arise from one or another set of initial conditions and properties of the inertia tensors of the rigid bodies. Moreover, although there are various libraries which implement the Jacobi elliptic functions and integrals, they may introduce some code dependencies, which are not always desirable. In addition, use of such functions may require extra calculations to be preformed. On the contrary, although approximate, the methods based on five or four rotations are more easy to implement and more robust to the choice of initial conditions of properties of the rigid bodies as well as cheaper to calculate.

Received: May 17, 2011
Published: August 22, 2011

$$\begin{aligned}&\dot{\vec{r}}_i = \frac{\vec{p}_i}{m_i}\\ &\dot{\vec{p}}_i = 0\\ &\dot{A}_i = -skew(I_i^{-1}\vec{l}_i^{\,b})A_i\\ &\dot{\vec{l}}_i^{\,b} = \vec{l}_i^{\,b} \times I_i^{-1}\vec{l}_i^{\,b}\\ &\qquad\text{(a)}\end{aligned} \qquad \begin{aligned}&\dot{\vec{r}}_i = 0\\ &\dot{\vec{p}}_i = -\frac{\partial \phi}{\partial \vec{r}_i}\\ &\dot{A}_i = 0\\ &\dot{\vec{l}}_i^{\,b} = -rot\left(A_i^T \frac{\partial \phi}{\partial A_i}\right)\\ &\qquad\text{(b)}\end{aligned}$$

Figure 1. Equations of motion generated by sub-Hamiltonians (a) h1 and (b) h2.

Here, we report the robust and easy-to-implement method for solving the FRB problem with machine precision. It relies only on recursive relations defined by Euler equations and does not employ any special functions. This might be advantageous in many situations. It is important to note that our method might be considered simultaneously both symplectic and time-reversible. The time reversibility immediately follows from the propagator construction. Although the symplecticity does not follow from the method structure, it is a consequence of the exact nature of the solution (if properly converged).

We also extend our method as well as some of those reported recently for the case of the NVT ensemble by combining them with Nose−Poincare thermostat.[37−39] Finally, the case study of a water cluster is used to evaluate and compare different integrators.

2. METHOD

2.1. NVE Ensemble. One of the outstanding methods for developing the rigid body molecular dynamics integrators is a symplectic splitting technique.[40−43] It leads to time-reversible and symplectic (usually, but not always) integrators, which are necessary for performing long-time simulations. Furthermore, one of the common splitting ways is to divide the full Hamiltonian (H) of the system of interest into a Hamiltonian of the free rigid body (h_1) and an additional Hamiltonian of interactions (h_2):

$$\begin{aligned} H &= h_1 + h_2\\ h_1 &= T(\vec{p}^{\,N}, \vec{l}^{\,N})\\ h_2 &= \phi(\vec{r}^{\,N}, A^N) \end{aligned} \qquad (1)$$

where $T(\vec{p}^{\,N},\vec{l}^{\,b,N})$ is a kinetic energy term that depends on momenta of the rigid body centers of mass $\vec{p}^{\,N} \equiv \{\vec{p}_i\}, i \in 1{-}N$, and the angular moment of the rigid bodies $\vec{l}^{\,N} \equiv \{\vec{l}_i\}, i \in 1{-}N$, and $\phi(\vec{r}^{\,N},A^N)$ is a potential energy term that depends on the positions of the rigid body centers of mass $\vec{r}^{\,N} \equiv \{\vec{r}_i\}$, $i \in 1{-}N$, and the orientations (attitude matrices) of the rigid bodies $A^N \equiv \{A_i\}$, $i \in 1{-}N$.

Then, the full evolution operator can be factorized: $e^{iLdt} = e^{iL_2(dt)/2}e^{iL_1 dt}e^{iL_2(dt)/2} + O(dt^3)$, where

$$\begin{aligned} &iL : H \rightarrow iL\\ &iL = iL_1 + iL_2\\ &iL = \{\cdot, H\}, iL_1 = \{\cdot, h_1\}, iL_2 = \{\cdot, h_2\} \end{aligned} \qquad (2)$$

are the Liouville operators for corresponding Hamiltonians and $\{\cdot, H\}$ denotes the Poisson bracket generated by Hamiltonian H.

In other words, it is possible to split the torqued rigid body problem into the free rigid body problem (FRB) and a perturbation term[24] (Figure 1).

The equations in Figure 1 can be solved separately. As one can see, the equations (Figure 1a) are the FRB problem, which can be solved in different ways. Existing methods to solve the FRB problem have been discussed in the Introduction. The symplectic splitting scheme of van Zon[26] is especially interesting, because it essentially provides the exact solution to this problem. However, for the correct functioning, many special cases must be treated carefully, including permutations of the axes to satisfy certain conditions. Moreover, the method relies on the set of special functions, which might be expensive to calculate, and it might lead to the additional source of rounding errors.

Another possible solution to the FRB problem is to use the Taylor expansion of both angular momentum $\{\vec{l}_i\}$, $i \in 1{-}N$, and direction vectors $\{\vec{u}_i\}$, $i \in 1{-}N$ (which specify the rigid body orientation) in the evolution time:

$$\begin{aligned} \vec{l}_\alpha(t + dt) &= \sum_{n=0}^{N} \frac{\vec{l}_\alpha^{\,(n)}(t)}{n!} dt^n + O(dt^{N+1}), \alpha \in x,y,z\\ \vec{u}_\alpha(t + dt) &= \sum_{n=0}^{N} \frac{\vec{u}_\alpha^{\,(n)}(t)}{n!} dt^n + O(dt^{N+1}), \alpha \in x,y,z\\ \vec{l}_\alpha^{\,(n)} \equiv \frac{d^n \vec{l}_\alpha}{dt^n},\ & \vec{u}_\alpha^{\,(n)} \equiv \frac{d^n \vec{u}_\alpha}{dt^n}, \alpha \in x,y,z \end{aligned} \qquad (3)$$

Here, we use the fact that the attitude matrix is essentially a set of three orthogonal direction vectors: $A_{I\rightarrow e} = (\vec{u}_1\,\vec{u}_2\,\vec{u}_3)$. Then, the evolution of each of these vectors might be directly obtained using the rigid-body Poisson bracket:[44]

$$\begin{aligned} \{F, G\} = &- \vec{l}\cdot(\nabla_{\vec{l}}F \times \nabla_{\vec{l}}G)\\ &- \vec{u}\cdot(\nabla_{\vec{l}}F \times \nabla_{\vec{u}}G - \nabla_{\vec{l}}G \times \nabla_{\vec{u}}F) \end{aligned} \qquad (4)$$

which explicitly can be written as

$$\dot{\vec{u}} = \{\vec{u}, H_1\} = \vec{u}\cdot(\nabla_{\vec{l}}H_1 \times \nabla_{\vec{u}}\vec{u}) \Leftrightarrow$$

$$\dot{\vec{u}} = \begin{pmatrix} 0 & \omega_z & -\omega_y\\ -\omega_z & 0 & \omega_x\\ \omega_y & -\omega_x & 0 \end{pmatrix} \vec{u} \qquad (5a)$$

$$\begin{aligned} \dot{\vec{l}} = \{\vec{l}, H_1\} = -\vec{l}\cdot(\nabla_{\vec{l}}\vec{l} \times \nabla_{\vec{l}}H_1) \Leftrightarrow \begin{pmatrix} \dot{l}_x\\ \dot{l}_y\\ \dot{l}_z \end{pmatrix}\\ = \begin{pmatrix} \alpha l_y l_z\\ \beta l_x l_z\\ \gamma l_x l_y \end{pmatrix} \end{aligned} \qquad (5b)$$

where

$$\begin{pmatrix} \omega_x\\ \omega_y\\ \omega_z \end{pmatrix} = \begin{pmatrix} Al_x\\ Bl_y\\ Cl_z \end{pmatrix} \text{ and } \alpha = C - B, \beta = A - C, \gamma = B - A \qquad (6)$$

Equation 5a corresponds to the equation of Figure 1a for the attitude matrix, while eq 5b is nothing else but the Euler equation for the free rigid body.

In order to use eqs 3, one must calculate the corresponding derivatives of the angular momenta as well as those for direction vectors up to the desired expansion order. This may easily be done employing the special structure of the equations of motion, which for convenience may be written as

$$\begin{aligned} \dot{l}_x &= \alpha l_y l_z \\ \dot{l}_y &= \beta l_x l_z \\ \dot{l}_z &= \gamma l_x l_y \\ \dot{u}_x &= C l_z u_y - B l_y u_z \\ \dot{u}_y &= -C l_z u_x + A l_x u_z \\ \dot{u}_z &= B l_y u_x - A l_x u_y \end{aligned} \tag{7}$$

Using the Leibniz rule, we can see that each n-th derivative of each variable $(\vec{l}, \vec{u}_1, \vec{u}_2, \vec{u}_3)$ may then be expressed via derivatives of other variables up to the order of $(n - 1)$:

$$\begin{aligned} (l_x)^{(n)} &= (\dot{l}_x)^{(n-1)} = \alpha \sum_{i=0}^{n-1} C_{n-1}^i (l_y)^{(i)} (l_z)^{(n-1-i)} \\ (l_y)^{(n)} &= (\dot{l}_y)^{(n-1)} = \beta \sum_{i=0}^{n-1} C_{n-1}^i (l_x)^{(i)} (l_z)^{(n-1-i)} \\ (l_z)^{(n)} &= (\dot{l}_z)^{(n-1)} = \gamma \sum_{i=0}^{n-1} C_{n-1}^i (l_x)^{(i)} (l_y)^{(n-1-i)} \\ (u_x)^{(n)} &= (\dot{u}_x)^{(n-1)} = \sum_{i=0}^{n-1} C_{n-1}^i (C (l_z)^{(i)} (u_y)^{(n-1-i)} \\ &\quad - B (l_y)^{(i)} (u_z)^{(n-1-i)}) \\ (u_y)^{(n)} &= (\dot{u}_y)^{(n-1)} = \sum_{i=0}^{n-1} C_{n-1}^i (-C (l_z)^{(i)} (u_x)^{(n-1-i)} \\ &\quad + A (l_x)^{(i)} (u_z)^{(n-1-i)}) \\ (u_z)^{(n)} &= (\dot{u}_z)^{(n-1)} = \sum_{i=0}^{n-1} C_{n-1}^i (B (l_y)^{(i)} (u_x)^{(n-1-i)} \\ &\quad - A (l_x)^{(i)} (u_y)^{(n-1-i)}) \end{aligned} \tag{8}$$

where $C_n^i = n!/(i!(n - i)!)$ are the binomial coefficients.

The first derivatives of all variables are calculated using the initial values of the variables themselves, as defined by eqs 7. Second derivatives may then be calculated using the first derivatives as well as the initial values of variables and so on up to the required order. The calculated derivatives then may be plugged into eqs 3 to propagate variables. The length of expansion may be chosen such that the last terms will be comparable to machine precision, which will result in a numerically exact solution of the FRB problem. We refer to the algorithm described above as Terec (TEylor RECursive, phonetically).

The evolution of the orientation of the rigid body may also be described in terms of unit quaternions. In that case, the second part of eqs 7 will read

$$\begin{aligned} \dot{q}_0 &= \frac{1}{2}(-A l_x q_1 - B l_y q_2 - C l_z q_3) \\ \dot{q}_1 &= \frac{1}{2}(A l_x q_0 - B l_y q_3 + C l_z q_2) \\ \dot{q}_2 &= \frac{1}{2}(A l_x q_3 + B l_y q_0 - C l_z q_1) \\ \dot{q}_3 &= \frac{1}{2}(-A l_x q_2 + B l_y q_1 + C l_z q_0) \end{aligned} \tag{9}$$

Similarly to direction vector eqs 8, the recursive equations for unit quaternion will read

$$\begin{aligned} (q_0)^{(n)} &= (\dot{q}_0)^{(n-1)} = \frac{1}{2}\sum_{i=0}^{n-1} C_{n-1}^i (-A (l_x)^{(i)} (q_1)^{(n-1-i)} \\ &\quad - B (l_y)^{(i)} (q_2)^{(n-1-i)} - C (l_z)^{(i)} (q_3)^{(n-1-i)}) \\ (q_1)^{(n)} &= (\dot{q}_1)^{(n-1)} = \frac{1}{2}\sum_{i=0}^{n-1} C_{n-1}^i (A (l_x)^{(i)} (q_0)^{(n-1-i)} \\ &\quad - B (l_y)^{(i)} (q_3)^{(n-1-i)} + C (l_z)^{(i)} (q_2)^{(n-1-i)}) \\ (q_2)^{(n)} &= (\dot{q}_2)^{(n-1)} = \frac{1}{2}\sum_{i=0}^{n-1} C_{n-1}^i (A (l_x)^{(i)} (q_3)^{(n-1-i)} \\ &\quad + B (l_y)^{(i)} (q_0)^{(n-1-i)} - C (l_z)^{(i)} (q_1)^{(n-1-i)}) \\ (q_3)^{(n)} &= (\dot{q}_3)^{(n-1)} = \frac{1}{2}\sum_{i=0}^{n-1} C_{n-1}^i (-A (l_x)^{(i)} (q_2)^{(n-1-i)} \\ &\quad + B (l_y)^{(i)} (q_1)^{(n-1-i)} + C (l_z)^{(i)} (q_0)^{(n-1-i)}) \end{aligned} \tag{10}$$

while the recursive relation for angular momentum will be the same as in eq 8. This version we call qTerec (quaternion Terec).

To facilitate the calculations described in MD simulations, the binomial coefficients up to a required degree may be precomputed once and for all. Also, if the length of the expansion is not long enough, the length of the direction vectors as well as that of the quaternion may change. Thus, we use the renormalization of the unit vectors (quaternion) to cure such a possible problem. It should be noted that although the renormalization of direction vectors will not solve the possible loss of the orthogonality, in the quaternion approach, this is not a problem. However, in our simulations, we have found that possible error in the orthogonality of vectors has practically no effect on dynamics and its stability and accuracy.

2.2. NVT Ensemble. All tested integration schemes (except for Omelyan[31]) can be coupled to a Nose−Poincare thermostat in a straightforward way. The algorithm of Omelyan[31] due to its leapfrog structure is less suitable for this purpose.

The Nose−Poincare thermostat is introduced via an extended system Hamiltonian called the Nose−Poincare Hamiltonian:[37]

$$\begin{aligned} H_{\mathrm{NP}} = s\Bigg[\sum_{i=1}^{N}\left(\frac{p'^2_i}{2m_i s^2}\right) + \sum_{i=1}^{N}\left(\frac{1}{2s^2}\vec{l}'^T_i I_i^{-1}\vec{l}'_i\right) \\ + \phi(\vec{r}^N, A^N) + \frac{p_s^2}{2Q} + g k_B T \ln(s) - H_0\Bigg] \end{aligned} \tag{11}$$

where the primed letters denote the virtual variables and non-primed are real variables

$$\begin{aligned} H_0 &= H(0), H \\ &= \sum_{i=1}^{N}\left(\frac{p'^2_i}{2m_i s^2}\right) + \sum_{i=1}^{N}\left(\frac{1}{2s^2}\vec{l}'^T_i I_i^{-1}\vec{l}'_i\right) \\ &\quad + \phi(\vec{r}^N, A^N) + \frac{p_s^2}{2Q} + g k_B T \ln(s) \end{aligned} \tag{12}$$

g is number of degrees of freedom, and k_B is Boltzmann constant.

The Hamiltonian (eq 11) may then be decomposed onto several sub-Hamiltonians:

$$
\begin{aligned}
H_{\mathrm{NP}} &= H_0 + H_1 + H_2 + H_3 \\
H_0 &= s\left(\sum_{i=1}^{N} \frac{p'^{2}_{i}}{2m_i s^2} + g k_{\mathrm{B}} T \ln(s) - H_0\right) \\
H_1 &= s\sum_{i=1}^{N}\left(\frac{A_i \vec{l}'^{2}_{x,i}}{2s^2} + \frac{B_i \vec{l}'^{2}_{y,i}}{2s^2} + \frac{C_i \vec{l}'^{2}_{z,i}}{2s^2}\right) \\
H_2 &= s\phi(\vec{r}^{N}, A^{N}) \\
H_3 &= s\frac{p_s^2}{2Q}
\end{aligned}
\tag{13}
$$

Every sub-Hamiltonian gives rise to its own evolution operator:

$$
\begin{aligned}
&H_0(s, \vec{p}'^{N}) \rightarrow D_0 = \sum_{i=1}^{N}\left(\frac{\partial H_0}{\partial \vec{p}_i'}\frac{\partial}{\partial \vec{r}_i} - \frac{\partial H_0}{\partial \vec{r}_i}\frac{\partial}{\partial \vec{p}'_i}\right) \\
&+ \left(\frac{\partial H_0}{\partial p_s}\frac{\partial}{\partial s} - \frac{\partial H_0}{\partial s}\frac{\partial}{\partial p_s}\right) = \sum_{i=1}^{N}\left(\frac{\vec{p}'_i}{m_i s}\frac{\partial}{\partial \vec{r}_i}\right) \\
&+ \left(\sum_{i=1}^{N}\frac{p'^{2}_{i}}{2m_i s^2} - g k_{\mathrm{B}} T \ln(s) + H_0 - g k_{\mathrm{B}} T\right)\frac{\partial}{\partial p_s} \\
&H_1(s, \vec{l}'^{N}) \rightarrow D_1 = \left(\sum_{i=1}^{3}\left(\dot{\vec{u}}_i\frac{\partial}{\partial \vec{u}_i} + \dot{\vec{l}}_i\frac{\partial}{\partial \vec{l}_i}\right)\right) \\
&+ \left(\frac{\partial H_1}{\partial p_s}\frac{\partial}{\partial s} - \frac{\partial H_1}{\partial s}\frac{\partial}{\partial p_s}\right) \\
&= \tilde{D} + \left(\sum_{i=1}^{N}\left(\frac{A_i}{2s^2}\vec{l}'^{2}_{x,i} + \frac{B_i}{2s^2}\vec{l}'^{2}_{y,i} + \frac{C_i}{2s^2}\vec{l}'^{2}_{z,i}\right)\right)\frac{\partial}{\partial p_s} \\
&H_2(s, \vec{r}^{N}, A^{N}) \rightarrow D_2 = \sum_{i=1}^{N}\left(\frac{\partial H_2}{\partial \vec{p}_i'}\frac{\partial}{\partial \vec{r}_i} - \frac{\partial H_2}{\partial \vec{r}_i}\frac{\partial}{\partial \vec{p}'_i}\right) \\
&+ \sum_{i=1}^{N}\left(\frac{\partial H_2}{\partial \vec{l}_i'}\frac{\partial}{\partial q_i} - \frac{\partial H_2}{\partial q_i}\frac{\partial}{\partial \vec{l}'_i}\right) \\
&+ \left(\frac{\partial H_2}{\partial p_s}\frac{\partial}{\partial s} - \frac{\partial H_2}{\partial s}\frac{\partial}{\partial p_s}\right) \\
&= \sum_{i=1}^{N}\left(s\vec{F}_i\frac{\partial}{\partial \vec{p}'_i}\right) \\
&+ \sum_{i=1}^{N} s(\vec{\tau}_i)\frac{\partial}{\partial \vec{l}'_i} - \phi(\vec{r}^{N}, q^{N})\frac{\partial}{\partial p_s} \\
&H_3(s, p_s) \rightarrow D_3 = \left(\frac{\partial H_3}{\partial p_s}\frac{\partial}{\partial s} - \frac{\partial H_3}{\partial s}\frac{\partial}{\partial p_s}\right) \\
&= \frac{s p_s}{Q}\frac{\partial}{\partial s} - \frac{p_s^2}{2Q}\frac{\partial}{\partial p_s}
\end{aligned}
\tag{14}
$$

Finally, the full propagator (evolution operator) may be represented as

$$
\begin{aligned}
e^{Ddt} &= e^{D_3 dt/2} e^{D_2 dt/2} e^{D_0 dt} e^{D_1 dt} e^{D_2 dt/2} e^{D_3 dt/2} + O(dt^3) \\
e^{D_1 dt} &= e^{D_{11} dt/2} e^{\tilde{D} dt} e^{D_{11} dt/2} + O(dt^3)
\end{aligned}
\tag{15}
$$

thus leading to the second-order factorization scheme. To build the explicit integrator, we only have to define the action of every composing operator.

The action of operators in D_0, D_2, and D_{11} results in the translation of corresponding variables. The nontrivial operators are thus D_3 and $\tilde{D}$. The action of the first one was described by Nose[38] and may be represented as

$$
\exp[D_3 dt]\begin{pmatrix} s \\ p_s \end{pmatrix} = \begin{pmatrix} s\left(1 + \dfrac{p_s}{2Q}dt\right)^2 \\ p_s/\left(1 + \dfrac{p_s}{2Q}dt\right) \end{pmatrix}
\tag{16}
$$

The operator $\tilde{D}$ describes the FRB problem in scaled angular moments. Hence, the equations of motion it generates are similar to eqs 5a and 5b:

$$
\begin{aligned}
\dot{\vec{u}} &= \{\vec{u}, H_1\} = \vec{u}\cdot(\nabla_{\vec{l}'} H_1 \times \nabla_{\vec{u}}\vec{u}) \Leftrightarrow \\
\dot{\vec{u}} &= \frac{1}{s}\begin{pmatrix} 0 & \omega'_z & -\omega'_y \\ -\omega'_z & 0 & \omega'_x \\ \omega'_y & -\omega'_x & 0 \end{pmatrix}\vec{u}
\end{aligned}
\tag{17a}
$$

$$
\begin{aligned}
\dot{\vec{l}'} &= \{\vec{l}', H_1\} = -\vec{l}'\cdot(\nabla_{\vec{l}'}\vec{l}' \times \nabla_{\vec{l}'} H_1) \Leftrightarrow \begin{pmatrix} \dot{l}'_x \\ \dot{l}'_y \\ \dot{l}'_z \end{pmatrix} \\
&= \frac{1}{s}\begin{pmatrix} \alpha l'_y l'_z \\ \beta l'_x l'_z \\ \gamma l'_x l'_y \end{pmatrix}
\end{aligned}
\tag{17b}
$$

Effectively, this means that the operator $e^{\tilde{D}dt}$ is equivalent to the solution of the FRB problem for the time dt/s. This follows from the fact that if, for some operator $D = B(\partial/(\partial C))$, the evolution operator action is e^{Ddt}:$C(t) \rightarrow C(t + dt)$, then for operator $D' = (1/s)B(\partial/(\partial C))$, the evolution operator $e^{D'dt}$ action will be $e^{D'dt} = e^{D(dt)/(s)}$:$C(t) \rightarrow C(t + (dt)/s)$. Thus, the coupling of the rigid bodies to a Nose–Poincare thermostat is simply the application of the integrators for the NVE case for scaled time with corresponding propagation of the thermostat variables.

For convenience, we present now the full explicit integration scheme to perform RBMD simulations in the NVT ensemble.

1.

$$
e^{D_3 dt/2}:\begin{pmatrix} s \\ p_s \end{pmatrix} \rightarrow \begin{pmatrix} s\left(1 + \dfrac{p_s}{2Q}\dfrac{dt}{2}\right)^2 \\ p_s/\left(1 + \dfrac{p_s}{2Q}\dfrac{dt}{2}\right) \end{pmatrix}
\tag{18}
$$

2.

$$e^{D_2 dt/2} : \begin{pmatrix} \vec{p}'_i \\ \vec{l}'_i \\ p_s \end{pmatrix} \rightarrow \begin{pmatrix} \vec{p}'_i + s\vec{F}_i \frac{dt}{2} \\ \vec{l}'_i + s\vec{\tau}_i \frac{dt}{2} \\ p_s - \phi(\vec{r}^N, q^N) \frac{dt}{2} \end{pmatrix} \tag{19}$$

3.

$$e^{D_{11} dt/2} : p_s \rightarrow p_s + \frac{dt}{2} \sum_{i=1}^{N} \left(\frac{A_i}{2s^2} \vec{l}'^2_{x,i} + \frac{B_i}{2s^2} \vec{l}'^2_{y,i} + \frac{C_i}{2s^2} \vec{l}'^2_{z,i} \right) \tag{20}$$

4. FRB problem propagation with the time step dt/s, e.g., Terec (see eqs 8)
5. Exactly step 3
6.

$$e^{D_0 dt} : \begin{pmatrix} \vec{r}_i \\ p_s \end{pmatrix} \rightarrow \begin{pmatrix} \vec{r}_i + \frac{\vec{p}'_i}{m_i s} dt \\ p_s + \left(\sum_{i=1}^{N} \frac{\vec{p}'^2_i}{2m_i s^2} - g k_B T(\ln(s) + 1) + H_0 \right) dt \end{pmatrix} \tag{21}$$

7. Update atomic coordinates; calculate forces and torques
8. Exactly step 2
9. Exactly step 1

3. RESULTS AND DISCUSSION

In order to test the developed integrators, we performed MD simulations of the $(H_2O)_{23}$ cluster. The TIP3P[45] interaction potential was used to describe intermolecular interactions. Each water molecule was treated as a separate rigid fragment. Thus, the inclusion of intramolecular interactions was not necessary. Each simulation runs for 10^7 steps, which for a time step of 1 fs is equivalent to a 10 ns trajectory. The initial velocity distribution corresponded to a temperature of 250 K.

The algorithms were tested in three stages. In the first stage, we compared the performance of both the Terec and qTerec algorithms for different integration time steps using different expansion sizes. The methods were characterized by two quantities: the total energy trend (b quantity in eq 22a) and the total energy standard deviation (sd(E) in eq 22b).

The former quantity describes the stability of the method and therefore may be a quantitative measure of symplecticity of the method. We calculated it via a linear fit of the total energy versus trajectory time (eq 22a). It should also be noted that the symplecticity of the method cannot be simply judged on the basis of its stability. For this purpose, one should consider the phase space volume preservation. However, in most cases, the good stability of the method might reflect its symplecticness. Therefore, we consider the total energy trend as the measure of symplecticity in this sense.

The latter quantity describes how much of the total energy fluctuates around its mean value, and thus it is a measure of the method precision.

$$|E(t)| = a + bt \tag{22a}$$

$$\mathrm{sd}(E) = \sqrt{\frac{\sum_{i=1}^{N} (E_i - \bar{E})^2}{N}} \qquad \bar{E} = \frac{\sum_{i=1}^{N} E_i}{N} \tag{22b}$$

For each method, we considered the expansion sizes of 5, 7, 10, 12, 15, and 20 terms. Terms are explained in Figure 2. For each expansion size, the simulations with time steps of 0.5, 1, 2.5, 5, and 7.5 fs have been performed. The trajectory lengths varied from 5 to 75 ns accordingly.

We found that all methods showed good stability (Figure 2a) and accuracy properties (Figure 2b). The only exception was the Terec5 method, where the expansion size was not enough to achieve machine accuracy. It is interesting to note that the qTerec5 method showed significantly better properties than its Terec5 cousin. This is a consequence of the deorthogonaliztion of the direction vectors during simulations in the Terec5 method. The use of a quaternion in the qTerec5 variant precludes any problems with a possible deorthogonaliztion of the attitude matrix, thus leading to significantly better properties.

As shown in Figure 2 for the expansions longer than five terms, the properties of the corresponding integrators are almost independent of the expansion size. This indicates that the computations converge to machine precision. In some cases, the longer expansions are in fact slightly less stable and less precise. This may be due to accumulation of the rounding errors when dealing with very small numbers. Thus, for future use or by default, we chose 10 term series expansions for our methods, that is, Terec10 and qTerec10.

All methods are stable and accurate enough for all tested time steps except for 7.5 fs. This is clearly shown in Figure 2a,b as the abrupt change of linear relations $\ln(|E|) - \ln(dt)$ and $\ln(\mathrm{sd}(E)) - \ln(dt)$.

In the second stage, the developed integrators (namely, Terec10 and qTerec10) have been compared to existing integration schemes which we label as DLML,[24] NO_SQUISH,[28] MN,[30] Omelyan,[31] and Jacobi.[26] The first two algorithms are known to be both symplectic and time-reversible, while the next two are not symplectic but are time-reversible. Finally, the method based on an analytical solution to the FRB problem which uses Jacobi integrals of the first kind is by construction exact, so it should in principle be both symplectic and time-reversible as well. The comparison was based on the total energy trend (Figure 3a) as well as on the standard deviation of the total energy (Figure 3b).

As the comparison in Figure 3 illustrates, the Terec and qTerec methods show the properties of the known symplectic integrators (DLML and NO_SQUISH). The nonsymplectic schemes differ significantly from symplectic ones in both stability and precision. These observations are valid for different time steps up to 5 fs. All integrators become unstable at 7.5 fs. The properties of the Jacobi integrator are very similar to those of symplectic schemes as well as to Terec10 and qTerec10

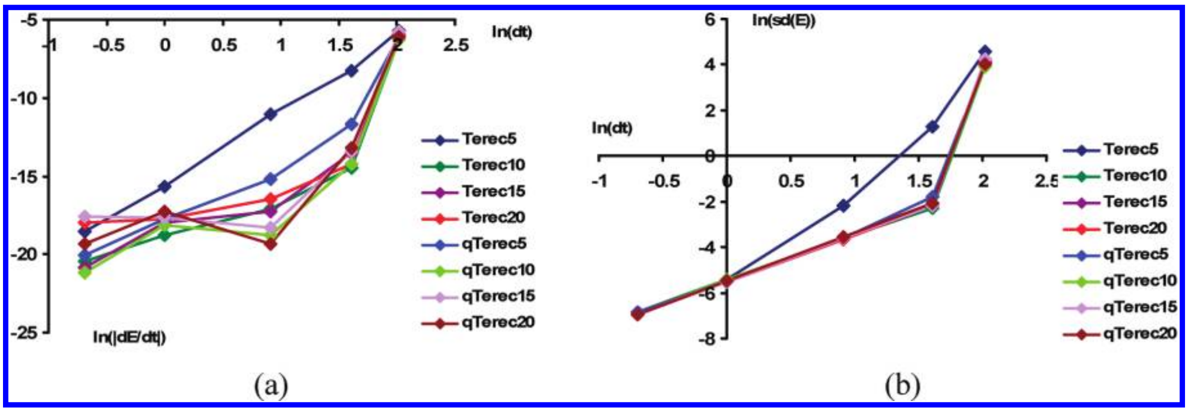

Figure 2. Comparison of the Terec and qTerec methods for different expansion sizes and different time steps. (a) energy trend, stability characteristics; (b) standard deviation, accuracy characteristics.

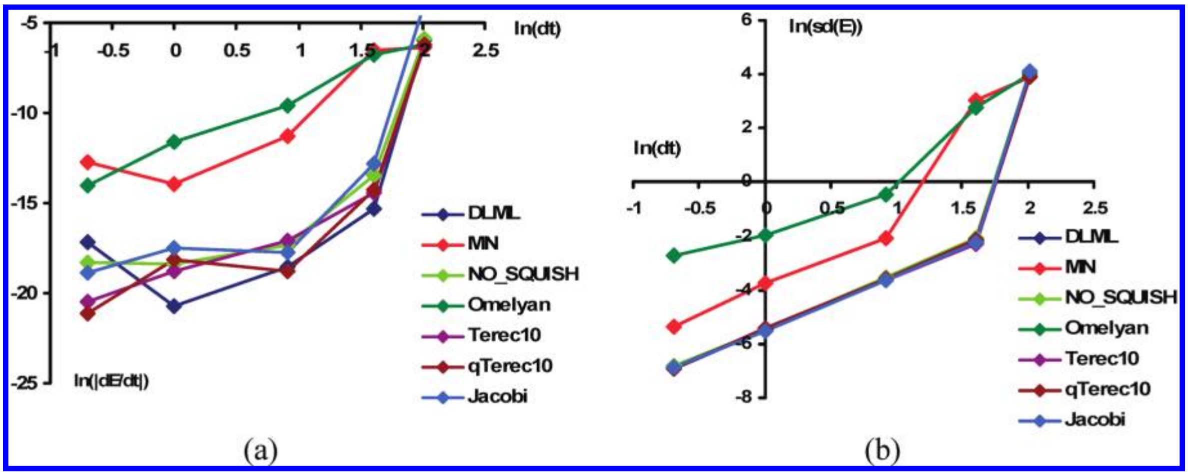

Figure 3. Comparison of Terec and qTerec methods with other existing integrators for NVE ensemble: (a) energy trend, stability characteristics; (b) standard deviation, accuracy characteristics.

algorithms. This indicates that all of them achieve the best possible precision and stability for a given system. It should also be stressed, that despite the algorithm used for integration of the FRB problem, the overall order of algorithms is limited by the smallest order in the entire decomposition scheme (that is where the forces and torques are applied). However, as is clear from Figure 3, the algorithm for solving the FRB problem may significantly impact the stability of the overall algorithms as well as its precision. Thus, it is still important to use an appropriate integrator for the FRB part.

In addition to the quality of conservation of the Hamiltonian (total energy) of the system under consideration, we also studied the analogous properties of the total linear and angular momenta of the system. This is important because the rescaling of the direction vectors and quaternions used in Terec and qTerec methods might potentially influence the conservation of these quantities. Thus, we were interested in how the rescaling affects these quantities. As we expected, the rescaling practically does not affect either the linear or angular momenta significantly, as long as the Taylor series expansion possesses a sufficiently large number of terms. This may be understood in the following way: Assume some quantity x (in our case, it may be either quaternion of the direction vector) has an exact value of x_{exact} and the Taylor series approximation of $x_{\text{Taylor}} = x_{\text{exact}} + dx$, where dx is the error. The rescaled value of x will be $x_{\text{Taylor}}/x_{\text{exact}} = 1 + dx/x_{\text{exact}}$. As the number of terms in the Taylor series expansion increases, the error goes to zero very rapidly ($dx \rightarrow 0$). In that case, the rescaling operation will practically be the identity operation and thus will not have a significant impact on conserved properties. Corresponding data are presented in Supporting Information section S1. It shows that for a relatively small number of terms (Terec5), the rescaling affects conservation of the total linear momentum. However, starting from 10 terms (Terec10), it practically has no effect on the conserved properties and is needed only for consistency.

It should also be noted that if one wants to combine the exact solution to the FRB problem with the torques and forces part of the integrator, it is crucial to consider more than two possible ways to perform Jacobi ordering used in the Jacobi algorithm as implemented by van Zon and Schofield.[26] In fact, there are six distinct permutations of the axes, one of which may lead to required Jacobi ordering. However, since some of such permutations are odd, one should take particular care about the direction of time flow. The details and corresponding derivations of such modifications to the original algorithm[26] are presented in Supporting Information section S2.

Finally, we tested the performance of the above algorithms (except for Omelyan) in the NVT ensemble. As Figure 4 shows,

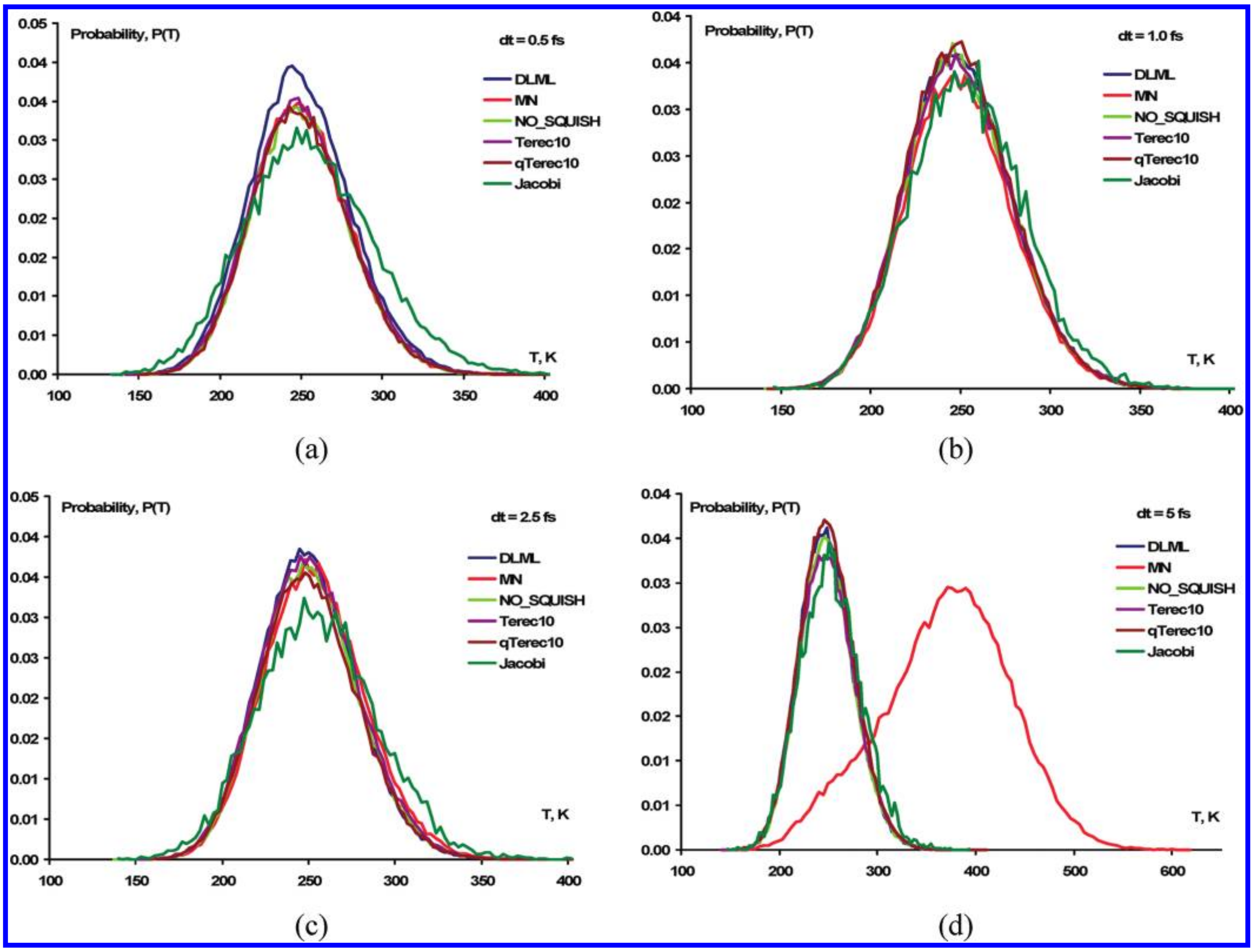

Figure 4. Temperature distribution in the NVT ensemble generated by different methods and with different integration time steps: (a) 0.5 fs, (b) 1.0 fs, (c) 2.5 fs, (d) 5.0 fs. The target temperature was set to 250 K.

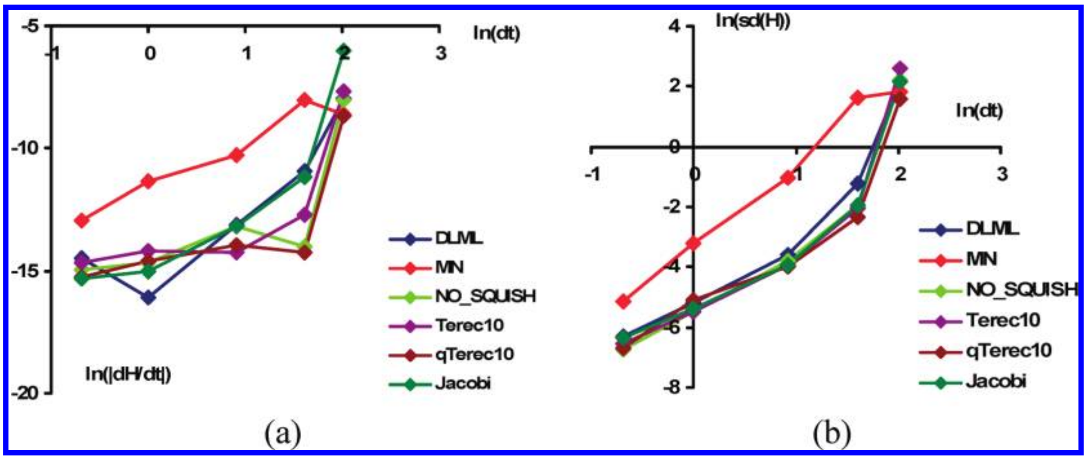

Figure 5. Comparison of Terec and qTerec methods with other existing integrators for the NVT ensemble: (a) Nose−Poincare Hamiltonian trend, stability characteristics; (b) standard deviation, accuracy characteristics.

the temperature distributions are very similar for all methods and for all time steps up to 5 fs. At *dt* = 5 fs, the MN integrator is no longer stable and does not generate the correct distribution, while other methods still perform correctly.

We also studied the stability and accuracy of the methods by examining the properties similar to those defined in eqs 22, but using Nose−Poincare Hamiltonian 11 instead of the total energy. It should be noted that the quantity (eq 11) is not only

Table 1. Run Time for a 5 ns Simulation of a Cluster of 23 Water Molecules

method	time, s
DLML	184 ± 4
MN	274± 5
NO_SQUSIH	560 ±4
Omelyan	92 ± 1
Terec10	373 ± 3
qTerec10	289 ± 3
Jacobi	1406 ± 70

the conserved quantity but is a true Hamiltonian. Thus, its trend gives information about the symplecticity of the NVT integrator, as the trend of total energy characterizes the symplecticity of the NVE integrators in the sense discussed earlier. The comparison of such quantities is presented in Figure 5.

Similarly to the NVE ensemble, one may observe two groups of methods—one is nonsymplectic (MN in this case), and the other group has all symplectic methods (DLML, NO_SQUISH, Terec10, and qTerec10). In contrast to the NVE ensemble, the properties of the integrator have more of an effect on the maximal integration time step for which the dynamics is still stable. As we can see, the instability occurs for the MN integrator already at *dt* = 5 fs, while such a time step is still acceptable for this method in the microcanonical ensemble. The Terec and qTerec methods once again show properties comparable to those of existing symplectic integrators.

Finally, in the third stage, the performances of all methods were compared to each other. To do this, we considered the same system, namely, a cluster of 23 water molecules, but without any interactions. This is necessary since the computation time of all interactions is much larger than the time required for performing an integration of equations of motion. That would obscure the actual speed of the integration algorithms. In other words, we considered a system of the 23 free rigid bodies, the motion of which is determined by the initial distribution of angular and linear momenta, constrained to correspond to a given temperature. In all cases, said temperature was set to 300 K. The trajectory time was set to 5×10^6 steps with an integration time step of 1 fs, which corresponds to 5 ns trajectories. It is important to note that for the purposes of a speed comparison, the trajectory length has no effect, so it could be chosen to produce any reasonable execution time.

The results of such a comparison are summarized in Table 1. The Terec method turns out to be even faster than the symplectic decomposition scheme NO_SQUISH. This is probably because the latter method uses many trigonometric functions for each integration time step. Both methods in turn are slower than the other decomposition schemes, including a quaternion version of the Terec method (qTerec). The latter is only slightly slower than the MN algorithm. However, the precision and stability comparisons made in previous stages make the MN algorithm less favorable than qTerec (and even Terec). The fastest Omelyan algorithm also suffers from stability and accuracy problems.

Finally, the only outlier is the Jacobi method. The execution time for this method becomes on average 5 times slower than that for most other algorithms. The Terec and qTerec methods also give a numerically exact solution (sometimes the precision and stability are even higher, see Figures 3 and 5), but for a fraction of the cost associated with the Jacobi method may thus be more attractive for some MD applications.

Table 2. Speed Comparisons for Different Taylor Series Expansion Sizes for Both Terec and qTerec Methods

expansion size	Terec, s	qTerec, s
5	221 ± 1	186 ± 4
10	373 ± 3	289 ± 3
15	616 ± 9	453 ± 4
20	937 ± 7	645 ± 3

As is expected, as the expansion size increases, so do the execution times for corresponding versions of Terec or qTerec algorithms (Table 2). We can also note that for all expansion sizes, the quaternion version is usually faster than the orientation directions counterpart. This is because the first method propagates only four quaternion components, while the other propagates three components of the three direction vectors (that is, in total nine components).

4. CONCLUSIONS

In this work, we reported new numerically exact methods to solve the FRB problem that allows us to develop new integration schemes for rigid-body MD simulations. We showed how this approach might further be used for the construction of symplectic and time-reversible integrators in both microcanonical (NVE) and canonical (NVT) ensembles. Since our approach solves the FRB up to machine precision, it may be considered as an efficient and easy-to-implement alternative for existing exact solution methods, which involve Jacobi elliptic functions. Although for big integration time steps the analytic solutions might be superior, in usual MD simulations the integration time step is limited by the highest vibration frequency in the system. As a result, in most cases, it is practically impossible to use time steps larger than 5 fs, and it significantly enhances the applicability of our methods.

We showed that our integrators have characteristics not worse (but even better in some cases) than those of existing symplectic integration schemes for all time steps up to 5 fs. However, our method differs from those schemes in that it solves the FRB problem exactly (up to machine precision), and it does not need the evaluation of the Jacobi elliptic functions. Moreover, different special cases are treated in the same way as for general asymmetric rigid bodies, which facilitates the implementation of the method in computer code.

In addition, we performed a comparative study of the existing rigid body integration schemes (integrators) focusing on their stability, precision, and performance properties. We found that the time-reversible and symplectic schemes of DLML and NO_SQUISH as well as the Jacobi method based on an analytic solution of FRB show much better properties than those methods which are not symplectic (Omelyan, MN). For our new integrator, the properties depend on the expansion length. For expansion lengths as small as five terms, the integrator based on direction vectors (Terec) shows the properties comparable to tested nonsymplectic schemes, while the quaternion-based algorithms (qTerec) show much better accuracy and stability, comparable to those of symplectic schemes. For a bigger number of expansion terms, the properties of both integrators become comparable (and even better in some cases) to those of symplectic schemes. Thus, the reported methods are effectively symplectic (which follows from the fact that the exact solution is by definition a symplectic mapping).

In terms of performance, our algorithms are much faster than the Jacobi method, providing the same and even better accuracy and stability. Moreover, our method is very robust and does not need to consider many special cases, nor deal with some internal (Jacobi) ordering. This makes them very easy to implement and efficient to run. We showed that the new methods are even faster than the existing symplectic decomposition scheme NO_SQUISH.

Although for conventional molecular dynamics the difference in performance of all algorithms is usually neglected by a significantly slower interaction calculation step, it may be more important in such methods as discrete molecular dynamics[46−49] where the interactions are calculated relatively rarely and efficiently. In such methods, it may be necessary to solve the FRB problem for relatively long times. This may not be accomplished by conventional splitting schemes, which are approximate by construction. Using the exact method (Jacobi) described by van Zon will solve the problem, but it would take approximately 5 times more time than with our algorithms.

We also showed that the new algorithms (as well as most of the existing ones) may be combined with the Nose−Poincare thermostat in a straightforward fashion. The corresponding mappings are similar to those used for the NVE ensemble and differ only in intrinsic scaling of the integration time step. We demonstrated that such coupling indeed generates correct distributions for all integrators considered. Moreover, our new integrators (Terec and qTerec) work in both NVT and NVE ensembles and show properties comparable to those of the existing symplectic schemes.

The advantage of our method becomes clear if one compares each of the integrators one by one. In some cases, our method is more stable and precise (vs Omelyan and MN); in others, it is faster (vs Jacobi and NO_SQUISH); in still others, it is capable of performing exact integration of the FRB problem for longer time steps (vs all splitting schemes); in some cases, it is more robust and easy to implement (vs Jacobi). We thus believe that these new integrators will be useful for long time scale simulations of various types of molecular systems.

ASSOCIATED CONTENT

S Supporting Information. An extension of the Jacobi method by R. van Zon is presented, giving the opportunity to perform Jacobi ordering for any combination of angular momentum components and any distinct principal components of the inertia tensor. The derivation of the corresponding transformations of the attitude matrix equation of motion is presented. This material is available free of charge via the Internet at http://pubs.acs.org.

AUTHOR INFORMATION

Corresponding Author

*E-mail: aa7@rice.edu.

ACKNOWLEDGMENT

The authors acknowledge support from the Welch Foundation (Grant C-1559) and from the U.S. National Science Foundation (Grant ECCS-0708765). This work was also supported in part by the Shared University Grid at Rice University funded by the U.S. National Science Foundation under grant EIA-0216467 and a partnership between Rice University, Sun Microsystems, and Sigma Solutions.

REFERENCES

(1) Chennamsetty, N.; Voynov, V.; Kayser, V.; Helk, B.; Trout, B. L. *Proteins: Struct., Funct., Bioinf.* **2011**, *79*, 888.

(2) Kuczera, K.; Jas, G. S.; Elber, R. *J. Phys. Chem. A* **2009**, *113*, 7461.

(3) Hall, B. A.; Sansom, M. S. P. *J. Chem. Theory Comput.* **2009**, *5*, 2465.

(4) Karplus, M.; Kuriyan, J. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 6679.

(5) Jusufi, A.; DeVane, R. H.; Shinoda, W.; Klein, M. L. *Soft Matter* **2011**, *7*, 1139.

(6) Izvekov, S.; Voth, G. A. *J. Chem. Theory Comput.* **2006**, *2*, 637.

(7) Marrink, S. J.; Risselada, H. J.; Yefimov, S.; Tieleman, D. P.; de Vries, A. H. *J. Phys. Chem. B* **2007**, *111*, 7812.

(8) Treptow, W.; Klein, M. L. *J. Am. Chem. Soc.* **2010**, *132*, 8145.

(9) Zhong, Q.; Jiang, Q.; Moore, P. B.; Newns, D. M.; Klein, M. L. *Biophys. J.* **1998**, *74*, 3.

(10) Alkis, S.; Jiang, P.; Wang, L.-L.; Roitberg, A. E.; Cheng, H.-P.; Krause, J. L. *J. Phys. Chem. C* **2007**, *111*, 14743.

(11) Hautman, J.; Klein, M. L. *J. Chem. Phys.* **1989**, *91*, 4994.

(12) Jung, H. H.; Won, Y. D.; Shin, S.; Kim, K. *Langmuir* **1999**, *15*, 1147.

(13) Sellers, H.; Ulman, A.; Shnidman, Y.; Eilers, J. E. *J. Am. Chem. Soc.* **1993**, *115*, 9389.

(14) Tierney, H. L.; Baber, A. E.; Sykes, E. C. H.; Akimov, A.; Kolomeisky, A. B. *J. Phys. Chem. C* **2009**, *113*, 10913.

(15) Akimov, A.; Kolomeisky, A. B. *J. Phys. Chem. C* **2011**, *115*, 125.

(16) Horinek, D.; Michl, J. *Proc. Natl. Acad. Sci. U. S. A.* **2005**, *102*, 14175.

(17) Clarke, L. I.; Horinek, D.; Kottas, G. S.; Varaksa, N.; Magnera, T. F.; Hinderer, T. P.; Horansky, R. D.; Michl, J.; Price, J. C. *Nanotechnology* **2002**, *13*, 533.

(18) Chen, J.; Trout, B. L. *J. Phys. Chem. B* **2010**, *114*, 13764.

(19) Amirjalayer, S.; Tafipolsky, M.; Schmid, R. *Angew. Chem., Int. Ed.* **2007**, *46*, 463.

(20) Ovchinnikov, V.; Trout, B. L.; Karplus, M. *J. Mol. Biol.* **2010**, *395*, 815.

(21) Akimov, A. V.; Nemukhin, A. V.; Moskovsky, A. A.; Kolomeisky, A. B.; Tour, J. M. *J. Chem. Theory Comput.* **2008**, *4*, 652.

(22) Kupchenko, I. V.; Moskovsky, A. A.; Nemukhin, A. V.; Kolomeisky, A. B. *J. Phys. Chem. C* **2011**, *115*, 108.

(23) Konyukhov, S. S.; Kupchenko, I. V.; Moskovsky, A. A.; Nemukhin, A. V.; Akimov, A. V.; Kolomeisky, A. B. *J. Chem. Theory Comput.* **2010**, *6*, 2581.

(24) Dullweber, A.; Leimkuhler, B.; McLachlan, R. *J. Chem. Phys.* **1997**, *107*, 5840.

(25) Kamberaj, H.; Low, R. J.; Neal, M. P. *J. Chem. Phys.* **2005**, *122*, 224114.

(26) van Zon, R.; Schofield, J. *J. Comput. Phys.* **2007**, *225*, 145.

(27) van Zon, R.; Omelyan, I. P.; Schofield, J. *J. Chem. Phys.* **2008**, *128*, 136102.

(28) Miller, T. F.; Eleftheriou, M.; Pattnaik, P.; Ndirango, A.; Newns, D.; Martyna, G. J. *J. Chem. Phys.* **2002**, *116*, 8649.

(29) Ikeguchi, M. *J. Comput. Chem.* **2004**, *25*, 529.

(30) Matubayasi, N.; Nakahara, M. *J. Chem. Phys.* **1999**, *110*, 3291.

(31) Omelyan, I. P. *Phys. Rev. E* **1998**, *58*, 1169.

(32) Okumura, H.; Itoh, S. G.; Okamoto, Y. *J. Chem. Phys.* **2007**, *126*, 084103.

(33) Celledoni, E.; Safstrom, N. *Int. J. Model., Identification Control* **2006**, *27*, 95.

(34) Celledoni, E.; Säfström, N. *J. Phys. A* **2006**, *39*, 5463.

(35) Trotter, H. F. *Proc. Am. Math. Soc.* **1959**, *10*, 545.

(36) Raedt, H. D.; Raedt, B. D. *Phys. Rev. A* **1983**, 28.

(37) Bond, S. D.; Leimkuhler, B. J.; Laird, B. B. *J. Comput. Phys.* **1999**, *151*, 114.

(38) Nose, S. *J. Phys. Soc. Jpn.* **2001**, *70*, 75.

(39) Kleinerman, D. S.; Czaplewski, C.; Liwo, A.; Scheraga, H. A. *J. Chem. Phys.* **2008**, *128*, 245103.

(40) Tuckerman, M. E.; Alejandre, J.; López-Rendón, R.; Jochim, A. L.; Martyna, G. J. *J. Phys. A* **2006**, *39*, 5629.

(41) Tuckerman, M. E.; Berne, B. J. *J. Chem. Phys.* **1991**, *95*, 8362.

(42) Tuckerman, M. E.; Liu, Y.; Ciccotti, G.; Martyna, G. J. *J. Chem. Phys.* **2001**, *115*, 1678.

(43) Tuckerman, M.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990.

(44) Leimkuhler, B.; Reich, S. *Simulating Hamiltonian Dynamics*; Cambridge University Press: Cambridge, U.K., 2004; pp 212−216.

(45) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. *J. Chem. Phys.* **1983**, *79*, 926.

(46) Alder, B. J.; Wainwright, T. E. *J. Chem. Phys.* **1959**, *31*, 459.

(47) van Zon, R.; Schofield, J. *J. Chem. Phys.* **2008**, *128*, 154119.

(48) de la Peña, H.; L. van Zon, R.; Schofield, J.; Opps, S. B. *J. Chem. Phys.* **2007**, *126*, 074105.

(49) Dokholyan, N. V.; Buldyrev, S. V.; Stanley, H. E.; Shakhnovich, E. I. *Fold Des.* **1998**, *3*, 577.