

## Fast time-reversible algorithms for molecular dynamics of rigid-body systems

Yasuhiro Kajima, Miyabi Hiyama, Shuji Ogata, Ryo Kobayashi, and Tomoyuki Tamura

Citation: *The Journal of Chemical Physics* **136**, 234105 (2012); doi: 10.1063/1.4729284

View online: <http://dx.doi.org/10.1063/1.4729284>

View Table of Contents: <http://scitation.aip.org/content/aip/journal/jcp/136/23?ver=pdfcov>

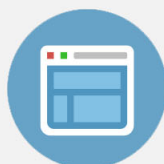
Published by the [AIP Publishing](#)

---



## Re-register for Table of Content Alerts

Create a profile.



Sign up today!



# Fast time-reversible algorithms for molecular dynamics of rigid-body systems

Yasuhiro Kajima,<sup>1,a)</sup> Miyabi Hiyama,<sup>2</sup> Shuji Ogata,<sup>1</sup> Ryo Kobayashi,<sup>1</sup>  
and Tomoyuki Tamura<sup>1</sup>

<sup>1</sup>Graduate School of Engineering, Nagoya Institute of Technology, Nagoya, Aichi 466-8555, Japan

<sup>2</sup>Graduate School of Information Science, Nagoya University, Nagoya, Aichi 464-8601, Japan

(Received 8 March 2012; accepted 30 May 2012; published online 18 June 2012)

In this paper, we present time-reversible simulation algorithms for rigid bodies in the quaternion representation. By advancing a time-reversible algorithm [Y. Kajima, M. Hiyama, S. Ogata, and T. Tamura, *J. Phys. Soc. Jpn.* **80**, 114002 (2011)] that requires iterations in calculating the angular velocity at each time step, we propose two kinds of iteration-free fast time-reversible algorithms. They are easily implemented in codes. The codes are compared with that of existing algorithms through demonstrative simulation of a nanometer-sized water droplet to find their stability of the total energy and computation speeds. © 2012 American Institute of Physics. [<http://dx.doi.org/10.1063/1.4729284>]

## I. INTRODUCTION

Molecules are often handled as rigid bodies in molecular dynamics simulation. Despite the simplification, in the case of the TIP4P potential<sup>1</sup> for H<sub>2</sub>O molecules, for instance, one can reproduce various physical properties of interest with reasonable accuracies such as the freezing and boiling conditions, the electric permittivity, and the interfacial energy of ice and water. The simplification by ignoring the fast vibration of constituting atoms of a molecule is highly effective for taking a long time step to realize a long-time simulation.<sup>2</sup> Several time-integration algorithms for rotational motion of rigid body molecules have been devised, e.g., (i) the Gear's predictor-corrector algorithm,<sup>2</sup> (ii) the Matubayasi-Nakahara's algorithm,<sup>3</sup> (iii) the symplectic algorithms,<sup>4-8</sup> (iv) the angular momentum Verlet (AMV) algorithm,<sup>9</sup> and (v) the numerically exact time-reversible (NET) algorithm.<sup>10</sup>

The algorithm (i) is very accurate for a short time step. However, it is not time reversible, and becomes unstable or the total energy increases significantly in a long-time simulation run. It is highly unstable for longer time steps. The algorithm (ii) is time reversible and non-symplectic. It shows high stability, however, it is slightly complicated with its procedure composed of several parts that use auxiliary functions. The algorithms (iii) have the feature of time reversibility in addition to the symplecticness and show very high stability; a conserved quantity that is close to Hamiltonian exists. The algorithm (iv) is easy to understand and interesting since it is constructed in an analogous manner to the velocity-Verlet algorithm. Although it is not time reversible in the strict sense, it shows smaller fluctuation in the total energy than does the leap-frog algorithm<sup>2</sup> when it is applied to the system of tetrahedral molecules. However, we find the total energy increases monotonously during the simulation runs for some systems including a water droplet. The algorithm (v) is time reversible

and is more stable than the AMV algorithm approaching to the quality of the algorithm (ii). However, it requires an iteration procedure to get the angular velocity resulting in a relatively slow simulation speed as compared to the AMV algorithm.

The aim of this paper is to propose fast time-reversible algorithms, called the FT algorithms, for rigid body molecules without such an iteration procedure by advancing the NET algorithm. We will give two FT algorithms. The difference between them lies only in the method of eliminating the iteration procedure. In the method, the three components of the angular velocity vector are treated as a set. Mutually different treatments are applied to the three components with the feature of phase-space conservation in the other method. The FT algorithms will be fast time-reversible ones for rigid molecules, since each of which consists of relatively fewer operations of basic arithmetic and square root (see Sec. III C and Table I). Simulations of the FT algorithms will show much greater stability than that of the NET algorithm, and comparable stability to that of the Matubayasi-Nakahara algorithm.<sup>3</sup>

The rest of the paper is organized as follows. In Sec. II, we will formulate the FT algorithms to propose a set of procedures for each. In Sec. III, the FT algorithms will be applied to simulate a water droplet composed of 499 H<sub>2</sub>O molecules to demonstrate their stability in the total energy and computation speeds. Section IV is devoted to summary and concluding remarks.

## II. FAST TIME-REVERSIBLE ALGORITHMS

The motion of a rigid molecule is decomposed into the translational motion and the rotational motion around the centroid. The equations of the FT algorithms for the translational motion are given in Sec. II A 1 by employing the velocity-Verlet algorithm. The velocity-Verlet algorithm is time reversible and symplectic for point particles showing high stability. The equations of the FT algorithms for the rotational

<sup>a)</sup>Also at Nagoya Zokei University, Komaki, Aichi 485-8563, Japan.  
Electronic mail: y-kajima@nifty.com.

TABLE I. Computation timings averaged over  $10^6$  measurements required to update angular velocity and angular position on various machines for various algorithms.

Algorithm	Computation timing ( $10^{-8}$ s)			Number of operations		
	Core i7 <sup>a</sup>	SPARC <sup>b</sup>	Xeon <sup>c</sup>	Four rules <sup>d</sup>	Trigonometric <sup>e</sup>	Square root
FT1	7.2	20	9.5	154	0	1
FT2	5.7	19	8.8	82	0	1
Symplectic	21	52	29	274	10	0
MN <sup>f</sup>	11	65	15	166	18	1
NET <sup>g</sup>	7.5	26	12	169	0	1

<sup>a</sup>3.1GHz Intel Core i7 950 with Intel Fortran compiler Ver. 12.0, option = “-fast.”

<sup>b</sup>2.5GHz Fujitsu SPARC64VII with Fujitsu Fortran compiler, option = “-Kfast.”

<sup>c</sup>3.0GHz Intel Xeon E5472 with Intel Fortran compiler Ver. 11.1, option = “-fast.”

<sup>d</sup>The four rules of arithmetic, i.e., addition, subtraction, multiplication (including exponentiation), and division.

<sup>e</sup>Trigonometric functions.

<sup>f</sup>Matubayasi-Nakahara algorithm.<sup>3</sup>

<sup>g</sup>NET algorithm with one time iteration.<sup>10</sup>

motion are derived in Secs. II A 2, II A 3, and II B. In Sec. II C, we show the procedure of the FT algorithm.

## A. Equations of FT algorithms

For simplicity, we present the set of equations common to two FT algorithms for a single rigid molecule, which is composed of the updates of centroid position, centroid velocity, angular position, and angular velocity. In actual simulation of a molecular system, the equations will be applied in parallel to all the rigid molecules.

### 1. Determination of centroid position and velocity

We employ the time-reversible velocity-Verlet algorithm to describe the translational motion of the centroid of a rigid molecule

$$\vec{r}(t + \Delta t) = \vec{r}(t) + \Delta t \left( \vec{v}(t) + \frac{\Delta t}{2m} \vec{f}(t) \right), \quad (1)$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\Delta t}{2m} (\vec{f}(t) + \vec{f}(t + \Delta t)). \quad (2)$$

Here, the vectors  $\vec{r}$  and  $\vec{v}$  represent the position and velocity of the centroid, respectively. The  $\vec{f}$  is the summation of the forces on the constituting atoms of the molecule, and  $m$  is the mass of the molecule. The force on each atom is assumed to be a function of the atomic positions only.

### 2. Determination of angular position

The angular position of a rigid molecule is described with the quaternion. The aim of this subsection is to derive Eq. (20) below, by which we update the angular position. We introduce a coordinate frame fixed to a rigid body molecule so that the moment of inertia tensor is diagonal; that is, the body-fixed coordinates of a point are obtained as its projections on the principal axes of inertia. We assume that the origin  $O_b$  of the body-fixed frame coincides with the centroid of the rigid molecule. Similarly, the space-fixed frame is introduced, whose origin is denoted by  $O_s$ .

Let a matrix  $\overleftrightarrow{R}_q$  rotate the three axes of the space-fixed frame to be parallel to that of the body-fixed frame. The  $\overleftrightarrow{R}_q$  is parametrized by a unit quaternion  $\vec{q} = {}^t(q_0, q_1, q_2, q_3)$  (the superscript “t” means the transpose operation) with  $|\vec{q}| = 1$

$$\overleftrightarrow{R}_q = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (3)$$

We add the superscript “(b)” or “(s)” to clarify that the associated vector is represented as a  $3 \times 1$  matrix in the body-fixed or space-fixed frame, respectively. Then, for any point  $P$ ,

$$\overrightarrow{O_b P}^{(s)} = \overleftrightarrow{R}_q \overrightarrow{O_b P}^{(b)}. \quad (4)$$

The quaternion and Euler angles<sup>11</sup> are related to each other through  $q_0 = \cos(\theta/2)\cos[(\phi + \psi)/2]$ ,  $q_1 = \sin(\theta/2)\cos$

$[(\phi - \psi)/2]$ ,  $q_2 = \sin(\theta/2)\sin[(\phi - \psi)/2]$ , and  $q_3 = \cos(\theta/2)\sin[(\phi + \psi)/2]$ . Here  $\theta$ ,  $\phi$ , and  $\psi$  are the three Euler angles of the body-fixed frame relative to the space-fixed one in the standard convention.<sup>11,12</sup> Hereafter, we exploit the unit quaternion  $\vec{q}$  exclusively to represent the rotational position of a rigid body in the space-fixed frame through Eq. (4).

Since we have assumed that the three axes of the body-fixed frame are the principal axes of inertia, we can write the

angular momentum  $\vec{L}^{(b)} = {}^t(L_x, L_y, L_z)$  as follows:

$$L_x = I_x \omega_x(t), \quad L_y = I_y \omega_y(t), \quad L_z = I_z \omega_z(t), \quad (5)$$

where  $I_x, I_y$ , and  $I_z$  are the principal moments of inertia, and  $\vec{\omega}^{(b)}(t) = {}^t(\omega_x(t), \omega_y(t), \omega_z(t))$  is the angular velocity of the rigid molecule. In the following, we will use  $\vec{\omega}(t)$  to mean  $\vec{\omega}^{(b)}(t)$ . It is known that the following identity holds:<sup>3,4</sup>

$$\frac{d}{dt} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}. \quad (6)$$

Let

$$\overleftrightarrow{A}[\vec{\omega}] = \begin{pmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{pmatrix} \quad (7)$$

and

$$\vec{q}(t) = {}^t(q_0(t), q_1(t), q_2(t), q_3(t)). \quad (8)$$

Notice here that  $\overleftrightarrow{A}[\vec{\omega}]$  above is not a product of  $\overleftrightarrow{A}$  and  $\vec{\omega}$  but a matrix of  $\vec{\omega}$ . Then Eq. (6) is rewritten as<sup>3</sup>

$$\frac{d}{dt} \vec{q}(t) = \overleftrightarrow{A} \left[ \frac{1}{2} \vec{\omega}(t) \right] \vec{q}(t), \quad (9)$$

and then

$$\begin{aligned} \frac{d^2}{dt^2} \vec{q}(t) &= \frac{d}{dt} \left( \frac{d}{dt} \vec{q}(t) \right) \\ &= \left( \overleftrightarrow{A} \left[ \frac{1}{2} \vec{\omega}(t) \right] \right)^2 \vec{q}(t) + \overleftrightarrow{A} \left[ \frac{1}{2} \frac{d}{dt} \vec{\omega}(t) \right] \vec{q}(t). \end{aligned}$$

Since  $(\overleftrightarrow{A}[\frac{1}{2}\vec{\omega}(t)])^2 = -|\frac{1}{2}\vec{\omega}(t)|^2 \overleftrightarrow{E}$  ( $\overleftrightarrow{E}$  is the identity matrix), the equation above yields

$$\frac{d^2}{dt^2} \vec{q}(t) = -\left| \frac{1}{2} \vec{\omega}(t) \right|^2 \vec{q}(t) + \overleftrightarrow{A} \left[ \frac{1}{2} \frac{d}{dt} \vec{\omega}(t) \right] \vec{q}(t). \quad (10)$$

From Eqs. (9) and (10),

$$\begin{aligned} \vec{q}(t) + \frac{d}{dt} \vec{q}(t) \Delta t + \frac{1}{2} \frac{d^2}{dt^2} \vec{q}(t) \Delta t^2 \\ = \vec{q}(t) + \overleftrightarrow{A} \left[ \frac{1}{2} \vec{\omega}(t) \right] \vec{q}(t) \Delta t \\ + \frac{1}{2} \left( -\left| \frac{1}{2} \vec{\omega}(t) \right|^2 \vec{q}(t) + \overleftrightarrow{A} \left[ \frac{1}{2} \frac{d}{dt} \vec{\omega}(t) \right] \vec{q}(t) \right) \Delta t^2 \\ = \left( 1 - \frac{1}{2} \left| \frac{1}{2} \vec{\omega}(t) \right|^2 \Delta t^2 \right) \vec{q}(t) \\ + \overleftrightarrow{A} \left[ \frac{1}{2} \left( \vec{\omega}(t) + \frac{1}{2} \frac{d}{dt} \vec{\omega}(t) \Delta t \right) \right] \vec{q}(t), \quad (11) \end{aligned}$$

where we have used the fact that  $\overleftrightarrow{A}[\vec{\omega}]$  is linear with respect to  $\vec{\omega}$ . Let us define

$$\begin{aligned} s &= 1 - \frac{1}{2} \left| \frac{1}{2} \vec{\omega}(t) \right|^2 \Delta t^2 \quad \text{and} \quad \overleftrightarrow{V} \\ &= \frac{1}{2} \left( \vec{\omega}(t) + \frac{1}{2} \frac{d}{dt} \vec{\omega}(t) \Delta t \right) \Delta t. \quad (12) \end{aligned}$$

Then, Eq. (11) is equal to  $(s \overleftrightarrow{E} + \overleftrightarrow{A}[\overleftrightarrow{V}]) \vec{q}(t)$ , and thus

$$\begin{aligned} \vec{q}(t + \Delta t) &= \vec{q}(t) + \frac{d}{dt} \vec{q}(t) \Delta t + \frac{1}{2} \frac{d^2}{dt^2} \vec{q}(t) \Delta t^2 + O(\Delta t^3) \\ &= (s \overleftrightarrow{E} + \overleftrightarrow{A}[\overleftrightarrow{V}]) \vec{q}(t) + O(\Delta t^3). \quad (13) \end{aligned}$$

Since  $s > 0$  and  $|\overleftrightarrow{V}| < 1$  for usual time steps,<sup>13</sup> it follows from Eq. (13) that

$$\vec{q}(t + \Delta t) = \left( \sqrt{1 - |\overleftrightarrow{V}|^2} \overleftrightarrow{E} + \overleftrightarrow{A}[\overleftrightarrow{V}] \right) \vec{q}(t) + O(\Delta t^3). \quad (14)$$

Here, we have used the fact

$$\begin{aligned} s &= 1 - \frac{1}{2} \left| \frac{1}{2} \vec{\omega}(t) \right|^2 \Delta t^2 \\ &= 1 - \frac{1}{2} \left| \frac{1}{2} \vec{\omega}(t) + \frac{1}{4} \frac{d}{dt} \vec{\omega}(t) \Delta t \right|^2 \Delta t^2 + O(\Delta t^3) \\ &= 1 - \frac{1}{2} |\overleftrightarrow{V}|^2 + O(\Delta t^3) = \sqrt{1 - |\overleftrightarrow{V}|^2} + O(\Delta t^3). \end{aligned}$$

Note that  $\sqrt{1 - |\overleftrightarrow{V}|^2} \overleftrightarrow{E} + \overleftrightarrow{A}[\overleftrightarrow{V}]$  in Eq. (14) is an orthogonal matrix, which conserves the distance or the norm of the quaternion.

We define the orthogonal matrix  $\overleftrightarrow{R}[\vec{v}]$  for a vector  $\vec{v} = {}^t(v_x, v_y, v_z)$  ( $|\vec{v}| < 1$ ) as

$$\begin{aligned} \overleftrightarrow{R}[\vec{v}] &= \sqrt{1 - |\vec{v}|^2} \overleftrightarrow{E} + \overleftrightarrow{A}[\vec{v}] \\ &= \begin{pmatrix} \sqrt{1 - |\vec{v}|^2} & -v_x & -v_y & -v_z \\ v_x & \sqrt{1 - |\vec{v}|^2} & v_z & -v_y \\ v_y & -v_z & \sqrt{1 - |\vec{v}|^2} & v_x \\ v_z & v_y & -v_x & \sqrt{1 - |\vec{v}|^2} \end{pmatrix}. \quad (15) \end{aligned}$$

It is easy to see that  $\overleftrightarrow{R}[\vec{v}]$  satisfies

$$\overleftrightarrow{R}[\vec{v}] \overleftrightarrow{R}[-\vec{v}] = \overleftrightarrow{E}. \quad (16)$$

Using the notations above, we rewrite Eq. (14) as

$$\vec{q}(t + \Delta t) = \overleftrightarrow{R}[\overleftrightarrow{V}] \vec{q}(t) + O(\Delta t^3). \quad (17)$$

Let us define

$$\vec{\phi}(t, \Delta t) = \vec{\omega}(t) + \frac{1}{2} \frac{d}{dt} \vec{\omega}(t) \Delta t. \quad (18)$$

Then we have  $\overleftrightarrow{V} = \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t$  and Eq. (17) yields

$$\vec{q}(t + \Delta t) = \overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t \right] \vec{q}(t) + O(\Delta t^3). \quad (19)$$

From Eq. (19), we obtain an equation to update the quaternion  $\vec{q}$  as

$$\vec{q}(t + \Delta t) = \overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t \right] \vec{q}(t). \quad (20)$$

Here,  $\overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t \right]$  is an orthogonal matrix (i.e., distance conserving), and hence  $|\vec{q}(t + \Delta t)| = |\vec{q}(t)|$ . Consequently, the  $\vec{q}$  is normalized automatically for any  $\Delta t$ .

There remains the task of calculating  $\vec{\phi}(t, \Delta t)$ . From Eq. (18),

$$\begin{aligned} \phi_x(t, \Delta t) &= \omega_x(t) + \frac{1}{2} \frac{d\omega_x(t)}{dt} \Delta t, \\ \phi_y(t, \Delta t) &= \omega_y(t) + \frac{1}{2} \frac{d\omega_y(t)}{dt} \Delta t, \\ \phi_z(t, \Delta t) &= \omega_z(t) + \frac{1}{2} \frac{d\omega_z(t)}{dt} \Delta t. \end{aligned} \quad (21)$$

Here, the derivatives of the angular velocity are given by the following Euler's equation of motion:

$$\frac{d\omega_i}{dt} = \frac{I_j - I_k}{I_i} \omega_j \omega_k + \frac{t_i}{I_i}, \quad (22)$$

where  $(i, j, k) = (x, y, z), (y, z, x),$  and  $(z, x, y)$ , and  $t_i$  is the component of the torque  $\vec{\tau}^{(b)} = (t_x, t_y, t_z)$ . Then we have

$$\begin{aligned} \phi_x(t, \Delta t) &= \omega_x(t) + (\alpha \omega_y \omega_z + \delta t_x) \Delta t, \\ \phi_y(t, \Delta t) &= \omega_y(t) + (\beta \omega_x \omega_z + \lambda t_y) \Delta t, \\ \phi_z(t, \Delta t) &= \omega_z(t) + (\gamma \omega_x \omega_y + \mu t_z) \Delta t, \end{aligned} \quad (23)$$

where

$$\begin{aligned} \alpha &= \frac{I_y - I_z}{2I_x}, \quad \beta = \frac{I_z - I_x}{2I_y}, \quad \gamma = \frac{I_x - I_y}{2I_z}, \\ \delta &= \frac{1}{2I_x}, \quad \lambda = \frac{1}{2I_y}, \quad \text{and} \quad \mu = \frac{1}{2I_z}. \end{aligned} \quad (24)$$

We can calculate  $\vec{\phi}(t, \Delta t)$  with Eq. (23), by which we update the angular position of the rigid molecule with Eq. (20).

Note that the equations of  $\vec{\phi} = (\phi_x, \phi_y, \phi_z)$  in Eq. (23) will be changed slightly later within the order of  $\Delta t^2$  to avoid an iteration procedure in determining angular velocity. Even in that case, the order of error of updated quaternion remains within  $\Delta t^3$  (see Sec. II B).

We remark that  $\vec{\phi}(t, \Delta t)$  introduced in Eq. (18) can be regarded as the angular velocity at the midstep  $t + \frac{1}{2} \Delta t$  in Ref. 3, and that we can derive Eq. (20) also by using the equations in that reference. However, our derivation above will be helpful to clarify our method.

### 3. Determination of angular velocity

The principal aim of this subsection is to derive Eq. (29) below that the updated angular velocity should obey. We determine the updated angular velocity in the same way as the NET algorithm, that is, we determine it *so as to* satisfy the time-reversibility condition.<sup>10</sup> From Eq. (20), the time-reversibility condition gives

$$\overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t + \Delta t, -\Delta t) (-\Delta t) \right] \vec{q}(t + \Delta t) = \vec{q}(t). \quad (25)$$

Combining Eqs. (20) and (25), we have

$$\overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t + \Delta t, -\Delta t) (-\Delta t) \right] \overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t \right] \vec{q}(t) = \vec{q}(t). \quad (26)$$

Therefore, we determine  $\vec{\phi}(t + \Delta t, -\Delta t)$  by the following equation:

$$\overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t + \Delta t, -\Delta t) (-\Delta t) \right] \overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t \right] = \overleftrightarrow{E}. \quad (27)$$

Since

$$\overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) (-\Delta t) \right] \overleftrightarrow{R} \left[ \frac{1}{2} \vec{\phi}(t, \Delta t) \Delta t \right] = \overleftrightarrow{E} \quad (28)$$

from Eq. (16), we find

$$\vec{\phi}(t + \Delta t, -\Delta t) = \vec{\phi}(t, \Delta t) \quad (29)$$

from Eq. (27). Using Eqs. (23) and (29), we obtain the following equations:

$$\begin{aligned} \omega_x^+ - (\alpha \omega_y^+ \omega_z^+ + \delta t_x^+) \Delta t &= \omega_x + (\alpha \omega_y \omega_z + \delta t_x) \Delta t, \\ \omega_y^+ - (\beta \omega_x^+ \omega_z^+ + \lambda t_y^+) \Delta t &= \omega_y + (\beta \omega_x \omega_z + \lambda t_y) \Delta t, \\ \omega_z^+ - (\gamma \omega_x^+ \omega_y^+ + \mu t_z^+) \Delta t &= \omega_z + (\gamma \omega_x \omega_y + \mu t_z) \Delta t, \end{aligned} \quad (30)$$

where we mean  $\omega_i = \omega_i(t)$ ,  $\omega_i^+ = \omega_i(t + \Delta t)$ ,  $t_i = t_i(t)$ , and  $t_i^+ = t_i(t + \Delta t)$  for  $i = \{x, y, z\}$ . Equations (30) contain unknown variables  $\omega_x^+$ ,  $\omega_y^+$ , and  $\omega_z^+$ . If we combine these three equations to obtain a single equation of a single variable, the degree of the single variable equation is at least 5. It is difficult to solve it algebraically. In the NET algorithm, we solve similar equations numerically for every time step by an iteration method.

In Ref. 10, we derived the equations similar in meaning to Eqs. (20), (23), and (29). However, those equations in Ref. 10 were formulated suitable for iteration procedure. We cannot apply the following iteration-free method directly to those equations. Therefore, we have derived essential equations in the present subsection.

### B. Ideas to remove iteration procedure

We propose two methods to determine the updated angular velocity without iteration. As mentioned before, we modify Eqs. (23) slightly within the order of  $\Delta t^2$ . We use the modified equations to evolve quaternion in time by Eq. (20) and to get the updated angular velocity by setting them into Eq. (29). Hereafter, we abbreviate  $\phi_x(t, \Delta t)$ ,  $\phi_y(t, \Delta t)$ ,  $\phi_z(t, \Delta t)$ ,  $\phi_x(t + \Delta t, -\Delta t)$ ,  $\phi_y(t + \Delta t, -\Delta t)$ ,  $\phi_z(t + \Delta t, -\Delta t)$ ,  $\omega_x(t)$ ,  $\omega_y(t)$ ,  $\omega_z(t)$ ,  $\omega_x(t + \Delta t)$ ,  $\omega_y(t + \Delta t)$ , and  $\omega_z(t + \Delta t)$  as  $\phi_x$ ,  $\phi_y$ ,  $\phi_z$ ,  $\phi_x^+$ ,  $\phi_y^+$ ,  $\phi_z^+$ ,  $\omega_x$ ,  $\omega_y$ ,  $\omega_z$ ,  $\omega_x^+$ ,  $\omega_y^+$ , and  $\omega_z^+$ , respectively.

### 1. Method 1: Modification to linear equations

Instead of Eqs. (23), we redefine  $\vec{\phi}(t, \Delta t) = {}^l(\phi_x, \phi_y, \phi_z)$  by the following combined equations:

$$\tilde{\omega}_x = \omega_x + \delta t_x \Delta t, \quad \tilde{\omega}_y = \omega_y + \lambda t_y \Delta t, \quad \tilde{\omega}_z = \omega_z + \mu t_z \Delta t \quad (31)$$

and

$$\begin{aligned} \phi_x &= \tilde{\omega}_x + \alpha \tilde{\omega}_z \phi_y \Delta t, \quad \phi_y = \tilde{\omega}_y + \beta \tilde{\omega}_x \phi_z \Delta t, \quad \phi_z \\ &= \tilde{\omega}_z + \gamma \tilde{\omega}_y \phi_x \Delta t. \end{aligned} \quad (32)$$

We solve these linear equations (Eqs. (32)) to get  $\vec{\phi}(t, \Delta t)$ . The so defined  $\phi_x$ ,  $\phi_y$ , and  $\phi_z$  differ from the original ones within the order of  $\Delta t^2$ , which is easy to see by solving Eqs. (31) and (32). For example,  $\phi_x = (\tilde{\omega}_x + \alpha \tilde{\omega}_y \tilde{\omega}_z \Delta t + \alpha \beta \tilde{\omega}_x \tilde{\omega}_z^2 \Delta t^2) / (1 - \alpha \beta \gamma \tilde{\omega}_x \tilde{\omega}_y \tilde{\omega}_z \Delta t^3) = \omega_x + (\alpha \omega_y \omega_z + \delta t_x) \Delta t + O(\Delta t^2)$ . Therefore, the order of error of the updated quaternion given by Eq. (20) for  $\vec{\phi} = {}^l(\phi_x, \phi_y, \phi_z)$  defined above remains within  $\Delta t^3$ .

Then,  $\vec{\phi}(t + \Delta t, -\Delta t) = {}^l(\phi_x^+, \phi_y^+, \phi_z^+)$  is given similarly by

$$\begin{aligned} \tilde{\omega}_x^+ &= \omega_x^+ - \delta t_x^+ \Delta t, \quad \tilde{\omega}_y^+ = \omega_y^+ - \lambda t_y^+ \Delta t, \quad \tilde{\omega}_z^+ \\ &= \omega_z^+ - \mu t_z^+ \Delta t \end{aligned} \quad (33)$$

and

$$\begin{aligned} \phi_x^+ &= \tilde{\omega}_x^+ - \alpha \tilde{\omega}_z^+ \phi_y^+ \Delta t, \quad \phi_y^+ = \tilde{\omega}_y^+ - \beta \tilde{\omega}_x^+ \phi_z^+ \Delta t, \quad \phi_z^+ \\ &= \tilde{\omega}_z^+ - \gamma \tilde{\omega}_y^+ \phi_x^+ \Delta t. \end{aligned} \quad (34)$$

Since  $\phi_x^+ = \phi_x$ ,  $\phi_y^+ = \phi_y$ , and  $\phi_z^+ = \phi_z$  from Eq. (29), we can calculate the updated angular velocity  ${}^l(\omega_x^+, \omega_y^+, \omega_z^+)$  by considering the reverse order operation of Eqs. (33) and (34). Equations (34) are linear in  $\tilde{\omega}_x^+$ ,  $\tilde{\omega}_y^+$ , and  $\tilde{\omega}_z^+$ , which are easy to solve. Then we get the updated angular velocity  ${}^l(\omega_x^+, \omega_y^+, \omega_z^+)$  from Eqs. (33).

Method 1 requires to solve three linear equations for three variables twice (one for Eq. (32) and the other for Eq. (34) to get the updated angular velocity. In Subsection II B 2, a method consisting of substitutions only but losing symmetry with respect to the three components in Eqs. (23) will be proposed.

### 2. Method 2: Successive substitution

Instead of Eqs. (23), we redefine  $\vec{\phi}(t, \Delta t) = {}^l(\phi_x, \phi_y, \phi_z)$  successively as follows. The arrow that is facing left in an equation below stands for substitution in a computer code. We proceed in the sequence

$$\phi_x(1) \leftarrow \omega_x, \quad \phi_y(1) \leftarrow \omega_y, \quad \phi_z(1) \leftarrow \omega_z, \quad (35)$$

$$\begin{aligned} \phi_x(2) &\leftarrow \phi_x(1) + \delta t_x \Delta t, \quad \phi_y(2) \leftarrow \phi_y(1) + \lambda t_y \Delta t, \\ \phi_z(2) &\leftarrow \phi_z(1) + \mu t_z \Delta t, \end{aligned} \quad (36)$$

$$\phi_x \leftarrow \phi_x(2) + \alpha \phi_y(2) \phi_z(2) \Delta t, \quad (37a)$$

$$\phi_y \leftarrow \phi_y(2) + \beta \phi_x \phi_z(2) \Delta t, \quad (37b)$$

$$\phi_z \leftarrow \phi_z(2) + \gamma \phi_x \phi_y \Delta t. \quad (37c)$$

Here,  $\phi_x$ ,  $\phi_y$ , and  $\phi_z$  defined in Eqs. (37a)–(37c) differ from the original ones within the order of  $\Delta t^2$ . Thus, the order of error of the updated quaternion remains within  $\Delta t^3$  as before.

Then,  $\vec{\phi}(t + \Delta t, -\Delta t) = {}^l(\phi_x^+, \phi_y^+, \phi_z^+)$  is given similarly as above, and we can solve it to get the updated angular velocity  ${}^l(\omega_x^+, \omega_y^+, \omega_z^+)$  as follows:

$$\phi_z^+(2) \leftarrow \phi_z^+ + \gamma \phi_x^+ \phi_y^+ \Delta t, \quad (38a)$$

$$\phi_y^+(2) \leftarrow \phi_y^+ + \beta \phi_x^+ \phi_z^+(2) \Delta t, \quad (38b)$$

$$\phi_x^+(2) \leftarrow \phi_x^+ + \alpha \phi_y^+(2) \phi_z^+(2) \Delta t, \quad (38c)$$

$$\phi_x^+(1) \leftarrow \phi_x^+(2) + \delta t_x^+ \Delta t, \quad \phi_y^+(1) \leftarrow \phi_y^+(2) + \lambda t_y^+ \Delta t,$$

$$\phi_z^+(1) \leftarrow \phi_z^+(2) + \mu t_z^+ \Delta t, \quad (39)$$

$$\omega_x^+ \leftarrow \phi_x^+(1), \quad \omega_y^+ \leftarrow \phi_y^+(1), \quad \omega_z^+ \leftarrow \phi_z^+(1). \quad (40)$$

For Eqs. (38)  $\phi_i^+ = \phi_i$  ( $i = \{x, y, z\}$ ) from Eq. (29). Note that Eqs. (38) are arranged in the direction opposite to Eqs. (37) so that the time reversibility holds.

We denote the FT algorithms with Methods 1 and 2 as FT1 and FT2 algorithms, respectively. Note that the FT2 algorithm satisfies the following equation of phase-space conservation (see Eq. (14) in Ref. 3):

$$\left| \frac{\partial(\vec{r}(t + \Delta t), \vec{v}(t + \Delta t), \vec{q}(t + \Delta t), \vec{\omega}(t + \Delta t))}{\partial(\vec{r}(t), \vec{v}(t), \vec{q}(t), \vec{\omega}(t))} \right| = 1. \quad (41)$$

### C. FT algorithm for single time step

We describe here the procedure for a single time step of the FT algorithm, especially for FT2, for a single rigid molecule composed of atoms. In actual simulation of a molecular system, the algorithm will be applied in parallel to all the rigid molecules. The procedure of the FT1 algorithm is similar.

The procedure of the FT2 algorithm consists of the following eight steps. They are the steps required to evaluate  $\vec{r}$ ,  $\vec{v}$ ,  $\vec{q}$ , and  $\vec{\omega}$  at time  $t + \Delta t$  from that at time  $t$ . Step 4 is devoted to the calculation of the quaternion and step 8 the angular velocity. These two steps distinguish the FT algorithm from the other ones. The other steps are generally installed in every algorithm for the motion of rigid molecule in the quaternion representation.

*Setting:* The body-fixed frame is introduced with its axes corresponding to the principal axes of inertia of a rigid molecule, whose origin coincides with the centroid of the rigid molecule. We denote its principal moments of inertia by  $I_x$ ,  $I_y$ , and  $I_z$ . Here, we use the constants  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $\delta$ ,  $\lambda$ , and  $\mu$  defined in Eqs. (24). Take the data at time  $t$  of the molecule:

the position of the centroid  $\vec{r}(t)^{(s)}$ , the velocity of the centroid  $\vec{v}(t)^{(s)}$ , the quaternion of the rigid molecule  $\vec{q}(t)$ , and the angular velocity of the rigid molecule  $\vec{\omega}(t)^{(b)} = {}^t(\omega_x, \omega_y, \omega_z)$ .

*Step 1:* Calculate the atomic positions of the molecule in the space-fixed frame by Eqs. (3) and (4).

*Step 2:* Calculate the forces on the atoms using the atomic positions calculated in step 1. Then calculate the force  $\vec{f}(t)^{(s)}$  acting on the centroid, and the torque  $\vec{\tau}(t)^{(b)} = {}^t(t_x, t_y, t_z)$ .

*Step 3:* Determine the updated position of the centroid by Eq. (1).

*Step 4:* Determine  $\vec{\phi}(t, \Delta t) = {}^t(\phi_x(t, \Delta t), \phi_y(t, \Delta t), \phi_z(t, \Delta t))$  by Eqs. (35)–(37). Set it into Eq. (20) to determine the updated quaternion.

*Step 5:* Calculate the updated positions of the atoms in the space-fixed frame by Eq. (4). (In actual simulation, we install the procedure to normalize the quaternion every thousand steps to avoid numerical error. It is not always necessary.)

*Step 6:* Calculate the updated forces on the atoms. Then obtain the torque  $\vec{\tau}^+(t) = {}^t(t_x^+, t_y^+, t_z^+)$  and the force  $\vec{f}(t + \Delta t)$  acting on the centroid.

*Step 7:* Determine the updated velocity of the centroid by Eq. (2).

*Step 8:* Determine the updated angular velocity,  $\omega_x^+$ ,  $\omega_y^+$ , and  $\omega_z^+$ , by Eqs. (38)–(40) after setting  $\phi_x^+ = \phi_x$ ,  $\phi_y^+ = \phi_y$ , and  $\phi_z^+ = \phi_z$  where  $\phi_x$ ,  $\phi_y$ , and  $\phi_z$  are obtained in step 4 and  $t_x^+$ ,  $t_y^+$ , and  $t_z^+$  in step 6.

Note that if we choose to use the FT1 algorithm, we have only to replace the equations in steps 4 and 8 by Eqs. (31) and (32) and Eqs. (33) and (34), respectively. The FT algorithms can also be applied to linear molecules. For a linear molecule, we set z axis of the body-fixed frame along the molecule. Then we have  $I_x = I_y > 0$  and  $I_z = 0$ . Here, we set  $\omega_3 = 0$ ,  $\alpha = \frac{1}{2}$ ,  $\delta = 1/I_x$ ,  $\beta = -\frac{1}{2}$ ,  $\lambda = 1/I_y$ ,  $\gamma = 0$ ,  $\mu = 0$ , and  $t_z = 0$ . Under such a setting, the procedure explained above can be used for linear molecules.

### III. COMPUTATION RESULTS: COMPARISON AMONG FT1, FT2, SYMPLECTIC, AND NET ALGORITHMS

The FT algorithms are applied to a water droplet in vacuum. We choose to use the TIP4P (Ref. 1) inter-molecular potential, in which a H<sub>2</sub>O molecule is described as a rigid, planar four charged points. It is known that various physical properties<sup>1,14</sup> are reproduced well in both liquid and crystalline phases with the TIP4P potential. Our purposes of the present application are to examine the stability and the computation timings of the FT algorithms in realistic settings through comparison of that of existing algorithms.

To prepare a water droplet, we first cut, from crystalline ice in Ih-phase,<sup>15</sup> a collection of 499 molecules in spherical shape. Second, we keep the temperature of the system at around  $T = 300$  K by controlling both translational and angular velocities for more than 1.0 ns to obtain a water droplet with a diameter of about 3.0 nm in vacuum at thermal equilibrium; no molecule is detached from the droplet. This preparation simulation is performed with  $\Delta t = 2.5$  fs using either FT, symplectic, or NET algorithm to obtain three initial configurations for each algorithm.

For precise comparison, we follow the method in Ref. 3 and introduce the local error  $\epsilon(1)$  and the global error  $\tilde{\epsilon}$  (see, below). We define

$$\epsilon(n) = \left\langle \left| \frac{E(i+n)}{E(i)} - 1 \right| \right\rangle, \quad (42)$$

where  $E(i)$  is the total energy of the system at step- $i$  and the average  $\langle \dots \rangle$  is taken over all possible  $i$  and three runs of  $10^5$  steps starting from different configurations. The  $\epsilon(1)$  gives the relative error of the total energy after  $\Delta t$ . The global error is defined as  $\tilde{\epsilon} = \lim_{n \rightarrow \infty} \epsilon(n)/n$ . To reduce the numerical fluctuation, we, in practice, calculate

$$\tilde{\epsilon} = \frac{\epsilon(10000) - \epsilon(1000)}{9000}. \quad (43)$$

In this section, the symplectic algorithm is coded according to Ref. 8 after removing unnecessary parts as the thermostat part, and the quaternions are normalized every thousand steps to avoid numerical error as in the case of the FT algorithm.

#### A. Stability of FT1 algorithm

Figure 1 shows the local error,  $\epsilon(1)$ , and the global error,  $\tilde{\epsilon}$ , for the FT1, symplectic, and NET algorithms with various  $\Delta t$ .<sup>16</sup> It is seen in Fig. 1 that  $\log \epsilon(1)$  grows linearly with  $\log \Delta t$  with the slope of approximately 3 in the three algorithms. It reflects the fact that the algorithms contain the local errors of order  $\Delta t^3$ . We find in Fig. 1 that the global error,  $\tilde{\epsilon}$ , in the FT1 algorithm is about 10% of that in the NET algorithm and is intermediate of those in the NET and symplectic algorithms.

The global errors in the symplectic algorithm increase significantly for  $\Delta t \geq 6$  fs approaching to the FT1 results as seen in Fig. 1. We do not know the reason of this, but we think that the trajectory of the conserved quantity  $\tilde{H}$  of Miller's

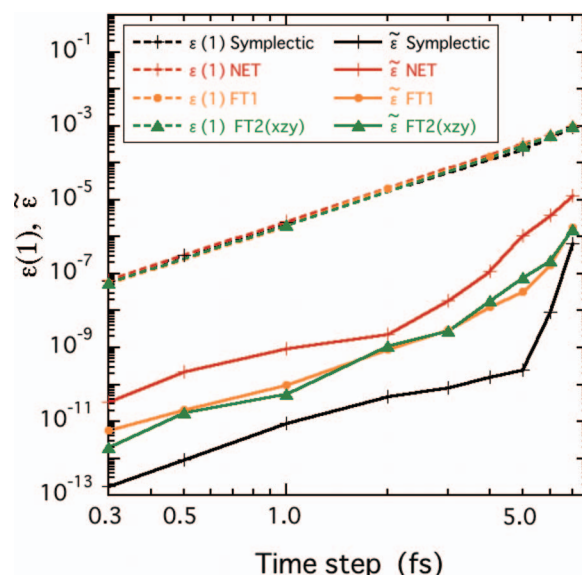


FIG. 1. Local errors,  $\epsilon(1)$ , and global errors,  $\tilde{\epsilon}$ , in the water droplet simulation at  $T = 300$  K in the symplectic, FT1, FT2 (xzy), and NET algorithms for various values of  $\Delta t$ .

algorithm assured by the symplectic method may not be contained in a restricted area of  $(q, p)$  phase space or may not converge (since  $\tilde{H}$  is a power series with respect to  $\Delta t$ ) for some large  $\Delta t$ . If it is the case, there may be a possibility that our method works better than symplectic. However, we cannot find such a time step as our method works better than symplectic. The global errors of the symplectic algorithm do not exceed that of the FT algorithms even in  $\Delta t \geq 6$  fs. More exactly, in the cases where  $\Delta t \geq 8$  fs both FTs and symplectic become unstable, and it is difficult to compare the stability of the algorithms precisely.

We note that there exist other possibilities of modifying Eqs. (23) similar to Eqs. (32). One is the way to use the equations below instead of Eqs. (32)

$$\begin{aligned}\phi_x &= \tilde{\omega}_x + \alpha\phi_z\tilde{\omega}_y\Delta t, & \phi_y &= \tilde{\omega}_y + \beta\phi_x\tilde{\omega}_z\Delta t, \\ \phi_z &= \tilde{\omega}_z + \gamma\phi_y\tilde{\omega}_x\Delta t.\end{aligned}\quad (44)$$

The other is to take the mean of Eqs. (32) and (44)

$$\begin{aligned}\phi_x &= \tilde{\omega}_x + \frac{1}{2}\alpha(\tilde{\omega}_z\phi_y + \phi_z\tilde{\omega}_y)\Delta t, \\ \phi_y &= \tilde{\omega}_y + \frac{1}{2}\beta(\tilde{\omega}_x\phi_z + \phi_x\tilde{\omega}_z)\Delta t, \\ \phi_z &= \tilde{\omega}_z + \frac{1}{2}\gamma(\tilde{\omega}_y\phi_x + \phi_y\tilde{\omega}_x)\Delta t.\end{aligned}\quad (45)$$

These two algorithms give quite similar results for  $\epsilon(1)$  and  $\tilde{\epsilon}$  to that of the FT1 algorithm.

## B. Stability of FT2 algorithm

Here, we compare both local and global errors of the FT2 algorithm with that of other algorithms. In Sec. II B 2, we have obtained  $\phi_x$ ,  $\phi_y$ , and  $\phi_z$  by substituting successively as Eqs. (35)–(37). However, the sequential order of Eqs. (37a)–(37c) can be different. There are  $3! = 6$  ways of permuting Eqs. (37a)–(37c). We perform the simulations for all six cases, to find no significant differences. Here, we show three simulations of the six cases. For the case denoted as  $(xyz)$ , the order is Eqs. (37a), (37b), and (37c). For  $(yzx)$ , the order is Eqs. (37b), (37c), and (37a). For  $(xzy)$ , the order is Eqs. (37a), (37c), and (37b). In each case, Eqs. (37a)–(37c) are arranged in the direction opposite to Eqs. (37a)–(37c). In the present setting, the magnitudes of the principal moment of inertia are  $I_x > I_z > I_y$ .

Figure 2 shows the local error,  $\epsilon(1)$ , and the global error,  $\tilde{\epsilon}$ , of the simulations with the FT2 and NET algorithms for various  $\Delta t$ .<sup>16</sup> We find in Fig. 2 that the three global errors,  $\tilde{\epsilon}(xyz)$ ,  $\tilde{\epsilon}(yzx)$ , and  $\tilde{\epsilon}(xzy)$ , in the FT2 algorithm are only about 10% of that in the NET algorithm.

Figure 1 also shows that the global errors of the FT1 and FT2 (we compare here especially FT2( $xzy$ )) algorithms. The global errors are almost the same between the two algorithms and are inferior to the symplectic algorithm by about an order of magnitude.

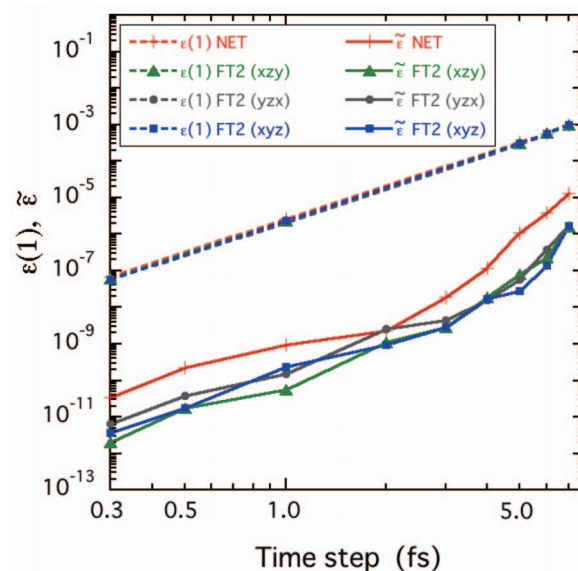


FIG. 2. Local errors,  $\epsilon(1)$ , and global errors,  $\tilde{\epsilon}$ , in the water droplet simulation at  $T = 300$  K in the FT2 and NET algorithms for various values of  $\Delta t$ .

## C. Computation timings of FT algorithms

In this subsection, we discuss computation timings of various time-reversible algorithms of rotational motion that use quaternion to represent angular position. To compare computation timings of such algorithms, we compare the required timings of the parts that update quaternion and angular velocity of a rigid molecule. We assume the remaining part is almost same in every algorithm using quaternion representation.

Table I shows computation timings averaged over  $10^6$  measurements and numbers of operations<sup>17</sup> of the FT1, FT2, symplectic, Matubayasi-Nakahara, and NET algorithms required for updating angular velocity and quaternion. The number of iteration in the NET algorithm is set to one, which is not realistic but gives us the lower limit of its computation timings. Computation timings are measured on 3.1 GHz Intel Core i7, 2.5 GHz Fujitsu SPARC64VII, and 3.0 GHz Intel Xeon E5472. The FT2 algorithm requires less than 100 operations and all of the operations are four basic arithmetic operations except for one square root operation.

As seen in Table I, computation timings depend on machines and the order can be reversed. However, we see that the computation timing for updating angular velocity and quaternion of the FT2 algorithm is no more than that of the FT1, symplectic, Matubayasi-Nakahara, and NET algorithms. Thus, assuming the remaining part of algorithm is almost same for every time-reversible one, we think that the FT2 algorithm is a fast algorithm in the time-reversible ones using quaternion representation.

## IV. SUMMARY AND CONCLUDING REMARKS

We have proposed the fast time-reversible (FT1 and FT2) simulation algorithms for rigid molecules. The stability of the FT algorithms for various values of the time step is compared



with that of the NET and symplectic algorithms, through demonstrative simulation of a water droplet composed of 499 molecules at 300 K.

The global errors of the FT algorithms are only about 10% of that of the NET algorithm. We think that the smallness is caused by various factors including the elimination of accumulation error in the iteration. The local and global errors of the FT algorithms are almost the same as that of Matubayasi-Nakahara<sup>3</sup> algorithm.

It may seem that the FT algorithms are similar to Matubayasi-Nakahara algorithm. However, there are differences between the two algorithms. The main difference lies in the choice of equations to be modified in order to get the updated angular velocity. To get the updated angular velocity, Matubayasi and Nakahara modified *differential equations* for the angular velocity, and got several differential equations. They integrated them one by one, determined the angular velocity at the midstep  $t + \frac{1}{2}\Delta t$ , integrated these equations in reverse order, and then determined the angular velocity at time  $t + \Delta t$ . The error involved in the algorithm is  $\Delta t^3$ . In the FT algorithms, on the other hand, we have introduced temporally (algebraic) equations by which we update the quaternion within the error of  $\Delta t^3$ , and have modified these *equations* within  $\Delta t^3$ . Then, we have solved these equations conversely to get the angular velocity at time  $t + \Delta t$ .

Noteworthy features of the FT algorithms are the following: (i) The FT2 algorithm will be a fast time-reversible algorithm for rotational motion comparable to other fast time-reversible ones using quaternion. (ii) From the viewpoint of total energy conservation, the FT algorithms are superior to the NET algorithm, and are comparable to the Matubayasi-Nakahara algorithm. (iii) Even if the equations of dynamics involve the friction term or an external force field, the FT algorithms are useful by interpreting the force appropriately for negative time step  $-\Delta t$ . (iv) The FT2 algorithm satisfies Eq. (41) as the Matubayasi-Nakahara algorithm does.

## ACKNOWLEDGMENTS

We are grateful for Professor Matubayasi for his valuable comments. This research was supported by Grant-in-Aid for

Scientific Research (Kakenhi: 23310074) of Japan Society for the Promotion of Science (JSPS). The timing test was carried out in part on Fujitsu FX1 at Information Technology Center at Nagoya University.

- <sup>1</sup>A. Leach, *Molecular Modelling: Principles and Applications*, 2nd ed. (Prentice Hall, New Jersey, 2001).
- <sup>2</sup>M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Oxford University Press, Oxford, 1987).
- <sup>3</sup>N. Matubayasi and M. Nakahara, *J. Chem. Phys.* **110**, 3291 (1999).
- <sup>4</sup>T. F. Miller III, M. Eleftheriou, P. Pattnaik, A. Ndirango, D. Newns, and G. J. Martyna, *J. Chem. Phys.* **116**, 8649 (2002).
- <sup>5</sup>A. Koi, B. B. Laird, and B. J. Leimkuhler, *J. Chem. Phys.* **107**, 2580 (1997).
- <sup>6</sup>A. Dullweber, B. Leimkuhler, and R. McLachlan, *J. Chem. Phys.* **107**, 5840 (1997).
- <sup>7</sup>R. van Zon and J. Schofield, *Phys. Rev. E* **75**, 056701 (2007).
- <sup>8</sup>H. Okumura, S. G. Itoh, and Y. Okamoto, *J. Chem. Phys.* **126**, 084103 (2007).
- <sup>9</sup>M. Hiyama, T. Kinjo, and S. Hyodo, *J. Phys. Soc. Jpn.* **77**, 064001 (2008).
- <sup>10</sup>Y. Kajima, M. Hiyama, S. Ogata, and T. Tamura, *J. Phys. Soc. Jpn.* **80**, 114002 (2011).
- <sup>11</sup>H. Goldstein, *Classical Mechanics*, 2nd ed. (Addison-Wesley, MA, 1980).
- <sup>12</sup>In Ref. 10, we used  $(\xi, \eta, \zeta, \chi)$  instead of  $(q_0, q_1, q_2, q_3)$ . The relations are,  $q_0 = \chi$ ,  $q_1 = \eta$ ,  $q_2 = -\xi$ , and  $q_3 = \zeta$ .
- <sup>13</sup>If  $s = 1 - \frac{1}{2}|\frac{1}{2}\vec{\omega}(t)|^2\Delta t^2 < 0$ , it implies that  $|\vec{\omega}(t)|\Delta t > 2\sqrt{2} > \frac{2\pi}{3}$ . Roughly speaking, it means that three steps make one rotation. It is the time step too long for a molecular dynamics simulation to be applied. Actually, in our simulation,  $1 \geq s > 0.99$  ( $\Delta t = 2$  fs), and  $1 \geq s > 0.75$  ( $\Delta t = 10$  fs). The situation of  $s < 0$  never happened in our simulations. Similarly, if  $\vec{V} > 1$ , we have  $|\vec{\omega}(t) + \frac{1}{2}\frac{d}{dt}\vec{\omega}(t)|\Delta t > 2 \approx \frac{2\pi}{3}$  ( $= 2.09 \dots$ ). Since  $\vec{\omega}(t) + \frac{1}{2}\frac{d}{dt}\vec{\omega}(t)\Delta t$  is roughly equal to the mean of angular velocities of the present step and the next time step,  $|\vec{V}| < 1$  is always satisfied in actual simulation.
- <sup>14</sup>W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, *J. Chem. Phys.* **79**, 926 (1983).
- <sup>15</sup>D. Eisenberg and W. Kauzman, *The Structure and Properties of Water* (Oxford University Press, Oxford, 2005).
- <sup>16</sup>Since the system temperature increases significantly after  $10^5$  steps for  $\Delta t = 7.0$  fs, we calculate  $\epsilon(1)$  in a short run of  $10^3$  steps, and regard  $(\epsilon(2000) - \epsilon(100))/(2000 - 100)$  in a run of 5000 steps as  $\tilde{\epsilon}$  in such a case.
- <sup>17</sup>We define the number of operations by the sum of mathematical operations appeared in the algorithm for updating angular velocity and quaternion of a rigid molecule. Here, mathematical operations are addition, subtraction, multiplication, division, exponentiation, trigonometric function, and square root, and each of these operations is counted as one operation. Substitution is not considered as operation. For example, the number of operations of Eqs. (36) is 9. Operations in the FT2 algorithm mean the operations appeared in Eqs. (35), (36), (37) and (38)–(40), and (20), and operations in the symplectic algorithm are operations appeared in Eqs. (58)–(84) in Ref. 8, omitting operations concerning with thermostat and translational motion.