

# The Euler Revolutions: Rotational Motion in the Laboratory Frame II.

**W. Smith**, Computational Science and Engineering Department,  
Daresbury Laboratory, Warrington WA4 4AD, United Kingdom (E-mail:  
[w.smith@dl.ac.uk](mailto:w.smith@dl.ac.uk))

## Abstract

In this article we describe a new algorithm for rotational motion in molecular dynamics simulations based on an earlier proposed scheme for rigid bodies. The new algorithm requires neither quaternions nor Euler angles and works by updating the local Cartesian axes of the rotating body, the components of which also provide the rotation matrix.

## Introduction

The first article in this series [1] examined the physics of torque driven rotational motion in both the laboratory frame and the principal frame in which the moment of inertia tensor is diagonal. The relationship between the two descriptions was discussed and it was suggested that, for molecular dynamics, in which integration of the equations of motion is entirely numerical, working in the principal frame was not necessarily the best option. This theme was expanded in the second article [2] and a new algorithm for rigid molecules was presented that was couched entirely in the laboratory frame. Molecular orientation was implicitly defined by the position vectors of the atoms in a molecule with respect to the molecular centre of mass. In consequence neither Euler angles nor quaternions were needed. This third article examines a variant of the algorithm presented in [2] which is appropriate for other kinds of rigid body besides those defined as assemblies of point particles, such as Gay-Berne ellipsoids [3] or Gaussian molecules [4]. The application of the algorithm to linear molecules is also discussed.

In the following section we outline the basics of rotational motion and present key equations. These will not be fully derived here, so the reader is directed to our previous articles [1,2] and to a more advanced text [5] if a more detailed background is required.

## The Rotational Equations of Motion for Rigid Molecules

The algorithm we describe here is relevant to a rigid body with a three-dimensional shape. It is characterised by a total mass,  $M$ , and a moment of inertia tensor,  $I$ , which is a  $3 \times 3$  matrix with components that vary with time. The molecule's centre of mass has a position  $\vec{R}$  in space, a momentum  $\vec{P}$  and a translational velocity  $\vec{V}$ . The molecule also rotates about its centre of mass with an angular momentum  $\vec{J}$  and angular velocity  $\vec{\omega}$ . In this algorithm all the vectors and tensors are defined in the laboratory frame of reference.

The laboratory frame is defined by mutually orthogonal unit vectors  $\vec{e}_1, \vec{e}_2, \vec{e}_3$ , fixed in space. A general vector,  $\vec{v}$ , with components  $(v_1, v_2, v_3)$  is therefore written as

$$\vec{v} = v_1 \vec{e}_1 + v_2 \vec{e}_2 + v_3 \vec{e}_3. \quad (1)$$

The local frame of a rotating body is also based on a set mutually orthogonal unit vectors,  $\vec{e}_1', \vec{e}_2', \vec{e}_3'$ , which is chosen so that the moment of inertia tensor in this frame is a diagonal matrix.

In the local frame the vector,  $\vec{v}$ , has components  $(v_1', v_2', v_3')$  and can be written as

$$\vec{v} = v_1' \vec{e}_1' + v_2' \vec{e}_2' + v_3' \vec{e}_3'. \quad (2)$$

This is the same vector as that given in (1), but expressed in a different reference frame. The relationship between the two representations is defined by the  $3 \times 3$  rotation matrix,  $\mathbf{R}$ , such that

$$\vec{e}_i = \sum_{j=1}^3 R_{ij} \vec{e}_j' \quad \text{and} \quad \vec{e}_i' = \sum_{j=1}^3 R_{ji} \vec{e}_j \quad (3)$$

and

$$v_i = \sum_{j=1}^3 R_{ij} v_j' \quad \text{and} \quad v_i' = \sum_{j=1}^3 R_{ji} v_j. \quad (4)$$

It is useful to note that, since

$$\vec{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \vec{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \vec{e}_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad (5)$$

we can write directly from the second equation in (3)

$$\vec{e}_1' = \begin{bmatrix} R_{11} \\ R_{21} \\ R_{31} \end{bmatrix}, \quad \vec{e}_2' = \begin{bmatrix} R_{12} \\ R_{22} \\ R_{32} \end{bmatrix}, \quad \text{and} \quad \vec{e}_3' = \begin{bmatrix} R_{13} \\ R_{23} \\ R_{33} \end{bmatrix}. \quad (6)$$

So the rotation matrix may be written in the abbreviated form

$$\mathbf{R} = [\vec{e}_1', \vec{e}_2', \vec{e}_3'], \quad (7)$$

in which the vectors form columns in the matrix. The relationship (7) is central to the new algorithm. In essence the algorithm integrates the motion of the vectors  $\vec{e}_1', \vec{e}_2', \vec{e}_3'$ , which from (7) also provides the updated rotation matrix.

The equations of motion for a rigid body are written as [1,2,5]:

$$\dot{\vec{P}} = \vec{F}, \quad \dot{\vec{J}} = \vec{T}, \quad (8)$$

where  $\vec{F}$  is the net force acting on the body and  $\vec{T}$  is the net torque. Formally these are derived from the potential energy of the body by differentiation:

$$\vec{F} = -\frac{\partial}{\partial \vec{R}} \phi(\vec{R}, \mathbf{R}) \quad \vec{T} = -\sum_{j=1}^3 \vec{e}_j' \times \frac{\partial \phi(\vec{R}, \mathbf{R})}{\partial \vec{e}_j'}. \quad (9)$$

The potential depends on the position and the orientation of the body, which are represented by the arguments  $\vec{R}$  and  $\mathbf{R}$ . The second of these equations is a general expression for the torque, based on the method described by Allen and Germano [6]. The standard relations between momentum and velocity are:

$$\vec{P} = M \vec{V}, \quad \vec{J} = \mathbf{I} \vec{\omega}, \quad (10)$$

with the help of which we can write the equations of motion as

$$\dot{\vec{V}} = \vec{F}/M, \quad \dot{\vec{\omega}} = \mathbf{I}^{-1}(\vec{T} - \dot{\mathbf{I}} \vec{\omega}). \quad (11)$$

The second of these equations recognises that, unlike the mass,  $M$ , the moment of inertia tensor,  $\mathbf{I}$ , is time dependent. Writing the components of  $\mathbf{I}$  as

$$I^{\alpha\beta} = \int_V (\delta_{\alpha\beta} d^2 - d^\alpha d^\beta) dm, \quad (12)$$

which is a mass integral over the volume  $V$ , where  $\vec{d}$  is the vector locating the mass element  $dm$  with respect to the molecular centre of mass. We can obtain the derivative,  $\dot{\mathbf{I}}$ , by differentiating with respect to time  $t$  inside the integral. This leads to

$$\begin{aligned} \dot{I}^{xy} = \dot{I}^{yx} &= \omega^z (I^{xx} - I^{yy}) - \omega^x I^{xz} + \omega^y I^{yz}, & \dot{I}^{xx} &= 2(\omega^y I^{xz} - \omega^z I^{xy}), \\ \dot{I}^{yz} = \dot{I}^{zy} &= \omega^x (I^{yy} - I^{zz}) - \omega^y I^{yx} + \omega^z I^{zx}, & \dot{I}^{yy} &= 2(\omega^z I^{yx} - \omega^x I^{yz}), \\ \dot{I}^{zx} = \dot{I}^{xz} &= \omega^y (I^{zz} - I^{xx}) - \omega^z I^{zy} + \omega^x I^{xy}, & \dot{I}^{zz} &= 2(\omega^x I^{zy} - \omega^y I^{zx}). \end{aligned} \quad (13)$$

We require one other equation. In a body rotating with angular velocity  $\vec{\omega}$ , a point in the body, located by the vector,  $\vec{d}$ , (originating at the centre of mass of the body,) has a velocity around the centre of mass given by

$$\vec{v} = \vec{\omega} \times \vec{d}. \quad (14)$$

This equation can, in principle, be integrated to update the vector  $\vec{d}$  as the body rotates. However, as our previous article [2] explained, this must be done with extreme care. We propose to use the techniques described there to integrate motion of the vectors  $\vec{e}_1'$ ,  $\vec{e}_2'$ ,  $\vec{e}_3'$ . We will

also use the scheme described in [2] to integrate the angular velocity equation (11). We describe the new algorithm in the next section.

### The Integration Algorithm

Our algorithm is couched in the velocity Verlet form [7] for both translational and rotational motion.

The mass  $M$  and the diagonal moment of inertia tensor  $\mathbf{I}^p$  (the *principal* tensor) are known constants. At the start of each time step it is assumed that  $\tilde{\mathbf{R}}, \tilde{\mathbf{V}}, \tilde{\mathbf{F}}, \mathbf{R}, \tilde{\mathbf{J}}$  and  $\tilde{\mathbf{T}}$  are known. The matrix  $\mathbf{R}$  (as defined by its use in equations (3) and (4)) can be used to construct the instantaneous moment of inertia tensor,  $\mathbf{I}$ , in the laboratory frame:

$$\mathbf{I} = \mathbf{R} \mathbf{I}^p \tilde{\mathbf{R}}. \quad (15)$$

The angular velocity at any time is given by

$$\tilde{\boldsymbol{\omega}} = \mathbf{I}^{-1} \tilde{\mathbf{J}}. \quad (16)$$

The  $3 \times 3$  matrix  $\mathbf{I}^{-1}$  being easily obtained from

$$\mathbf{I}^{-1} = \mathbf{R} (\mathbf{I}^p)^{-1} \tilde{\mathbf{R}}. \quad (17)$$

The new algorithm is presented in *Scheme 1*:

*Scheme 1:*

1. Start step  $n$  with  $\tilde{\mathbf{R}}^n, \tilde{\mathbf{V}}^n, \tilde{\mathbf{F}}^n, \tilde{\mathbf{J}}^n, \tilde{\mathbf{T}}^n, \mathbf{R}^n \equiv \{\tilde{\mathbf{e}}_i^n\}$ .
2. Calculate:  $\mathbf{I}^n = \mathbf{R}^n \mathbf{I}^p \tilde{\mathbf{R}}^n$  and  $(\mathbf{I}^n)^{-1} = \mathbf{R}^n (\mathbf{I}^p)^{-1} \tilde{\mathbf{R}}^n$ .
3. Calculate:  $\tilde{\boldsymbol{\omega}}^n = (\mathbf{I}^n)^{-1} \tilde{\mathbf{J}}^n$ .
4. Calculate:  $\dot{\mathbf{I}}^n$ , using  $\mathbf{I}^n, \tilde{\boldsymbol{\omega}}^n$  and equation (13).
5. Update:  $\tilde{\mathbf{V}}^{n+1/2} \leftarrow \tilde{\mathbf{V}}^n + \frac{\Delta t}{2M} \tilde{\mathbf{F}}^n$ .
6. Update:  $\tilde{\mathbf{J}}^{n+1/2} \leftarrow \tilde{\mathbf{J}}^n + \frac{\Delta t}{2} \tilde{\mathbf{T}}^n$ .
7. Update:  $\tilde{\boldsymbol{\omega}}^{n+1/2} \leftarrow \tilde{\boldsymbol{\omega}}^n + \frac{\Delta t}{2} (\mathbf{I}^n)^{-1} (\tilde{\mathbf{T}}^n - \dot{\mathbf{I}}^n \tilde{\boldsymbol{\omega}}^n)$ .
8. Update:  $\tilde{\mathbf{R}}^{n+1} \leftarrow \tilde{\mathbf{R}}^n + \Delta t \tilde{\mathbf{V}}^{n+1/2}$ .
9. Update:  $\tilde{\mathbf{e}}_i^{n+1} \leftarrow \tilde{\mathbf{e}}_i^n + \Delta t \tilde{\boldsymbol{\omega}}^{n+1/2} \times \tilde{\mathbf{e}}_i^n, \quad \forall \{\tilde{\mathbf{e}}_i^n\}$  (See below!)
10. Calculate:  $\mathbf{R}^{n+1} = [\tilde{\mathbf{e}}_1^{n+1}, \tilde{\mathbf{e}}_2^{n+1}, \tilde{\mathbf{e}}_3^{n+1}]$ .

11. Calculate  $\vec{F}^{n+1}$  and  $\vec{T}^{n+1}$  using equation (9).
12. Update:  $\vec{V}^{n+1} \leftarrow \vec{V}^{n+1/2} + \frac{\Delta t}{2M} \vec{F}^{n+1}$ .
13. Update:  $\vec{J}^{n+1} \leftarrow \vec{J}^{n+1/2} + \frac{\Delta t}{2} \vec{T}^{n+1}$ .

As was described in reference [2], step 9 in this algorithm requires a more sophisticated approach than the direct equation shown. We shall reiterate the explanation here.

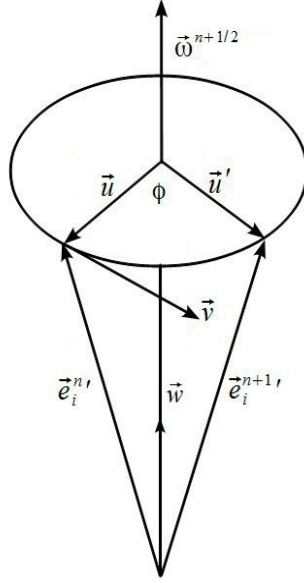


Figure 1. Rotating the Vector  $\vec{e}_i^{n'}$

Figure 1 shows the rotation of unit vector  $\vec{e}_i^{n'}$  about the vector  $\vec{\omega}^{n+1/2}$  through an angle  $\phi = \Delta t |\vec{\omega}^{n+1/2}|$ . This is the operation required by step 9 of *Scheme 1*. The simple linear step implied by step 9 will not allow the vector to follow the circle shown. To accomplish this numerically we first construct three vectors,  $\vec{w}$ ,  $\vec{u}$  and  $\vec{v}$ , (shown in Figure 1) which are defined as follows

$$\vec{w} = \frac{(\vec{e}_i^{n'} \cdot \vec{\omega}^{n+1/2})}{|\vec{\omega}^{n+1/2}|^2} \vec{\omega}^{n+1/2}, \quad (18)$$

$$\vec{u} = \vec{e}_i^{n'} - \vec{w}, \quad (19)$$

$$\vec{v} = \vec{\omega}^{n+1/2} \times \vec{e}_i^{n'}. \quad (20)$$

Vector  $\vec{w}$  is the projection of  $\vec{e}_i^{n'}$  along the vector  $\vec{\omega}^{n+1/2}$ . Vector  $\vec{u}$  defines the circle of rotation in Figure 1 and has a length equal to the circle radius. In the time step  $\Delta t$ ,  $\vec{u}$  rotates through the angle  $\phi$  to  $\vec{u}'$ . The angle  $\phi$  is given by

$$\phi = \Delta t |\vec{\omega}^{n+1/2}| = \Delta t |\vec{v}|/|\vec{u}|. \quad (21)$$

Vector  $\vec{v}$  is the instantaneous velocity of the point located by  $\vec{e}_i^n$ . The vectors  $\vec{w}$ ,  $\vec{u}$  and  $\vec{v}$  are all orthogonal by construction. It can be easily shown that

$$\vec{u}' = \cos(\phi)\vec{u} + \Delta t \frac{\sin(\phi)}{\phi} \vec{v}. \quad (22)$$

So  $\vec{e}_i^{n+1'}$  can be obtained from

$$\vec{e}_i^{n+1'} = \vec{u}' + \vec{w}. \quad (23)$$

The algorithm that performs these steps is presented in *Scheme 2*.

*Scheme 2:*

1. For each vector:  $\{\vec{e}_i^n, i=1,3\}$  :
  2. Calculate:  $\vec{w} = \left( \left( \vec{e}_i^n \cdot \vec{\omega}^{n+1/2} \right) / \left| \vec{\omega}^{n+1/2} \right|^2 \right) \vec{\omega}^{n+1/2}$
  3. Calculate:  $\vec{u} = \vec{e}_i^n - \vec{w}$
  4. Calculate:  $\vec{v} = \vec{\omega}^{n+1/2} \times \vec{e}_i^n$
  5. Calculate:  $\phi = \Delta t |\vec{\omega}^{n+1/2}|$
  6. Update:  $\vec{e}_i^{n+1'} \leftarrow \vec{w} + \vec{u} \cos(\phi) + \Delta t \vec{v} (\sin(\phi)/\phi)$
7. End For

This should be used in place of step 9 of *Scheme 1*. We note that when angle  $\phi$  is particularly small we may use Maclaurin's expansion of the trigonometric functions  $\sin(\phi)$  and  $\cos(\phi)$  to simplify step 6 of *Scheme 2*, which becomes the computationally efficient form:

$$\vec{e}_i^{n+1'} \leftarrow \vec{w} + \vec{u} \left( 1 - \frac{\phi^2}{2} \left( 1 - \frac{\phi^2}{12} \left( 1 - \frac{\phi^2}{30} \right) \right) \right) + \Delta t \vec{v} \left( 1 - \frac{\phi^2}{6} \left( 1 - \frac{\phi^2}{20} \left( 1 - \frac{\phi^2}{42} \right) \right) \right). \quad (24)$$

The expansions in expression (24) neglect terms of order  $O(\phi^8)$  and are therefore highly accurate for  $\phi$  less than  $\sim 1^\circ$ . It is preferable to use them when  $\phi$  is small, since it avoids any numerical problems associated with the ratio  $\sin(\phi)/\phi$  in step 6 and is accurate enough to preserve the unit length of  $\vec{e}_i'$  for a long period. (A re-normalisation of  $\vec{e}_i'$  could be performed if this ever became problematic.)

### Adaptation to Linear Molecules

Linear molecules appear to present a problem for the new algorithm. Steps 3 and 7 of *Scheme 1* involve the inverse of matrix  $I$ . If one of the principal moments of inertia of the molecule is zero then formally the inverse  $I^{-1}$  does not exist. This implies that the algorithm cannot be used directly with linear molecules. However it is possible to patch the

algorithm with a simple change to accommodate linear molecules, as we explain below.

Firstly, consider the case of a uni-axial ellipsoid of the Gay-Berne or Gaussian kind. These are characterised by having two equal moments of inertia,  $I_{xx}^p = I_{yy}^p$ , (which follows from molecular symmetry) and one unique moment  $I_{zz}^p$ . All three moments are finite in this case. Next we write the angular equation of motion (11) in the form

$$\vec{T} = \mathbf{I} \dot{\vec{\omega}} + \dot{\mathbf{I}} \vec{\omega}, \quad (25)$$

where, in the laboratory frame,  $\mathbf{I}$  is not generally diagonal. We now multiply (25) by a rotation matrix  $\mathbf{R}$ , which is able to diagonalise  $\mathbf{I}$  at a nominated fixed point in time, and obtain

$$\tilde{\mathbf{R}} \vec{T} = \tilde{\mathbf{R}} \mathbf{I} \mathbf{R} \tilde{\mathbf{R}} \dot{\vec{\omega}} + \tilde{\mathbf{R}} \dot{\mathbf{I}} \mathbf{R} \tilde{\mathbf{R}} \vec{\omega}, \quad (26)$$

where we have exploited the relation  $\mathbf{R} \tilde{\mathbf{R}} = \mathbf{1}$ , where  $\mathbf{1}$  is the identity matrix. This equation can now be written as

$$\vec{T}^p = \mathbf{I}^p \dot{\vec{\omega}}^p + \tilde{\mathbf{R}} \dot{\mathbf{I}} \mathbf{R} \vec{\omega}^p, \quad (27)$$

where the superscript  $p$  indicates the principal frame of reference. It was shown in reference [1] that (27) is another form of Euler's classical equations of motion, which we write explicitly as

$$\begin{aligned} T_x^p &= I_{xx}^p \dot{\omega}_x^p - \omega_y^p \omega_z^p (I_{yy}^p - I_{zz}^p), \\ T_y^p &= I_{yy}^p \dot{\omega}_y^p - \omega_z^p \omega_x^p (I_{zz}^p - I_{xx}^p), \\ T_z^p &= I_{zz}^p \dot{\omega}_z^p - \omega_x^p \omega_y^p (I_{xx}^p - I_{yy}^p). \end{aligned} \quad (28)$$

From this we see that equation (25) and Euler's equations are physically equivalent. They differ only in the choice of reference frame. Equation (25) is couched in the laboratory frame, while (28) is in the principal frame, which is a stationary frame<sup>1</sup> in which the moment of inertia tensor is instantaneously diagonal. Euler's equations thus only hold at one particular instant, though we can always find a different principal frame for any other instant.

Because  $I_{xx}^p = I_{yy}^p$ , the third of the equations in (28) can be written as

$$T_z^p = I_{zz}^p \dot{\omega}_z^p. \quad (29)$$

This equation is solely responsible for rotation around the (principal) z-axis and is independent of the mechanics of rotation about the x- and y-axes. Furthermore, since an ellipsoidal molecule is rotationally symmetric about the z-axis, the torque component  $T_z^p$  derived from interaction with

---

<sup>1</sup> The principal frame differs from the local frame in that the latter is rotating with the molecule while the former is stationary. They happen to coincide at the instant equation (28) holds.

other molecules must be zero. It follows from (29) that  $\dot{\omega}_z^p$  must also be zero.

Thus we cannot expect  $\omega_z^p$  ever to change, since there is nothing to drive such change in *any* principal frame of reference. For this reason we may set  $\omega_z^p=0$  once and for all time, since it is dynamically redundant. Setting  $\omega_z^p=0$  (and ignoring for a moment that  $T_z^p=\dot{\omega}_z^p=0$ ) means Euler's equations (28) are now reduced to

$$\begin{aligned} T_x^p &= I_{xx}^p \dot{\omega}_x^p, \\ T_y^p &= I_{yy}^p \dot{\omega}_y^p, \\ T_z^p &= I_{zz}^p \dot{\omega}_z^p. \end{aligned} \tag{30}$$

The third equation of (30) is of course redundant since the dynamics of the molecule are governed *entirely* by the first two equations, which are not in any way coupled to the third. Nevertheless the algorithm presented in *Scheme 1* is entirely suitable for simulating this system, provided the simulation starts (and thus continues) with the condition  $\omega_z^p=0$ . *Scheme 1* works in this case because, despite the redundancy in the set of equations (30), the inverse  $I^{-1}$  of the moment of inertia tensor  $I$  can always be obtained.

At this point we note that the first two equations of (30) are the same as those for a linear molecule consisting of point particles. In this case the third equation does not exist, but as we have noted, it is also redundant for ellipsoidal molecules. It follows that all that is required to adapt *Scheme 1* to linear molecules, is to set  $I_{zz}^p$  to a convenient finite value and ensure that the simulation starts with  $\omega_z^p=0$ . The two systems, linear and ellipsoidal molecules, are dynamically isomorphic under these circumstances.

In practice it is best to initialise a simulation with every molecule defined in its principal frame so that the occurrence of a zero moment of inertia can be identified and reset to a finite value, while the corresponding component of the angular momentum is set to zero. The entire system can then be recast back to the laboratory frame. Note that subsequent values of the moment of inertia tensor  $I$  and its inverse are calculated from  $I^p$  using equations (15) and (17), which appear in step 2 of *Scheme 1*. It is not necessary to 'carry' current values of  $I$  or  $\dot{I}$  throughout the simulation.

Some precautions are advised. It should be noted that linearity implies the molecule has lost one rotational degree of freedom and this must be remembered when converting the kinetic energy into temperature. This also applies to uni-axial ellipsoids of the Gay-Berne or Gaussian kind. Also, some caution is necessary when deciding the number of degrees of freedom of a molecule based on inspecting the principal moments of inertia alone. It may be accidentally true that two or more moments are equal, without rendering a degree of freedom redundant.



## Simulation Tests

We have written a program, *revo2.f*, that incorporates the new rotation algorithm and have conducted a series of tests. Four test systems were created, all of them consisting of 27 molecules constructed from 4 Lennard-Jones (L-J) sites with identical L-J potential parameters. All simulations were of length  $10^6$  time steps with a  $10^4$  time step equilibration period. Two time steps  $\Delta t$  were tried:  $\Delta t=0.001$  and  $\Delta t=0.0005$ . The NVE ensemble was used for all simulations. The systems modelled were:

1. *Pseudo-methane*: a tetrahedral molecule with atom-atom distances of 1.011 and all atomic masses set at 1.0. The system reduced molecular density was  $\rho=0.175$ , ( corresponding to a reduced atom density of  $\rho=0.7$  ) and the reduced temperature was  $T=1.5$ .
2. *Pseudo-ammonia*: a trigonal pyramidal molecule with atomic masses of 1.0, 0.5, 0.5, 0.5 and bond length 1.175. The system reduced molecular density was  $\rho=0.2$  and reduced temperature  $T=1.0$ .
3. *Pseudo-water*: three L-J atoms with masses 1.0, 0.1, 0.1 arranged in a triangle with one massless L-J site close to the molecular centre of mass. The reduced molecular density was  $\rho=0.225$  and the reduced temperature was  $T=1.25$ .
4. *Pseudo-butane*: four identical L-J atoms in a straight line with masses 1.0 and bond length 0.3137. The reduced molecular density was  $\rho=0.15$  and the reduced temperature was  $T=1.5$ .

The results of these simulations are presented in Table 1. (L-J units are used throughout).

System	$\Delta t$	<Energy>	<Fluctuation>	Drift
$\psi$ -methane	0.001	-160.8	6.48e-4	4.44e-6
$\psi$ -methane	0.0005	-151.93	1.73e-4	-9.01e-6
$\psi$ -ammonia	0.001	-286.32	3.69e-3	1.02e-2
$\psi$ -ammonia	0.0005	-279.61	3.47e-4	2.91e-4
$\psi$ -water	0.001	-446.54	5.93e-1	2.05
$\psi$ -water	0.0005	-441.88	6.23e-2	2.13e-1
$\psi$ -butane	0.001	-356.04	1.10e-3	-8.89e-5
$\psi$ -butane	0.0005	-357.66	2.76e-4	7.27e-6

Table 1

In column 3 of Table 1 the average system energies for each simulation, obtained as an average over the last 990000 time steps, are presented. We note that differences in energy for the same system with a different time step,  $\Delta t$ , are most likely due to them settling down to a different

temperature at the end of the equilibration period. (This being a common occurrence with small systems). Apart from this, all the simulations show a high degree of stability. The RMS fluctuation in total energy (column 4) is acceptably small in all cases and it is probably significant that the fluctuation is largest in systems which have the smallest moments of inertia. Pseudo-water is the most obvious example of this. The reported energy drift in column 5 was obtained over the time scale of  $990000 \Delta t$  from the equilibrated simulation using a least-squares fit of the data. The drift in all cases is acceptable and in most of them negligible, but again it is pseudo-water that shows the largest drift. In the majority of cases, taking a smaller  $\Delta t$  value results in a reduction in both the fluctuation and the drift. The one exception to this is the drift reported for the pseudo-methane, which is larger for the shorter time step. However, in this case both values are negligible.

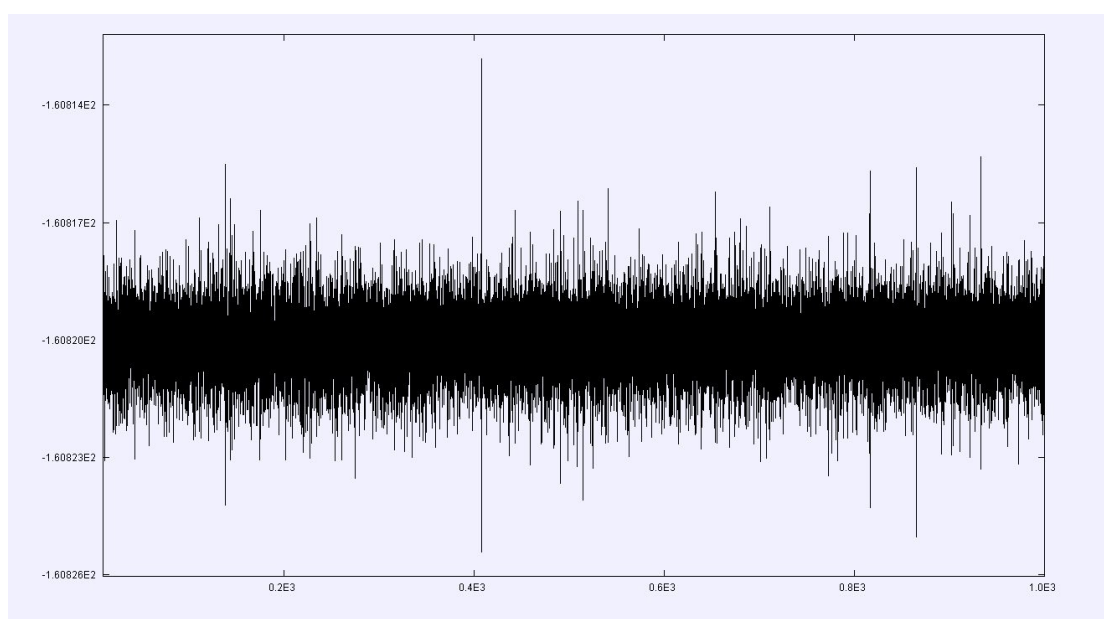


Figure 2

In Figure 2, we show a typical plot of the system energy *versus* time (in this case for the pseudo-methane system with  $\Delta t=0.001$ . The time interval shown is 990000 time steps.) The conservation of energy and small scale of the fluctuations are apparent.

For the linear molecule pseudo-butane, we have also monitored the supposed zero component of the angular velocity in the principal frame. We report that this was found to hold to better than 1 part in  $10^{10}$ , which surely confirms the theory presented above. (We have also subsequently found it to hold in simulations of uni-axial Gaussian ellipsoids, again to high accuracy, using the same rotation algorithm as presented here. In that case we were able to change the moment of inertia of the rotationally redundant axis by 4 orders of magnitude, without affecting the system dynamics noticeably for the 10,000 time steps of the test. This further confirms the theory.)

## Conclusion

The new algorithm presented appears to provide a reliable and accurate numerical integration of the rotation of rigid molecules and is easily extendable to linear molecules.

## References

- [1] W. Smith, *Euler among the Pedestrians*, (2002), Daresbury Laboratory, CCP5 Infoweb Knowledge Centre on [www.ccp5.ac.uk](http://www.ccp5.ac.uk)
- [2] W. Smith, *Hail Euler and Farewell: Rotational motion in the laboratory frame*, (2005), Daresbury Laboratory, CCP5 Infoweb Knowledge Centre on [www.ccp5.ac.uk](http://www.ccp5.ac.uk)
- [3] J.G. Gay and B.J. Berne, *Modification of the overlap potential to mimic a linear site-site potential*, J. Chem. Physics **74** (1981) 3316.
- [4] J.A. Purton and W. Smith, *Dissipative particle dynamics of non-spherical particles using a Gaussian density model*, Molecular Simulation 36 (2010) 796.
- [5] H. Goldstein, *Classical Mechanics*, Addison Wesley (1980).
- [6] M.P. Allen and G. Germano, Expressions for forces and torques in molecular simulations using rigid bodies, Molec. Physics **104** (2006) 3225.
- [7] M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids*, Oxford University Press (1980).