Numerical Integration Scheme for md_serial.f

Evangelos Coutsias[‡]
Department of Mathematics and Statistics

Paul M. Alsing[†]
Albuquerque High Performance Computing Center
University of New Mexico, Albuquerque, NM 87131

31 Aug 2000

Abstract

In these notes we detail the numerical integration scheme used to translate and rotate molecules in the 3D molecular dynamics code `md_serial.f`. We give details using the example of the H_2O molecule in the liquid state.

1 Introduction

The code `md_serial.f` numerically integrates Newton's equations of motion for the TIP4P model of water [1]. The integration scheme updates both the translational and rotational motion of the molecules in 3D space. The translational motion is evolved with a velocity-Verlet scheme which is second order accurate in Δt i.e. $\mathcal{O}(\Delta t^2)$. The rotational motion is affected through the use of quaternions [2]. The numerical update uses a variable order Adams-Bashforth scheme (up to 4th order) coupled with Newton-Raphson step(s) to solve the nonlinear quaternion angular velocity equations. The unique feature of our incorporation of a Newton step is the analytical computation

of the quaternion Jacobian. Below we outline the details of the integration scheme used to update the equations of motion from time t to $t + \Delta t$.

2 Equations of Motion

2.1 Translations

Newton's equations of motion for the translational degrees of freedom have the form

$$m\ddot{\boldsymbol{r}}_i = \boldsymbol{f}_i = \sum_{j=1 j\neq i}^{N_a} \boldsymbol{f}_{ij} \tag{1}$$

where $\boldsymbol{f}_{ij}$ is the force on molecule i due to molecule j, $\boldsymbol{f}_i$ is the total force on molecule i due to all the other molecules and N_a is the total number of atoms. We write this as a first order system

$$\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i \tag{2}$$

$$\dot{\boldsymbol{v}}_i = \frac{\boldsymbol{f}_i}{m}. \tag{3}$$

In our work we follow Rapaport[1] and scale all masses to the mass of the water molecule which we set to unity, $m_O + 2m_H \equiv 1$.

2.2 Rotations

For the rotational motion of the molecules, we must distinguish between two frames of reference: the *space-fixed* frame which is a fixed inertial reference frame and the *body-fixed* frame, which is a non-inertial frame affixed to the body. We will refer to these reference frames as the *body frame* and the *space frame*. Notationally, we will use capital case letters to indicate quantities in the body frame and lower case letters to indicate the same quantites refered to the space frame.

In the space-fixed frame the relevant equation of motion is the torque equation

$$\dot{\boldsymbol{m}}_i = \boldsymbol{\tau}_i = \sum_{j=1 j\neq i}^{N_a} \boldsymbol{\tau}_{ij}, \tag{4}$$

where $\boldsymbol{m} = (m_1, m_2, m_3)$ is the angular momentum of the particle of particle i, $\boldsymbol{\tau}_i j$ is the torque on particle i due to particle j and $\boldsymbol{\tau}_i = \boldsymbol{r}_i \times \boldsymbol{f}_i$ is the the total torque on particle i.

Often the torque equation Eq. (4) is considered in the body frame via transformation between body-refered and space-revered velocities

$$\left(\frac{d\boldsymbol{m}}{dt}\right)_{\text{space}} = \left(\frac{d\boldsymbol{M}}{dt}\right)_{\text{body}} + \boldsymbol{\Omega} \times \boldsymbol{m}, \tag{5}$$

where $\boldsymbol{M} = (M_1, M_2, M_3)$ is the angular momentum in the body frame and $\boldsymbol{\Omega} = (\Omega_1, \Omega_2, \Omega_3)$ is the instantaneous angular velocity of the rotation in the body frame. The utility of the body frame is that the principal axes are fixed in this frame and, choosing these to be the axes of the body-fixed coordinate system we may write

$$\boldsymbol{M} = \boldsymbol{\mathcal{I}}\boldsymbol{\Omega}, \tag{6}$$

where $\boldsymbol{\mathcal{I}} = \text{diagonal}\{I_1, I_2, I_3\}$ is the diagonal matrix (in the body frame) of the principal moments of inertia $\{I_i\}$. In the body frame Eq. (4) are the Euler equations of rigid body motion with a typical component

$$I_1 \dot{\Omega}_1 = T_1 + (I_2 - I_3)\,\Omega_2\,\Omega_3. \tag{7}$$

Note that in Eq. (7) all quantities are in the body frame *including* the torques $\boldsymbol{T}$ which are computed most easily in the space fixed frame, and must be transformed into the body frame via a time dependent rotation matrix $\boldsymbol{\mathcal{Q}}(q(t))$

$$\boldsymbol{\tau} = \boldsymbol{\mathcal{Q}}(q(t))\boldsymbol{T}. \tag{8}$$

The expression for $\boldsymbol{\mathcal{Q}}(q(t))$ (body-to-space rotation) in terms of the quaternions $q = (q_0, q_1, q_2, q_3) = (q_0, \boldsymbol{q})$ [3] (discussed in Section 4.2 below) is given on page 14 of [2] and in Eq. (36) below.

Instead of using the Euler equations in the body frame, we follow Allen & Tildesley [4] and use the torque equation Eq. (4) in the space frame, where the torques are computed most easily. However, we have not bypassed the problem of transforming quantities between the body and space frames, we have just hidden it slightly. This is because, in order to compute the torques $\boldsymbol{\tau}$ we must orient the molecules in physical space. We do this by applying $\boldsymbol{\mathcal{Q}}(q(t))$ to a fixed, fiducial orientation of the molecules in the body frame,

$$\boldsymbol{r}(t)_{space} = \boldsymbol{\mathcal{Q}}(q(t))\boldsymbol{R}(0)_{body}. \tag{9}$$

Note that in order to accomplish this, we need to update the time dependent rotation matrix $\boldsymbol{\mathcal{Q}}(q(t))$ via the equation

$$\dot{q} = \frac{1}{2} q\,\Omega \qquad \text{(quaternion multiplication)} \tag{10}$$

where the unbold-face quantities q and $\Omega = (0, \boldsymbol{\Omega})$ are quaternions and the right hand side (rhs) indicates *quaternion multiplication* (see [2]) and Section 4.2 below.

2.3 Equations of Motion

Here we collect the equations of motion discussed above and used in our codes

$$\begin{aligned}
\dot{\boldsymbol{r}}_i &= \boldsymbol{v}_i, && (11)\\
\dot{\boldsymbol{v}}_i &= \boldsymbol{f}_i, && (12)\\
\dot{\boldsymbol{m}}_i &= \boldsymbol{\tau}_i, && (13)\\
\boldsymbol{\Omega}_i &= \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{M}_i = \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{\mathcal{Q}}^{-1}\boldsymbol{m}_i, && (14)\\
\dot{q}_i &= \frac{1}{2} q_i\,\Omega_i. && (15)
\end{aligned}$$

where $\boldsymbol{\mathcal{Q}}^{-1} = \boldsymbol{\mathcal{Q}}^T$ (transpose) is the space-to-body rotation matrix.

3 Integration Scheme Implementation

In the following $\boldsymbol{r}^n \equiv \boldsymbol{r}(t_n)$, etc... where $t_n = n\Delta t$. The formulas below are for the temporal update of a single particle; we drop the subscript labeling the ith particle.

Step 1. velocity-Verlet update of positions,

$$\boldsymbol{r}^{n+1} = \boldsymbol{r}^n + \left(\boldsymbol{v}^n + \boldsymbol{f}^n \frac{\Delta t}{2}\right) \Delta t \tag{16}$$

Step 2. velocity-Verlet half-update of velocity using the forces computed at time t_n,

$$\boldsymbol{v}^{n+1/2} = \boldsymbol{v}^n + \boldsymbol{f}^n \frac{\Delta t}{2} \tag{17}$$

Step 3. rth order Adams-Bashforth predictor step, to estimate the angular momentum $\boldsymbol{m}^*$ in the space frame at one full time step Δt. Note $\Delta t_1 = \Delta t / d1(r)$ where $d1(r)$ and the coefficients $a(k,r)$ are given in Table 1,

$$\begin{aligned} \boldsymbol{m}^* &= \boldsymbol{m}^n + \Delta t_1 \sum_{k=0}^{r-1} a(k,r) \boldsymbol{\tau}^{n-k} \\ &= \boldsymbol{m}^n + \Delta t_1 [\, a(0,r)\boldsymbol{\tau}^n + a(1,r)\boldsymbol{\tau}^{n-1} + \ldots + a(r-1,r)\boldsymbol{\tau}^{n-(r-1)} \end{aligned} \quad (18)$$

Step 4. Transform the angular momentum from space frame to body frame $\boldsymbol{m} \to \boldsymbol{M}$ using the predicted value $\boldsymbol{m}^*$ above and the current values q^n of the quaternions in the space-to-body rotation matrix $\boldsymbol{\mathcal{Q}}^{-1}(q^n)$. Compute the angular velocity $\boldsymbol{\Omega}$ in the body frame

$$\boldsymbol{\Omega} = \boldsymbol{\mathcal{I}}^{-1} \boldsymbol{\mathcal{Q}}^{-1} \boldsymbol{m} \quad (19)$$

Step 5. Forming the pure quaternion $\Omega \equiv (0, \boldsymbol{\Omega})$, advance the quaternion q in time

$$\dot{q} = \frac{1}{2} q\, \Omega(q) \qquad =: F(q) (\text{quaternion multiplication}). \quad (20)$$

We can discretize the above equation implicitly in time using the Adams-Bashforth corrector scheme as in Step 7 below: $(q^{n+1} - q^n)/\Delta t = \mathcal{F}(q^{n+1}, q^n, \ldots, q^{n-(r-1)})$, where $\mathcal{F}$ is a nonlinear function (discussed below). The goal here is to solve this nonlinear equation for q^{n+1}. This step is discussed in detail below in Section 4.1.

Step 6. Given q^{n+1}, the values of the quaternions at $t_{n+1} = t_n + \Delta t$, we can compute the forces and torques at time t_{n+1}. For this computation we must orient the molecules in the space frame from a static configuration in the body frame with the time dependent rotation matrix $\boldsymbol{\mathcal{Q}}(q^{n+1})$.

Step 7. rth order Adams-Bashforth corrector step: correct the values of angular momentum $\boldsymbol{m}^{n+1}$ in the space frame using the newly computed values of the torques at time t_{n+1}, and at previous time values. The coefficients $b(k,r)$ are given in Table 1,

$$\begin{aligned} \boldsymbol{m}^{n+1} &= \boldsymbol{m}^n + \Delta t_1 \sum_{k=-1}^{r-2} b(k,r) \boldsymbol{\tau}^{n-k} \\ &= \boldsymbol{m}^n + \Delta t_1 [\, b(-1,r)\boldsymbol{\tau}^{n+1} + b(0,r)\boldsymbol{\tau}^n \\ &\qquad + b(1,r)\boldsymbol{\tau}^{n-1} \ldots + b(r-2,r)\boldsymbol{\tau}^{n-(r-2)} \,] \end{aligned} \quad (21)$$

Coefficients: rth order Adams-Bashforth predictor-corrector scheme									
Order r	d1(r)	a(0,r)	a(1,r)	a(2,r)	a(3,r)	b(-1,r)	b(0,r)	b(1,r)	b(2,r)
1	1	1	-	-	-	1	-	-	-
2	2	3	-1	-	-	1	1	-	-
3	12	23	16	5	-	5	8	-1	-
4	24	55	-59	37	-9	9	19	-5	1

Table 1: Coefficients for rth order Adams-Bashforth predictor (**a** coefficients) -corrector (**b** coefficents) integration scheme. Note that the sum across the rows for both the a and b coefficients individually should add up to $d1(r)$.

At this point, we may choose an additional Newton step for q, so as to base it entirely on the updated angular momentum. Currently not done. We stress that the forces, torques and quaternions are computed by predicting $\mathbf{m}^*$, updating q^{n+1} through Newton step from predicted $\omega^{\mathbf{n+1}}$, at which point space orientations are computed and forces and torques evaluated. Then a corrected form of angular momentum, $\mathbf{m}^{\mathbf{n+1}}$ is found.

Step 8. Complete the velocity-Verlet update of the velocity using the forces computed at time t_{n+1},

$$\boldsymbol{v}^{n+1} = \boldsymbol{v}^{n+1/2} + \boldsymbol{f}^{n+1}\frac{\Delta t}{2} \tag{22}$$

4 Newton Implicit Step

4.1 Outline of Method

In this section we solve the nonlinear quaternion equations Eq. (20) for q^{n+1} as outlined previously in Step 5. We discretize Eq. (20) using the Adams-Bashforth corrector as in Step 7 above

$$\begin{aligned} q^{n+1} &= q^n + \Delta t_1[\, b(-1,r)F(q^{n+1}) + b(0,r)F(q^n) \\ &+ b(1,r)F(q^{n-1}) + \ldots + b(r-2,r)F(q^{n-(r-2)})\,], \end{aligned} \tag{23}$$

where F is a nonlinear function of the q's whos form will be explicitly formulated below. As stated above, the goal is to solve the above implicit nonlinear equation for q^{n+1}. Thus we can rewrite Eq. (23) as

$$\begin{aligned} q^{n+1} - \Delta t_1 b(-1,r) F(q^{n+1}) &= q^n + \Delta t_1 [\, b(0,r)F(q^n) + b(1,r)F(q^{n-1}) + \\ &\quad \ldots + b(r-2,r)F(q^{n-(r-2)})\,], \\ &= q^n + \Delta t_1 \sum_{k=0}^{r-2} b(k,r) F(q^{n-k}) \end{aligned} \tag{24}$$

Note that the lhs of Eq. (24) contains the terms that depend on future value q^{n+1} while the rhs of Eq. (24) depends on the current and past values $\{q^n, q^{n-1}, \ldots\}$.

The Newton method [5] works by writing

$$q^{n+1} \equiv q^* + \phi, \tag{25}$$

where q^* is our current estimate of q^{n+1} and ϕ is the correction we add to q^* to get the next iterative improvement to q^{n+1}. At the start of the method we initialize q^* to q^n. Inserting Eq. (25) into Eq. (24) and expanding to first order in ϕ yields

$$\begin{aligned} \mathcal{D}\phi &\equiv [\, \Delta t_1 b(-1,r) DF(q^*) - \mathbf{1}\,]\phi \\ &= q^* - \Delta t_1 b(-1,r) F(q^*) - q^n - \Delta t_1 \sum_{k=0}^{r-2} b(k,r) F(q^{n-k}) \end{aligned} \tag{26}$$

where $DF(q^*)$ is the Jacobian matrix of $\mathcal{F}$ evaluated at q^* and $\mathcal{D}$ is the matrix operator in the square brackets. $DF(q^*)$ will be computed in Section 4.3 below. We can invert $\mathcal{D}$ above by LU decomposition, and solve for ϕ. We then add ϕ to the current value of q^* to obtain the next estimate of q^{n+1}. If $|\phi| < \epsilon$ for some preset tolerance ϵ we say the Newton iterations have converged. Typically, this takes only a few steps, often as few as one. If the Newton method has not converged, we set $q^* = q^{n+1}$, reevaluate the $DF(q^*)$ and the rhs of eq.26, and resolve for the next correction ϕ. Note that on the rhs of Eq. (26), only the terms $q^* - \Delta t_1 b(-1,r) F(q^*)$ which depend on q^* need to be updated every iteration. The rest of the terms on the rhs are constant i.e. they are evaluated at $\{q^n, q^{n-1}, \ldots, q^{n-(r-2)}\}$.

4.2 Quaternion notation and manipulations

In order to calculate the Jacobian $DF(q^*)$ in Eq. (26) above, we need to find the explicit form of $F(q)$. This, in turn, requires us to convert equations involving quaternion multiplication into ones involving ordinary 4×4 matrix multiplications of 4-vectors. Below, we follow the notes of [2], an lay out the necessary ingredients for calculating the Jacobian in the next section.

We define a quaternion q as a 4-vector $q = (q_0, q_1, q_2, q_3) \equiv (q_0, \boldsymbol{q})$ with the obvious addition law and the fundamental multipication law given by

$$\begin{aligned} a = (a_0, \boldsymbol{a}), \qquad & b = (b_0, \boldsymbol{b}) \\ a + b &= (a_0 + b_0, \boldsymbol{a} + \boldsymbol{b}) \\ a\, b &= (a_0 b_0 - \boldsymbol{a} \cdot \boldsymbol{b}, a_0 \boldsymbol{b} + b_0 \boldsymbol{a} + \boldsymbol{a} \times \boldsymbol{b}) \end{aligned} \tag{27}$$

We also define the *conjugate* quaternion q^c, the norm of a quaternion $N(q)$ and the inverse quaternion q^{-1} by

$$q^c = (q_0, -\boldsymbol{q}), \qquad (ab)^c = b^c a^c \tag{28}$$

$$N(q) = q_0^2 + \boldsymbol{q} \cdot \boldsymbol{q} \tag{29}$$

$$q^{-1} = \frac{q^c}{N(q)} = \frac{(q_0, -\boldsymbol{q})}{N(q)} \tag{30}$$

An important class of quaternion are the *pure* quaternions given by

$$\mathcal{Q}_0 = \{q \,|\, q = (0, \boldsymbol{q})\}. \tag{31}$$

Pure quaternions can be considered as ordinary 3-vectors $\boldsymbol{q} = (q_1, q_2, q_3)$ which have been promoted to 4-vectors. Note for pure quaternions $a, b \in \mathcal{Q}_0$, Eq. (27) and Eq. (28) yields

$$\begin{aligned} a = (0, \boldsymbol{a}), \qquad & b = (0, \boldsymbol{b}) \\ a\, b &= (-\boldsymbol{a} \cdot \boldsymbol{b},\ \boldsymbol{a} \times \boldsymbol{b}) \end{aligned} \tag{32}$$

$$a^c = (0, -\boldsymbol{a}) = -a \tag{33}$$

The utility of quaternions is that they can affect rotations without the singularities of say, the Euler angles rotation matrices. For the 3-vector $\boldsymbol{r}'$

obtained by a rotation of the vector $\boldsymbol{r}$ via the orthogonal rotation matrix $\boldsymbol{\mathcal{Q}}$ we have

$$\begin{aligned} r' = (0, \boldsymbol{r'}), \qquad & r = (0, \boldsymbol{r}) \\ r' &= q\, r\, q^c, \qquad (34) \\ (0, \boldsymbol{r'}) &= (q_0, \boldsymbol{q})\ (0, \boldsymbol{r})\ (q_0, -\boldsymbol{q}) \\ &= (0,\ \boldsymbol{\mathcal{Q}(q)} \cdot \boldsymbol{r}). \qquad (35) \end{aligned}$$

The rotation matrix $\boldsymbol{\mathcal{Q}}$ is given by

$$\boldsymbol{\mathcal{Q}}(q) = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 - q_0 q_3) & 2(q_1 q_3 + q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 + q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (36)$$

Since $\boldsymbol{\mathcal{Q}}$ is orthogonal we have $\boldsymbol{\mathcal{Q}}^{-1} = \boldsymbol{\mathcal{Q}}^T$, where $\boldsymbol{\mathcal{Q}}^T$ is $\boldsymbol{\mathcal{Q}}$ with the signs of the off-diagonal elements reversed.

For our purposes we are interested in writing quaternion multiplications in terms of ordinary 4×4 matrix multiplication acting on a 4-vector. For any two quaternions a and q we define the 4×4 matrices A_L and A_R as the action of a on q via

$$\begin{aligned} a\, q &\equiv A_L \cdot q, \qquad a \text{ operates on } q \text{ from the left} \qquad (37) \\ q\, a &\equiv A_R \cdot q, \qquad a \text{ operates on } q \text{ from the right} \qquad (38) \end{aligned}$$

Recall that matrix multiplication (the rhs of the above) always acts from the left on a 4-vector $q = (q_0, q_1, q_2, q_3)^T$ (i.e. q as a column vector). In Eq. (37) the quaternion multiplication $a\, q$ has a acting from the left on q translating into the 4×4 matrix A_L acting on q as a column 4-vector. In Eq. (38) the quaternion multiplication $q\, a$ has a acting from the right on q translating into the 4×4 matrix A_R acting on q as a column 4-vector. For $a = (a_0, a_1, a_2, a_3)$ the matrices A_L and A_R are given by

$$A_L = \begin{pmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & a_3 & -a_2 \\ a_2 & -a_3 & a_0 & a_1 \\ a_3 & a_2 & -a_1 & a_0 \end{pmatrix}, \quad A_R = \begin{pmatrix} a_0 & -a_1 & -a_2 & -a_3 \\ a_1 & a_0 & -a_3 & a_2 \\ a_2 & a_3 & a_0 & -a_1 \\ a_3 & -a_2 & a_1 & a_0 \end{pmatrix}. \quad (39)$$

An extremely important rule to remember when performing these operations is *all quaternion operations must be completed first, before ordinary* 4×4 *matrix multiplications can be carried out.*

The equation of main concern for us is Eq. (20) in Step 5. Using the above rules we can write this as

$$\begin{aligned} \dot{q} &= \frac{1}{2} q\,\Omega \\ &= \frac{1}{2}\Omega_R \cdot q, \end{aligned} \tag{40}$$

where the rotation generator matrix Ω_R is given by

$$\Omega_R = \begin{pmatrix} 0 & -\Omega_1 & -\Omega_2 & -\Omega_3 \\ \Omega_1 & 0 & \Omega_3 & -\Omega_2 \\ \Omega_2 & -\Omega_3 & 0 & \Omega_1 \\ \Omega_3 & \Omega_2 & -\Omega_1 & 0 \end{pmatrix} \tag{41}$$

Note that we need to distinguish between 3 types of angular velocity Omegas; (1) the 3-vector $\mathbf{\Omega} = (\Omega_1, \Omega_2, \Omega_3)$, (2) the pure quaternion $\Omega = (0, \mathbf{\Omega})$ and (3) the rotation generator matrix Ω_R given by Eq. (41) above.

4.3 Calculating the Jacobian

We are now in a position to determine the form of the function F in Eq. (30) and to compute its Jacobian DF, which are both needed for the Newton-Raphson computation of q^{n+1} in Step 5 in Section 4.1. First, to determine F we use Eq. (40) written in the form

$$\Omega = \mathcal{I}_4^{-1}(q^c\, m\, q) \tag{42}$$

$$\dot{q} = \frac{1}{2} q\,\Omega = \frac{1}{2} q\{\mathcal{I}_4^{-1}(q^c\, m\, q)\} \equiv F(q) \tag{43}$$

In the above we have defined the (*pseudo*-)inverse of the moment of inertia matrix in the body frame as the 4×4 matrix (hence the subscript 4),

$$\mathcal{I}_4^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & I_1^{-1} & 0 & 0 \\ 0 & 0 & I_2^{-1} & 0 \\ 0 & 0 & 0 & I_3^{-1} \end{pmatrix}. \tag{44}$$

The parentheses () in Eq. (42) and Eq. (43) separate the quaternion multiplication $(q^c\, m\, q)$ from the matrix multiplication by $\mathcal{I}_4^{-1}$.

To compute the Jacobian DF (with respect to q only; m is kept fixed throughout this calculation) we consider the first order variation of F. We drop the fraction $1/2$ and consider the variation $\Delta(q\,\Omega)$. Our goal is to write this as some 4×4 matrix acting on the column 4-vector Δq. We have

$$\begin{aligned}\Delta(q\,\Omega) &= \Delta q\,\Omega + q\,\Delta\Omega \qquad \text{(quaternion multiplication)}\\ &= \Omega_R \Delta q + Q_L \Delta\Omega, \qquad \text{(matrix multiplication)}\end{aligned} \tag{45}$$

In Eq. (45) we have defined the 4×4 matrix Q_L formed from q (see Eq. (39)) via $q\,\Delta\Omega \to Q_L\Delta\Omega$, where we have quaternion multiplication on the lhs of the $\to$ and matrix multiplication on the rhs. Note that the first term in Eq. (45) is in the form we desire, i.e. matrix multiplication of the column 4-vector Δq. Consider the second term in Eq. (45), in particular $\Delta\Omega$. From Eq. (43) we have

$$\begin{aligned}\Delta\Omega &= \Delta[\mathcal{I}_4^{-1}(q^c\,m\,q)]\\ &= \mathcal{I}_4^{-1}[q^c\,m\,\Delta q + \Delta q^c\,m\,q]\\ &= \mathcal{I}_4^{-1}[q^c\,m\,\Delta q - \Delta q^c\,\{m^c\,q\}]\\ &= \mathcal{I}_4^{-1}[q^c\,m\,\Delta q - \Delta q^c\,\{q^c\,m\}^c]\\ &= \mathcal{I}_4^{-1}[\{(q^c\,m)\,\Delta q\} - \{(q^c\,m)\,\Delta q\}^c]\end{aligned} \tag{46}$$

where in the third line of Eq. (46) we have used the fact the m is pure so that $m = -m^c$. In the fourth line we used the rightmost equality in Eq. (28), namely $(ab)^c = b^c a^c$ with $a \to q^c$ and $b \to m$. In the fifth line we use this rule again, but now with $a \to (q^c m)$ and $b \to \Delta q$. We now observe that the last line in Eq. (46) involves the difference of a quaternion and its conjugate. From the leftmost equality of Eq. (28) we have for an arbitrary quaternion p, $p - p^c = 2(0,\,\boldsymbol{p})$. Using this fact, the last line of Eq. (46) can be written simply as

$$\Delta\Omega = 2\,\mathcal{I}_4^{-1}\{(q^c\,m)\,\Delta q\}. \tag{47}$$

The reason for this is as follows. From the form of $\mathcal{I}_4^{-1}$ in Eq. (44), i.e. the zeros in the 0th row and column, we see that when $\mathcal{I}_4^{-1}$ acts on an arbitary quaternion p we obtain

$$\mathcal{I}_4^{-1}\,(p_0,\,\boldsymbol{p}) = (0, \boldsymbol{\mathcal{I}}^{-1}\boldsymbol{p}) \tag{48}$$

where $\boldsymbol{\mathcal{I}}^{-1} = \mathrm{diag}(I_1^{-1}, I_2^{-1}, I_3^{-1})$ is the usual 3×3 inverse of the moment of inertia matrix in the body frame. Let us identify $p = \{(q^c\,m)\,\Delta q\}$ in

Eq. (46). We can then write the last line of Eq. (46) as

$$\begin{aligned}\Delta\Omega &= \mathcal{I}_4^{-1}[(p_0,\,\boldsymbol{p}) - (p_0,\,-\boldsymbol{p})],\\ &= \mathcal{I}_4^{-1}[(0,\,2\,\boldsymbol{p})],\\ &= 2(0,\boldsymbol{I}^{-1}\cdot\boldsymbol{p}),\\ &= 2\mathcal{I}_4^{-1}(p_0,\,\boldsymbol{p}),\end{aligned} \tag{49}$$

where in the last line we have used Eq. (48). This final result of Eq. (49) is Eq. (47).

The importance of Eq. (47) is that we have Δq isolated on the far rhs of the expression. We can now turn the quaternion multiplication of $q^c m$ on Δq into matrix multiplication to obtain

$$\Delta\Omega = 2\,\mathcal{I}_4^{-1}\cdot Q_L^c\cdot M_L\cdot\Delta q \tag{50}$$

where M_L is the left-matrix formed from $m=(0,\boldsymbol{m})$. Substituting Eq. (50) into Eq. (45) yields the expression for the Jacobian

$$DF(q)\cdot\Delta q = \frac{1}{2}\Delta(q\,\Omega) = \frac{1}{2}[\Omega_R + 2\,Q_L^c\cdot\mathcal{I}_4^{-1}\cdot Q_L\cdot M_L]\cdot\Delta q. \tag{51}$$

Finally the operator $\mathcal{D}$ in Eq. (26) is given explicitly by the expression

$$\begin{aligned}\mathcal{D} &= [\,\Delta t_1 b(-1,r)DF(q^*) - \mathbf{1}\,]\\ &= \Delta t_2\, b(-1,r)\,[\Omega_R + 2\,Q_L\cdot\mathcal{I}_4^{-1}\cdot Q_L^c\cdot M_L] - \mathbf{1}.\end{aligned} \tag{52}$$

where $\Delta t_2 = \Delta t_1/2$. At each stage of the Newton step, the Q_L and Ω_R matrices in the square brackets in Eq. (52) are evaluated at q^*.

Then, eq.(26) becomes:

$$\left\{\Delta t_2 b(-1,r)\left[\Omega_R(q^*) + 2Q_L(q^*)\mathcal{I}_4^{-1}Q_L^c(q^*)M_L\right] - \mathbf{1}\right\}\phi = q^* - \Delta t_1 b(-1,r)F(q^*)$$

or

$$\mathcal{D}\phi = \mathcal{R}_1(q^*) + \mathcal{R}_2$$

where

$$\mathcal{R}_1(q^*) := q^* - \Delta t_1 b(-1,r)F(q^*)$$

and

$$\mathcal{R}_2 = -q^n - \Delta t_1\sum_{k=0}^{r-2} b(k,r)F(q^{n-k})$$

so that $\mathcal{R}_1$ is updated at each iteration while $\mathcal{R}_2$ is fixed.

5 The TIP4P model

The previous discussion applies to an arbitrary molecular calculation for which the moments of inertia and mutual interactions are specified. We give here the specifications for the **TIP4P** model of a water molecule that we are using, together with the calculation of pair-interaction energy, forces and torques.

5.1 Model specifications

The Lennard-Jones potential is given by (with $\mathbf{r}'_i$ the usual spatial coordinate of the center of mass of the i-th molecule with respect to an arbitrary origin **O fixed in space**):

$$V(r') = 4\epsilon \left\{ \left(\frac{\sigma}{r'}\right)^{12} - \left(\frac{\sigma}{r'}\right)^{6} \right\}$$

with corresponding force on particle i due to particle j given by

$$f_{i,j} = 24\frac{\epsilon}{\sigma} \left\{ 2\left(\frac{\sigma}{r'_{i,j}}\right)^{12} - \left(\frac{\sigma}{r'_{i,j}}\right)^{6} \right\} * \frac{\left[\mathbf{r}'_i - \mathbf{r}'_j\right]}{r'_{i,j}}$$

where $r'_{i,j} = |\mathbf{r}'_i - \mathbf{r}'_j|$ (magnitude of vector pointing from j to i). Now ϵ has units of *energy=force* $\times$ L and σ has units of length. Thus ϵ/σ has units of force. Also, torque $\tau := \mathbf{r}' \times \mathbf{f}'$ has units of energy.

We therefore measure:

all distances in units of σ:	$\mathbf{r} = \mathbf{r}'/\sigma$
all energies in units of ϵ:	$V \to V/\epsilon$
all forces in units of ϵ/σ:	$\mathbf{f}(\mathbf{r}') \to \mathbf{f}(\mathbf{r})$
all torques in units of ϵ:	$\tau' = \mathbf{r}' \times \mathbf{f}' \to \mathbf{r} \times \mathbf{f} = \tau$

Thus sigma and epsilon never appear explicitly in our formulations. From Rapaport [1], p. 207 $m_O = 16m_H$ and in reduced units we have the mass of the water molecule as $m_{H_2O} = m_O + 2m_H = 1$. The unit of time is fixed by setting $E = 1/2mv^2$ and writing this as $\epsilon \approx m\sigma^2/t^2$, i.e. $t = \sqrt{m\sigma^2/\epsilon}$. Thus dimensionless time is given by $t = t'/\sqrt{m\sigma^2/\epsilon}$. For water Rapaport [1] uses

ϵ	=	0.155 kcal/mole	= 1.08^{-14} erg/molecule
σ	=	3.154 Angstroms	
unit of time	=	1.66^{-12} s	

In the **body frame** the water molecule is oriented so that the center of mass **CM** is located at the origin, the Oxygen nucleus **O** lies along the negative Z-axis at the point $\mathbf{R}_O = \alpha \mathbf{e}_Z$, and the two Hydrogen nuclei, $\mathbf{H}^{(1)}$ and $\mathbf{H}^{(2)}$ lie on the YZ-plane, at locations $\mathbf{R}_{H^{(1)}} = \delta \mathbf{e}_Y + \gamma \mathbf{e}_Z$ and $\mathbf{R}_{H^{(2)}} = -\delta \mathbf{e}_Y + \gamma \mathbf{e}_Z$ respectively. The TIP4P model represents the electrical properties of the H_2O molecule by placing a negative charge of $-2q$ at the point $\mathbf{R}_{cn} = \beta \mathbf{e}_Z$ and a positive charge of $+q$ at each of the hydrogen nuclei. The dimensionless distances are $\alpha = -0.0206$, $\beta = 0.0274$, $\gamma = 0.165$ and $\delta = 0.240$. Here q is the electron charge. In the dimensionless units employed here we have $q^2/\epsilon_0 = 183.5$. The canonical arrangement in the body frame is shown in Fig. 1.

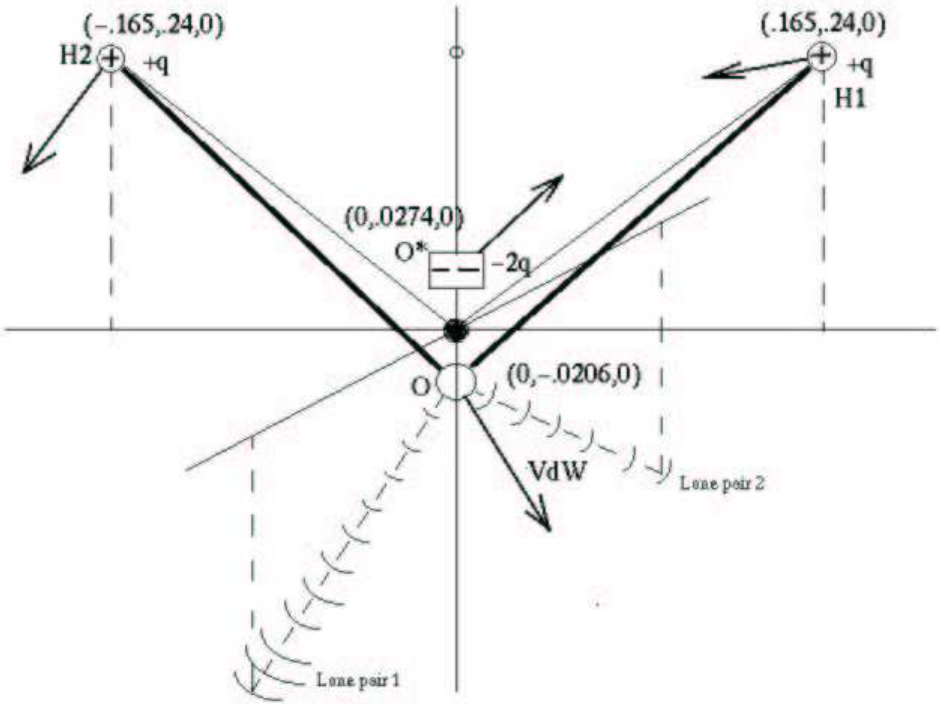

Figure 1. The molecule in the body frame

Also, the principal moments of inertia are:

$$I_Y = .0034 \quad , \quad I_Z = .0064 \quad , \quad I_X^2 = I_Y^2 + I_Z^2 \quad .$$

The ellipsoids of inertia and kinetic energy are shown in Fig. 2.

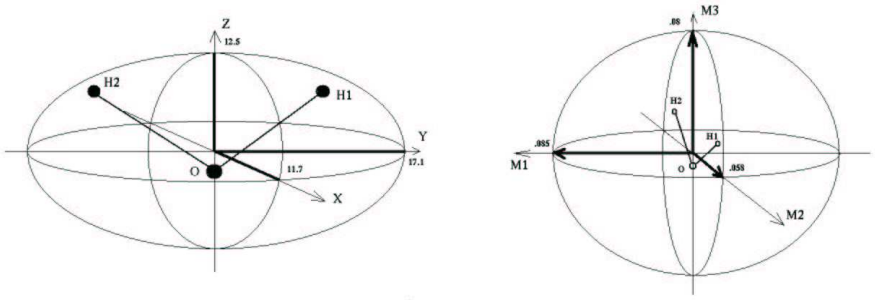

Figure 2. The ellipsoids of inertia and kinetic energy

Using eq.(9), at each instant, all body coordinates are converted to space coordinates, so that

$$\mathbf{r}_{\nu_i} = \mathbf{r}_i + \mathcal{Q}_i(t)\mathbf{R}_\nu \ ,$$

where the index ν ranges over the locations O, cn, $H^{(1)}$ and $H^{(2)}$. Thus, we use $\mathbf{r}_{\nu_i,\mu_j}$ to denote the vector from location μ on the j-th molecule to location ν on the i-th molecule. Then, the total potential between i-th and j-th molecules is given by

$$\begin{aligned} V_{ij} = V_{ji} &= 4\epsilon\left[\left(\frac{\sigma}{r_{O_i,O_j}}\right)^{12} - \left(\frac{\sigma}{r_{O_i,O_j}}\right)^{6}\right] \\ &- 4q^2\frac{1}{r_{cn_i,cn_j}} \\ &+ 2q^2\left[\frac{1}{r_{cn_i,H_j^{(1)}}} + \frac{1}{r_{cn_i,H_j^{(2)}}} + \frac{1}{r_{H_i^{(1)},cn_j}} + \frac{1}{r_{H_i^{(2)},cn_j}}\right] \\ &- q^2\left[\frac{1}{r_{H_i^{(1)},H_j^{(1)}}} + \frac{1}{r_{H_i^{(1)},H_j^{(2)}}} + \frac{1}{r_{H_i^{(2)},H_j^{(1)}}} + \frac{1}{r_{H_i^{(2)},H_j^{(2)}}}\right] \end{aligned}$$

Here and in the sequel an expression of the form Φ_{ij} will denote the computation of some function Φ (force f, potential energy V, torque τ) induced on the molecule **i** due to the molecule **j**.

5.2 Binary interactions: energy and force calculation

The calculation of the mutual forces among the N particles in the system requires the evaluation of $\frac{1}{2}(N-1)^2$ pair interactions. In this calculation care must be taken to count each pair-interaction only once. We introduce, following the convention introduced in the previous section, the force $\mathbf{f}_{\mu_i,\nu_j}$ on the site μ of molecule **i** due to the site ν on molecule **j.** Since in this model the pairwise potentials are all binary and central there holds the relationship:

$$\mathbf{f}_{\mu_i,\nu_j} = -\mathbf{f}_{\nu_j,\mu_i} \ .$$

The algorithm for the pairwise force $\mathbf{f}_{\mu_\mathbf{i},\nu_\mathbf{j}}$ computes all interactions for each (i,j) such that $1 \leq i \leq N-1$ while $i+1 \leq j \leq N$ (taking advantage of the relationship 5.2). Since there are 4 sites on each molecule, (the center for the vanderWaals interaction at O_i, the center of negative charge cn_i and the two hydrogen nuclei $H_i^{(1)}$ and $H_i^{(2)}$), there are a total of 10 interactions among sites (1 vanderWaals and 9 point charge interactions). These pairwise forces are then accumulated, by summing over all interactiond for a given site. The efficient calculation of torques (see next section) requires the computation of the total force on site μ of molecule **i** due to all other molecules, given by the sum:

$$\mathbf{f}_{\mu_i} = \sum_{j \neq i} \sum_{\nu} \mathbf{f}_{\mu_i,\nu_j}$$

while the total force on molecule **i** is found by evaluating one further sum:

$$\mathbf{f}_i = \sum_{\mu} \mathbf{f}_{\mu_i} \ , \ i = 1, \ldots, N \ .$$

5.3 Torque calculation

For the torque calculation we need to evaluate expressions:

$$\tau_i = \sum_{\mu_i} \mathbf{r}_{\mu_i} \times \mathbf{f}_{\mu_i} \ .$$

The simple, planar geometry of the molecule can be used to advantage here. If we recall that the Y and Z axes in the body frame of the i-th

molecule are given by the second and third columns of the rotation matrix, $\mathcal{Q}_{2,i}$ and $\mathcal{Q}_{3,i}$, we have

$$\begin{aligned}
\tau_i &= \mathbf{r_{O_i}} \times \mathbf{f_{O_i}} + \mathbf{r_{cn_i}} \times \mathbf{f_{cn_i}} + \mathbf{r}_{\mathbf{H_i^{(1)}}} \times \mathbf{f}_{\mathbf{H_i^{(1)}}} + \mathbf{r}_{\mathbf{H_i^{(2)}}} \times \mathbf{f}_{\mathbf{H_i^{(2)}}} \\
&= \alpha \mathcal{Q}_{3,i} \times \mathbf{f_{O_i}} + \beta \mathcal{Q}_{\mathbf{3,i}} \times \mathbf{f_{cn_i}} + (\gamma \mathcal{Q}_{\mathbf{3,i}} + \delta \mathcal{Q}_{\mathbf{2,i}}) \times \mathbf{f}_{\mathbf{H_i^{(1)}}} + (\gamma \mathcal{Q}_{\mathbf{3,i}} - \delta \mathcal{Q}_{\mathbf{2,i}}) \times \mathbf{f}_{\mathbf{H_i^{(2)}}} \\
&= \mathcal{Q}_{3,i} \times \left[\alpha \mathbf{f_{O_i}} + \beta \mathbf{f_{cn_i}} + \gamma (\mathbf{f}_{\mathbf{H_i^{(1)}}} + \mathbf{f}_{\mathbf{H_i^{(2)}}}) \right] + \delta \mathcal{Q}_{2,i} \times (\mathbf{f}_{\mathbf{H_i^{(1)}}} - \mathbf{f}_{\mathbf{H_i^{(2)}}})
\end{aligned}$$

The conventions used in calculating the forces and torques are shown in Fig. 3.

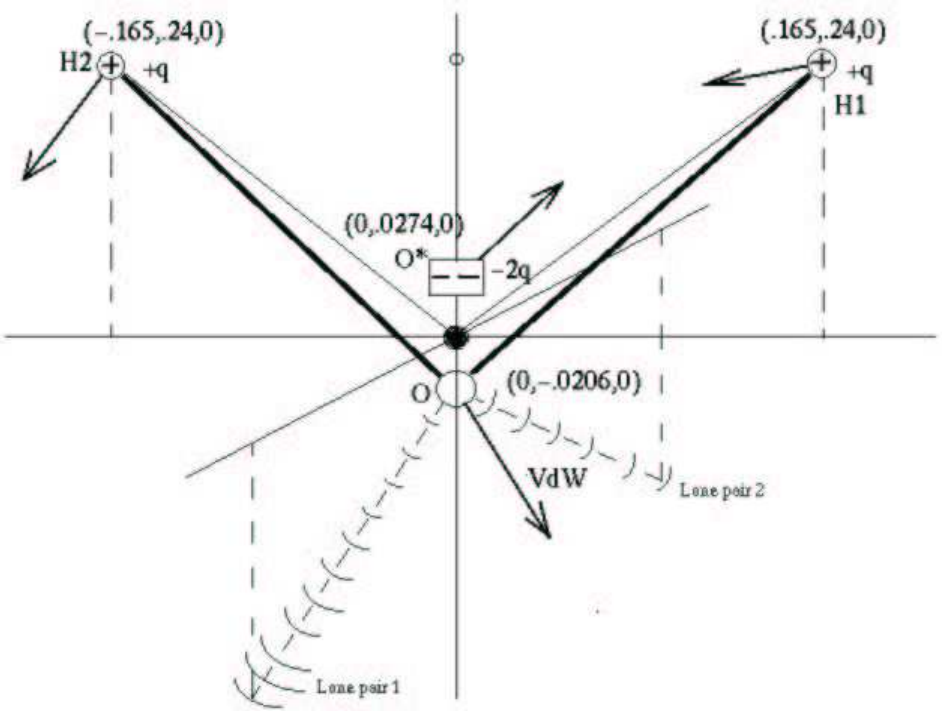

Figure 3. Force calculation conventions

References

[†] alsing@ahpcc.unm.edu

[‡] vageli@math.unm.edu

[1] Rappaport, D.C., *The Art of Molecular Dynamics Simulation* , Cambridge Univ. Press, New York, Chapter 8, pp. 191-221, 1995.

[2] E. A. Coutsias and L. Romero, *The Quaternions with applications to Rigid Body Dynamics*, notes from Dynamical Systems Seminar, UNM Math. Dept. and Center for Advanced Study, Jan. 1999.

[3] In our codes we use the Rapaport [1] notation $q = (q_1, q_2, q_3, q_4) = (\boldsymbol{q}, q_4)$. Although all formulas are easily translated via $q_0 \leftrightarrow q_4$, it is often too easy to make coding mistakes when trying to transcribe formulas from one notation to the other. In these notes, we adhere to the notation $q = (q_0, q_1, q_2, q_3) = (q_0, \boldsymbol{q})$ to agree with the notation in [2].

[4] Allen, M.P., & Tildesley, D.J., *Computer Simulation of Liquids* , Oxford Science Publ., New York, pp. 84-90, 1987.

[5] W. H. Press, S.A. Teukolsky, W.T. Vetterling, and B.R. Flannery, *Numerical Recipes in Fortran*, Section 9.6, *Newton-Raphson Method for Nonlinear Systems of Equations*, Cambridge Univ. Press, (1992).