

Molecular Dynamics Simulation Using Quaternions

D. C. RAPAPORT*

Department of Physics, Bar-Ilan University, Ramat-Gan, Israel

Received June 28, 1984

The use of quaternions as an aid to efficient computer simulation of molecular fluids is discussed. It is shown that the rotational motion of rigid bodies (the molecules) can be described by second order differential equations which are soluble using the same numerical methods used for the translational equations. An alternative derivation of the kinematic equations is also given, together with an outline of the methods used in an extensive molecular dynamics simulation of water. © 1985 Academic Press, Inc.

1. INTRODUCTION

Euler angles have long been used as a convenient means for representing the orientation of a rigid body and as the generalized coordinates in the rotational equations of motion [1, 2]. In systems more complex than those treated in elementary textbooks, for which numerical solution of the equations is the only approach possible, it is known that the Euler angles do not permit the formulation of a numerically stable set of equations [3]. To circumvent this difficulty when carrying out molecular dynamics simulation of rigid-molecule fluids alternative schemes have been proposed; these include redefining the Euler angles whenever a singularity is approached [4], use of holonomic constraints to eliminate explicit mention of rotational motion [5], and replacement of the Euler angles by quaternion components [3, 6]. The last of these schemes provides a particularly simple and computationally efficient reformulation of the dynamical problem.

Our principal purpose is to extend treatments of quaternions and show that the rotational dynamics can be expressed in terms of second-order differential equations, rather than the first-order equations used previously [3]. A consequence of this result is that both rotational and translational equations of motion can then be solved by means of the same numerical techniques. We also describe a very simple derivation of the expressions for the rotation matrix and angular velocity; the derivation follows an existing treatment [2] but has been extended to produce the equations required by the molecular dynamics calculations. It is interesting to observe that despite their usefulness, quaternions have not received wide coverage in present-day mechanics texts (see [3] for some earlier references).

The results described here have been used in extensive water and hydration simulations [7, 8]. The systems contained 343 molecules and were followed for

* Present address: IBM, Department 48B/428, Neighborhood Road, Kingston, N.Y. 12401.

periods of time of up to an order of magnitude longer than in previous work [9, 10]. Two factors contributed to the feasibility of the effort: First, the improved numerical stability of the quaternion form of the equations of motion allowed a significant increase in the time step employed in the numerical solution. Second, the computations were carried out on a high-speed array processor, an especially cost-effective tool for this kind of problem.

2. QUATERNION REPRESENTATION OF ROTATION

The starting point of the derivation is a theorem of Euler [1] — the most general displacement of a rigid body with one point fixed is a rotation about some axis. If \mathbf{c} denotes the direction of the axis and Φ the angle of rotation, then a vector \mathbf{r} fixed to the body is transformed into [2]

$$\mathbf{r}' = \cos \Phi \mathbf{r} + (1 - \cos \Phi)(\mathbf{c} \cdot \mathbf{r}) \mathbf{c} + \sin \Phi \mathbf{c} \times \mathbf{r}. \quad (1)$$

The inverse of (1) is obtained by interchanging \mathbf{r} and \mathbf{r}' and using the angle $-\Phi$. In terms of the usual 3×3 rotation matrix [1, 2]

$$\mathbf{r} = \mathcal{A} \mathbf{r}'; \quad (2)$$

here \mathcal{A} corresponds to a $-\Phi$ rotation, the transpose \mathcal{A}^T to a $+\Phi$ rotation. From (1),

$$\mathcal{A} = \cos \Phi \mathcal{I} + (1 - \cos \Phi) \mathbf{c} \mathbf{c} - \sin \Phi \mathcal{C}, \quad (3)$$

where \mathcal{I} is the unit matrix and

$$\mathcal{C} = \begin{bmatrix} 0 & -c_3 & c_2 \\ c_3 & 0 & -c_1 \\ -c_2 & c_1 & 0 \end{bmatrix}$$

with $\{c_m\}$ the components (or direction cosines) of \mathbf{c} .

The Euler homogeneous parameters are defined as [2]

$$\begin{aligned} q_m &= \cos \Phi/2 & m &= 0, \\ &= c_m \sin \Phi/2 & m &= 1 \cdots 3. \end{aligned} \quad (4)$$

Only three of the four parameters are independent since the normalization condition

$$\sum_{m=0}^3 q_m^2 = 1$$

is automatically satisfied. An alternative form of \mathcal{A} is obtained by combining (3) and (4),

$$\mathcal{A} = (q_0^2 - \mathbf{q}^2) \mathcal{I} + 2\mathbf{q}\mathbf{q} - 2q_0\mathcal{Q}, \quad (5)$$

where

$$\begin{aligned} \mathbf{q} &= \sin \Phi/2 \mathbf{c}, \\ \mathcal{Q} &= \sin \Phi/2 \mathcal{C}. \end{aligned} \quad (6)$$

The full rotation matrix required in the numerical simulations follows immediately from (5),

$$\mathcal{A} = 2 \begin{bmatrix} \frac{1}{2} - q_2^2 - q_3^2 & q_1 q_2 + q_0 q_3 & q_1 q_3 - q_0 q_2 \\ q_1 q_2 - q_0 q_3 & \frac{1}{2} - q_1^2 - q_3^2 & q_2 q_3 + q_0 q_1 \\ q_1 q_3 + q_0 q_2 & q_2 q_3 - q_0 q_1 & \frac{1}{2} - q_1^2 - q_2^2 \end{bmatrix}. \quad (7)$$

This version of the rotation matrix is much more efficiently handled computationally than that based on Euler angles ([1], Eq. 4-46) as it is both free of trigonometric functions and involves fewer multiplications.

The homogeneous parameters $\{q_m\}$ are readily expressed in terms of the Euler angles ϕ , θ , and ψ . From (5),

$$\begin{aligned} \cos \Phi &= \frac{1}{2}(\text{tr } \mathcal{A} - 1) \\ &= \frac{1}{2}[(\cos \theta + 1) \cos(\phi + \psi) + \cos \theta - 1], \end{aligned}$$

where we have used the Euler angle representation of \mathcal{A} . Use of (4) yields q_0 . The remaining parameters, e.g., q_1 , are obtained from (7),

$$q_1 = (\mathcal{A}_{23} - \mathcal{A}_{32})/4q_0$$

and then substituting the components of the Euler angle form of \mathcal{A} . The final expressions are

$$\begin{aligned} q_0 &= \cos(\theta/2) \cos \frac{1}{2}(\phi + \psi) \\ q_1 &= \sin(\theta/2) \cos \frac{1}{2}(\phi - \psi) \\ q_2 &= \sin(\theta/2) \sin \frac{1}{2}(\phi - \psi) \\ q_3 &= \cos(\theta/2) \sin \frac{1}{2}(\phi + \psi). \end{aligned} \quad (8)$$

The same results obtained by a different route appear in [2]; with a minor sign change they are also equivalent to the definitions used in [3]. Suitable linear combinations of the Euler homogeneous parameters produce the more familiar Cayley-Klein parameters [1, 2].

The subsequent derivations, both for the kinematics and dynamics, are most concisely described in terms of quaternions (due to Hamilton) [2]: A quaternion \mathbf{q} is defined whose components are the Euler homogeneous parameters

$$\mathbf{q} = q_0 + q_m e_m; \quad (9)$$

summation over $m = 1 \cdots 3$ is implied by the repeated index. The quantities e_m satisfy

$$\begin{aligned} e_m^2 &= -1 \\ e_m e_n &= \varepsilon_{mnk} e_k, \quad m \neq n, \end{aligned} \quad (10)$$

where $\varepsilon_{mnk} (=0, \pm 1)$ is the Levi-Civita density [1]. The quantities ie_m are equivalent to the Pauli spin matrices [2].

When the product of two quaternions is computed using (9) and (10) it becomes clear that (9) can also be written as

$$\mathbf{q} = q_0 + \mathbf{q}, \quad (11)$$

on the understanding that whenever a vector is used in the definition of a quaternion, each component of the vector is multiplied by the corresponding e_m . The rule for multiplication can then be expressed as

$$\mathbf{q}'' = \mathbf{q}\mathbf{q}',$$

where

$$q_0'' = q_0 q_0' - \mathbf{q} \cdot \mathbf{q}', \quad \mathbf{q}'' = q_0 \mathbf{q}' + q_0' \mathbf{q} + \mathbf{q} \times \mathbf{q}'. \quad (12)$$

The conjugate of \mathbf{q} is

$$\mathbf{q}^* = q_0 - \mathbf{q} \quad (13)$$

and the normalization condition

$$\mathbf{q}\mathbf{q}^* = \mathbf{q}^*\mathbf{q} = q_0 q_0 + q_m q_m = 1 \quad (14)$$

applies.

A rotation operator based on \mathbf{q} is obtained as follows: From (6) and (11)

$$\mathbf{q} = \cos \Phi/2 + \sin \Phi/2 \mathbf{c}. \quad (15)$$

Define quaternions with zero scalar components

$$\mathbf{r} = 0 + \mathbf{r}$$

and similarly \mathbf{r}' . The multiplication rule (12) and the inverse of (1) then lead to

$$\begin{aligned}\mathbf{q}^* \mathbf{r}' \mathbf{q} &= (\cos \Phi/2 - \sin \Phi/2 \mathbf{c}) \mathbf{r}' (\cos \Phi/2 + \sin \Phi/2 \mathbf{c}) \\ &= \cos \Phi \mathbf{r}' + (1 - \cos \Phi)(\mathbf{c} \cdot \mathbf{r}') \mathbf{c} - \sin \Phi \mathbf{c} \times \mathbf{r}' \\ &= \mathbf{r}.\end{aligned}$$

Thus $\mathbf{q} \mathbf{r} \mathbf{q}^*$ ($= \mathbf{r}'$) is completely equivalent to $\mathcal{A}^T \mathbf{r}$ ($= \mathbf{r}'$) — both represent the effect of rotating a vector in space. The inverse relation gives the changes in the components of the vector when the coordinate axes are rotated.

3. EQUATIONS OF MOTION

An expression for the angular velocity $\boldsymbol{\omega}$ in terms of quaternions is readily derived [2]. A vector fixed in the rotating body varies with time as

$$\mathbf{r}(t) = \mathcal{A}(t)^T \mathbf{r}(0),$$

where $\mathcal{A}(0) = \mathcal{I}$. Equivalently,

$$\mathbf{r}(t) = \mathbf{q}(t) \mathbf{r}(0) \mathbf{q}(t)^* \quad (16)$$

in terms of time-dependent quaternions. Differentiating with respect to t gives

$$\begin{aligned}\dot{\mathbf{r}}(t) &= \dot{\mathbf{q}}(t) \mathbf{r}(0) \mathbf{q}(t)^* + \mathbf{q}(t) \mathbf{r}(0) \dot{\mathbf{q}}(t)^* \\ &= \mathbf{g}(t) \mathbf{r}(t) - \mathbf{r}(t) \mathbf{g}(t),\end{aligned} \quad (17)$$

where use has been made of (14) and (16), and

$$\mathbf{g}(t) = \dot{\mathbf{q}}(t) \mathbf{q}(t)^*. \quad (18)$$

The scalar part of (17) is zero, hence [2]

$$\dot{\mathbf{r}} = \mathbf{g} \times \mathbf{r} - \mathbf{r} \times \mathbf{g} = \boldsymbol{\omega} \times \mathbf{r},$$

where the time-dependence is now implicit, and

$$\boldsymbol{\omega} = 2\mathbf{g}. \quad (19)$$

As defined in (19), the components of $\boldsymbol{\omega}$ are expressed in terms of space-fixed coordinate axes (which coincide with the body-fixed axes at $t=0$). The equations of rigid-body dynamics adopt their simplest form when expressed using the body-fixed axes (see below). If $\boldsymbol{\omega}'(t)$ denotes $\boldsymbol{\omega}(t)$ expressed in terms of the instantaneous principal axes (at time t) and $\mathbf{g}'(t)$ the corresponding quaternion, then

$$\mathbf{g}'(t) = \mathbf{q}(t)^* \mathbf{g}(t) \mathbf{q}(t)$$

describes the relation between their components. Use of (18) gives

$$\mathbf{g}' = \mathbf{q}^* \dot{\mathbf{q}} \mathbf{q}^* \mathbf{q} = \mathbf{q}^* \dot{\mathbf{q}}, \quad (20)$$

where the t -dependence is implicit. Hence, by (12),

$$\boldsymbol{\omega}' = 2(q_0 \dot{\mathbf{q}} - \dot{q}_0 \mathbf{q} + \dot{\mathbf{q}} \times \mathbf{q}),$$

or in matrix form:

$$\begin{bmatrix} \omega'_x \\ \omega'_y \\ \omega'_z \\ 0 \end{bmatrix} = 2 \begin{bmatrix} -q_1 & q_0 & q_3 & -q_2 \\ -q_2 & -q_3 & q_0 & q_1 \\ -q_3 & q_2 & -q_1 & q_0 \\ q_0 & q_1 & q_2 & q_3 \end{bmatrix} \begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix}. \quad (21)$$

The last of the four equations in (21) follows from $d/dt(\mathbf{q}^* \mathbf{q}) = 0$ and is included to orthogonalize the matrix and hence make the inverse relation between $\{\dot{q}_m\}$ and $\boldsymbol{\omega}'$ trivially obtainable. A result equivalent to (21) appears in [3]. Eq. (21) should be contrasted with the expression relating $\boldsymbol{\omega}'$ to the Euler angles and their derivatives ([1, Eqs. 4-103]); the matrix involved is singular at $\theta = 0$ and π and is a source of numerical instability when θ approaches these values [3]. The orthogonality of the matrix in (21) eliminates this problem.

The rotational dynamics of rigid bodies are based on the Euler equation

$$I_x \dot{\omega}'_x = N_x + (I_y - I_z) \omega'_y \omega'_z \quad (22)$$

and its cyclic permutations [1]; here I_x, \dots , are the principal moments of inertia and N_x, \dots , the components of the external torque acting on the body relative to the principal axes. The Euler equations can be expressed wholly in terms of the homogeneous parameters and their derivatives. The resulting differential equations are second order and do not involve $\boldsymbol{\omega}'$, whereas the end results of the earlier analysis [3] were the first-order equations (21) and (22). As stated in the Introduction the advantage of using second-order equations is that they can be solved by the same numerical techniques used for the translational equations of motion; under similar conditions (step size and order) the accuracy of the solution will equal that obtained using first-order equations.

The derivative of (20) is

$$\dot{\mathbf{g}}' = \dot{\mathbf{q}}^* \dot{\mathbf{q}} + \mathbf{q}^* \ddot{\mathbf{q}};$$

premultiplication by \mathbf{q} and rearranging leads to

$$\ddot{\mathbf{q}} = \mathbf{q}(\dot{\mathbf{g}}' - \dot{\mathbf{q}}^* \dot{\mathbf{q}}),$$

and since the scalar part of \mathbf{g}' is zero,

$$\ddot{\mathbf{q}} = \frac{1}{2}\mathbf{q} \left(\dot{\omega}' - 2 \sum_{m=0}^3 \dot{q}_m^2 \right). \quad (23)$$

If the matrix appearing in (21) is denoted by \mathcal{W} then (23) can be expressed as

$$\begin{bmatrix} \ddot{q}_0 \\ \ddot{q}_1 \\ \ddot{q}_2 \\ \ddot{q}_3 \end{bmatrix} = \frac{1}{2}\mathcal{W}^T \begin{bmatrix} \dot{\omega}'_x \\ \dot{\omega}'_y \\ \dot{\omega}'_z \\ -2\sum \dot{q}_m^2 \end{bmatrix}. \quad (24)$$

The components of $\dot{\omega}'$ in (24) can be replaced by the Euler equations (22), and the components of ω' appearing there by the expressions in (21). The resulting equations of motion are of the form

$$\ddot{q}_m = f_m(\{q_n\}, \{\dot{q}_n\}, \mathbf{N}), \quad m = 0 \cdots 3. \quad (25)$$

A complete description of the motion requires the equations for the motion of the centre of mass as well, namely

$$\ddot{r}_m = a_m(\{r_n\}, \mathbf{F}), \quad m = 1 \cdots 3, \quad (26)$$

where \mathbf{F} is the total force acting on the body.

4. APPLICATION

The equations derived above have been used in a series of molecular dynamics simulations of liquid water and effects associated with hydration. The results of these simulations have been detailed elsewhere [7, 8]; our purpose here is to outline the key technical aspects of the calculations. The recipe is completely general in that it can be applied to any fluid of rigid molecules whose dynamics are described by the Euler equations (22) and whose interactions can be described in terms of forces between pairs of sites on different molecules. A well-known example of a fluid of this kind is the MCY-CI model [11] used in the water simulations. A description of the main features of the calculation follows.

Initialization. The coordinates and velocities — both translational and rotational — are given suitable initial values; the molecules can for example be placed on the sites of an imaginary lattice, with randomly chosen orientations and velocities. Euler angles may be used to describe the initial orientation but they are then converted to the homogeneous parameters $\{q_m\}$ using (8) and not referenced further.

Time step. The temporal evolution of the system is followed by numerically solving the equations of motion (25, 26); the actual evolution occurs in a series of discrete time steps. A fifth-order predictor-corrector method [12, 13] is used to solve

the differential equations; predictor–corrector methods have the advantage over those of the Runge–Kutta type that the force calculations — the major part of the computation — need only be carried out once per time step. The method has a relatively minor disadvantage in that considerably more information concerning the system at previous time steps (either just the preceding step or several earlier steps depending on the particular method [12]) must be retained; this is not an important factor in the present work. In order to advance the system over a single time step the following sequence of calculations is carried out:

(1) Tentative estimates of molecular coordinates are obtained by combining information from earlier time steps using the appropriate predictor formulae [13].

(2) The coordinates of the interaction sites on each molecule in a space-fixed coordinate frame are computed using the rotation matrix (7) appropriate to the molecule.

(3) The combined forces acting on each interaction site are determined using the specific model potential function.

(4) The total force and torque on each molecule are determined; the torque components are converted to the principal axis coordinates of the molecule using the rotation matrix again.

(5) The corrector formulae [13] are applied to (25, 26) to produce revised estimates for the coordinates and velocities.

If periodic boundaries are used then their effects must be included when updating the translational coordinates and when determining the distances between interaction sites during the force calculations. Finally, since the numerical methods do not fully preserve the normalization condition for \mathbf{q} , updates to the rotational coordinates should include rescaling to maintain the normalization.

Measurement. At regular intervals various measurements are carried out and the data recorded to allow subsequent calculation of thermodynamic properties, mean temperature, radial distribution functions, angular distribution functions, etc. Quantities based on angles, such as the dipole–dipole correlation, are easily computed from the $\{q_n\}$ —if the dipole moment lies along the molecular z -axis then the scalar product of two such moments is simply the scalar product of the bottom rows of the \mathcal{A} -matrices for the two molecules.

Temperature adjustment. An initial phase of each simulation run is set aside to allow the system to equilibrate at the desired temperature. This is achieved by repeatedly scaling the translational and angular velocities until the mean kinetic energy settles down at the appropriate value. Due to the error inherent in the numerical solution method and the use of a finite-range cutoff in the potential function there is a small but persistent drift away from the equilibrium temperature throughout the course of the simulation; the accepted means of suppressing this drift is to monitor the kinetic temperature and, whenever the temperature deviation exceeds a certain threshold (or, equivalently, at regular intervals), rescale the velocities to restore the correct temperature. One of the factors affecting the rate of

temperature drift is the step size; use of quaternions permits a considerably larger step size (e.g., by a factor of five) for a given drift rate [3, 7].

Organization. Simulations of this kind often entail large quantities of computer time (measured in terms of even tens of hours). The total run is then normally divided into a number of separate computer tasks that are run in sequence (a scheme known as checkpointing). One design goal is to organize the calculation so that the data describing the current and recent states of the fluid, together with information concerning the status of the simulation and the results of partially completed measurements, are easily stored at the conclusion of one task and retrieved at the beginning of the next.

REFERENCES

1. H. GOLDSTEIN, "Classical Mechanics," Addison-Wesley, Reading, Mass., 1950.
2. H. C. CORBEN AND P. STEHLE, "Classical Mechanics," 2nd ed., Wiley, New York, 1960.
3. D. J. EVANS, *Mol. Phys.* **34** (1977), 317; D. J. EVANS AND S. MURAD, *ibid.* **34** (1977), 327.
4. J. BAROJAS, D. LEVESQUE, AND B. QUENTREC, *Phys. Rev. A* **7** (1973), 1092.
5. J. P. RYCKAERT, G. CICCOTTI, AND H. J. C. BERENDSEN, *J. Comput. Phys.* **23** (1977), 327.
6. G. S. PAWLEY, *Mol. Phys.* **43** (1981), 1321.
7. D. C. RAPAPORT AND H. A. SCHERAGA, *Chem. Phys. Lett.* **78** (1981), 491; *J. Phys. Chem.* **86** (1982), 873.
8. D. C. RAPAPORT, *Mol. Phys.* **48** (1983), 23; **50** (1983), 1151.
9. F. H. STILLINGER AND A. RAHMAN, *J. Chem. Phys.* **60** (1974), 1545.
10. A. GEIGER, A. RAHMAN, AND F. H. STILLINGER, *J. Chem. Phys.* **70** (1979), 263.
11. O. MATSUOKA, E. CLEMENTI, AND M. YOSHIMINE, *J. Chem. Phys.* **64** (1976), 1351.
12. C. W. GEAR, "Numerical Initial Value Problems in Ordinary Differential Equations," Prentice-Hall, Englewood Cliffs, N.J., 1971.
13. D. BEEMAN, *J. Comput. Phys.* **20** (1976), 130.