SETTLE: An Analytical Version of the SHAKE and RATTLE Algorithm for Rigid Water Models

Shuichi Miyamoto and Peter A. Kollman*

Department of Pharmaceutical Chemistry, University of California, San Francisco, California 94143

Received 9 December 1991; accepted 25 February 1992

An analytical algorithm, called SETTLE, for resetting the positions and velocities to satisfy the holonomic constraints on the rigid water model is presented. This method is still based on the Cartesian coordinate system and can be used in place of SHAKE and RATTLE. We implemented this algorithm in the SPASMS package of molecular mechanics and dynamics. Several series of molecular dynamics simulations were carried out to examine the performance of the new algorithm in comparison with the original RATTLE method. It was found that SETTLE is of higher accuracy and is faster than RATTLE with reasonable tolerances by three to nine times on a scalar machine. Furthermore, the performance improvement ranged from factors of 26 to 98 on a vector machine since the method presented is not iterative. © 1992 by John Wiley & Sons, Inc.

INTRODUCTION

Molecular dynamics (MD) simulations have become an important tool in the field of computational chemistry.[1] This approach has been applied to investigate relatively simple systems,[2] as well as complex macromolecular systems.[3] Because of recent advances in computing power, longer simulations with larger systems have been used to obtain robust results for slowly converging statistical mechanical averages.[4] These systems often include thousands of solvent molecules either in a periodic box or in a spherical cap using a relatively small integration time step of 1 fs.

The solvent models most frequently used are those of TIP3P and SPC water proposed by Jorgensen[5] and Berendsen,[6] respectively. In the MD simulations with these water models, holonomic constraints are usually employed to keep the bond lengths at their equilibrium values. This is done for TIP3P because this water model has been parameterized based on Monte Carlo simulations using rigid molecular geometries. Another practical reason for using these constraints is that they provide the possibility of using a larger time step size, and, hence, the ratio of simulation time to computational time is improved. If the number of water molecules is large, the computation time spent on the satisfaction of bond constraints is significant. Our experience with simulations using the TIP3P water model shows that typically 10–20% of the total calculation time is due to this part of the simulation. Mertz et al.[7] reported that 25% of computation time is spent on the constraint procedure for a pure solvent system. In addition, in the case of recent intensive simulations with a time increment of 1 fs, no constraints appear to be necessary for the solute molecules. Therefore, constraining rigid water molecules quickly and effectively will speed MD simulations.

For rigid models, either Euler angles[8] or quaternions[9] can be used to represent the configuration. An alternative is to use the methods SHAKE and RATTLE devised by Ryckaert et al.[10] and Andersen,[11] respectively. These methods retain the simplicity of using Cartesian coordinates and avoid many of the complications of Euler equations and quaternions, while incorporating the effects of the restrained geometry of the molecule.

In this paper we present an analytical algorithm for resetting the positions and velocities to satisfy the holonomic constraints on rigid water molecules. Since this method is still based on the Cartesian coordinate system, it can be used in place of SHAKE and RATTLE incorporated in the Verlet algorithm[12] and the velocity Verlet algorithm,[13] respectively. Thus, the approach described here can be straightforwardly integrated into most standard MD packages, which employ the SHAKE or RATTLE algorithm in their constraint routines. In fact we have implemented this algorithm, referred to as "SETTLE", in the SPASMS package of molecular mechanics and dynamics.[14]* Several series of MD simulations on a VAX8650 and an FPS500 were

*Author to whom all correspondence should be addressed.

*SPASMS is an acronym for San Francisco Package of Applications for the Simulation of Macromolecular Systems, a collection of computer programs to be distributed by the University of California at San Francisco.

CCC 0192-8651/92/080952-11$04.00

carried out to examine the performance of the new algorithm.

ALGORITHM FOR SETTLE

In this section we will discuss the procedure for integrating the equations of motion of a system with holonomic constraints. Then we derive SETTLE, leaving more detailed equations to the appendices.

We restrict our attention to the case of bond length constraints, which are expressed as

$$|\mathbf{r}_{ij}|^2 - d_{ij}^2 = 0 \tag{1}$$

where $\mathbf{r}_{ij} = \mathbf{r}_j - \mathbf{r}_i$ is the vector associated with the rigid bond between atoms i and j whose position vectors are $\mathbf{r}_i$ and $\mathbf{r}_j$, respectively, and d_{ij} is the constraint value. In addition, the time derivatives of the above equation give constraints on the velocities, that is,

$$\mathbf{r}_{ij} \cdot \mathbf{v}_{ij} = 0 \tag{2}$$

where $\mathbf{v}_{ij} = \mathbf{v}_j - \mathbf{v}_i$ is the relative velocity acting on the atoms i and j constituting the rigid bond and $\mathbf{v}_i$ is the velocity of atom i. In a geometrical sense, this means the net velocity acting on the rigid bond must be perpendicular to that bond. In other words, the component of the relative velocity along the bond should be zero. The above constraint eq. (2) can be applied to the velocity Verlet algorithm while the position constraint is applicable to all algorithms using the Cartesian coordinate system.

The coordinate constraints are usually implemented into an MD algorithm by first taking an integration step in the absence of any constraint forces and then fulfilling these constraints by adding displacement vectors. When integrating the equations of motion in Cartesian coordinates, the solution of equation for the constrained motion is written as

$$\mathbf{r}_i(t_0 + \delta t) = \mathbf{r}_i^0(t_0 + \delta t) + \delta\mathbf{r}_i(t_0) \tag{3}$$

where $\mathbf{r}_i^0(t_0 + \delta t)$ is the position vector after an unconstraint step (δt) and $\delta\mathbf{r}_i(t_0)$ is the displacement vector required to satisfy the distance restraints,

$$\begin{aligned}\delta\mathbf{r}_i(t_0) &= 1/2 \cdot (\delta t)^2/m_i \cdot \Sigma\mathbf{g}_{ij}(t_0) \\ &= 1/2 \cdot (\delta t)^2/m_i \cdot \Sigma\lambda_{ij}(t_0)\mathbf{r}_{ij}(t_0).\end{aligned} \tag{4}$$

In the above equation, Lagrangian multipliers, $\lambda_{ij}(t_0)$ ($\lambda_{ij} = \lambda_{ji}$) are to be chosen so that the constraint eq. (1) are satisfied at time $t_0 + \delta t$. A physical picture of this is that constraint forces $[\mathbf{g}_{ij}(t_0)]$ of equal magnitudes and opposite orientations are applied to the atoms i and j and are directed along the bond vectors $[\mathbf{r}_{ij}(t_0)]$ at time t_0. In the conventional method a set of quadratic equations for λ_{ij} is obtained by substitution of eqs. (3) and (4) into eq. (1). The solution to the quadratic equations is given by first solving them in their linear form and subsequently iterating them until all the constraints are fulfilled to within an acceptable tolerance. The linear equations are solved by either matrix inversion or by the well-known SHAKE method.[10] The computing time required for these procedures depends on the tolerance.

The velocity constraints are also implemented into the velocity Verlet algorithm by first calculating unconstrained velocities and then satisfying the restraints by adding correction velocities. The solution of the constrained velocity is therefore written as

$$\mathbf{v}_i(t_0 + \delta t) = \mathbf{v}_i^0(t_0 + \delta t) + \delta\mathbf{v}_i(t_0 + \delta t) \tag{5}$$

where $\mathbf{v}_i^0(t_0 + \delta t)$ is the velocity which would have resulted in without the constraints at time $t_0 + \delta t$ while those constraints at time t_0 are already taken into account.* $\delta\mathbf{v}_i(t_0 + \delta t)$ is the correction velocity necessary to satisfy the constraints,

$$\begin{aligned}\delta\mathbf{v}_i(t_0 + \delta t) &= \delta t/2m_i \cdot \Sigma\mathbf{g}_{ij}(t_0 + \delta t) \\ &= \delta t/2m_i \cdot \Sigma\lambda_{ij}(t_0 + \delta t)\mathbf{r}_{ij}(t_0 + \delta t)\end{aligned} \tag{6}$$

The constraint forces $[\mathbf{g}_{ij}(t_0 + \delta t)]$ are directed along the bond vectors $[\mathbf{r}_{ij}(t_0 + \delta t)]$ at time $t_0 + \delta t$ and are chosen so that the velocities fulfill the constraint eq. (2) exactly at time $t_0 + \delta t$. The iterative solution of these equations, RATTLE, has been given by Andersen.[11]

One of the characteristics of the new algorithm SETTLE is to directly determine $\mathbf{r}_i(t_0 + \delta t)$ by use of quasi-Euler angles without calculating constraint forces, $\mathbf{g}_{ij}(t_0)$ explicitly. In the case of the rigid water molecule, three bonds are constrained, i.e., two real O—H bonds with equal lengths and one fictitious H—H bond. In the following we explain the SETTLE algorithm by taking a water molecule H_2O as a triangle ABC. Oxygen and hydrogens corresponds to A, B, and C, respectively, with the mass of m_a and $m_b = m_c$. (In the case of HOD, m_b is not equal to m_c, resulting in simple modifications of the equations given in the appendices.) The canonical triangle with three sides of fixed lengths is uniquely determined and we define such a unique triangle as $\Delta a_0 b_0 c_0$, referring to its center of mass as d_0.

Suppose $\Delta A_0 B_0 C_0$ is a triangle at time t_0, $\Delta A_1 B_1 C_1$ is the triangle that would have been reached at time $t_0 + \delta t$ in the absence of any constraints, $\Delta A_3 B_3 C_3$ is the corresponding one after applying constraints and $\mathbf{v}_{A_3}^0$, $\mathbf{v}_{B_3}^0$, and $\mathbf{v}_{C_3}^0$, and $\mathbf{v}_{A_3}$,

* $\begin{aligned}\mathbf{v}_i(t_0 + \delta t) &= \mathbf{v}_i^0(t_0 + \delta t) + \delta\mathbf{v}_i(t_0 + \delta t) \\ &= \mathbf{v}_i(t_0 + 1/2 \cdot \delta t) + 1/2 \cdot \delta t\mathbf{f}(t_0 + \delta t) \\ &\quad + \delta\mathbf{v}_i(t_0 + \delta t) \\ &= \mathbf{v}_i(t_0) + 1/2 \cdot \delta t\mathbf{f}(t_0) + 1/2 \cdot (\delta t)^2/m_i \cdot \Sigma\mathbf{g}_{ij}(t_0) \\ &\quad + 1/2 \cdot \delta t\mathbf{f}(t_0 + \delta t) + \delta\mathbf{v}_i(t_0 + \delta t)\end{aligned}$

where $1/2 \cdot (\delta t)^2/m_i \cdot \Sigma\mathbf{g}_{ij}(t_0)$ is the velocity correction term due to the constraint forces at time t_0.

$\mathbf{v}_{B_3}$, and $\mathbf{v}_{C_3}$ are velocities of vertices before and after adding correction velocities at time $t_0 + \delta t$, respectively, as shown in Figure 1a. If we define the plane that includes $\Delta A_0B_0C_0$ as π_0, three planes that are parallel to π_0 and include one of each apex of $\Delta A_1B_1C_1$ can be designated as π_A, π_B, and π_C, respectively (Fig. 1b). Since the constraint forces are directed along the bond at time t_0, each of the displacement vectors is on π_A, π_B, and π_C, respectively, with the resulting positions of the vertices A_3, B_3, and C_3 lying on each of those planes, as shown in Figure 1b. The centers of mass of $\Delta A_1B_1C_1$ and $\Delta A_3B_3C_3$, D_1 and D_3, coincide with each other because the sum of the constraint forces is zero by definition. $\Delta a_0b_0c_0$ can be overlapped with $\Delta A_3B_3C_3$ through the proper rotation if the center of mass of $\Delta a_0b_0c_0$, d_0 is located in that of $\Delta A_3B_3C_3$, D_3.

Here, as shown in Figure 1c, let us introduce an alternative orthogonal coordinate system $X'Y'Z'$ in which the origin coincides with $D_1 = D_3 = d_0$, the $X'Y'$ plane is parallel to π_0, and the $Y'Z'$ plane includes A_1. Since π_0 and A_1 are available after an unconstrained step, the matrix of orthogonal transformation is uniquely defined. If we place $\Delta a_0b_0c_0$ on the $X'Y'$ plane so that d_0 is situated at the origin of $X'Y'Z'$ system and a_0 is on the Y' axis (Fig. 2a), then $\Delta A_3B_3C_3$ can be expressed as a rotamer of $\Delta a_0b_0c_0$ relative to the origin using the following procedure: After the initial rotation of $\Delta a_0b_0c_0$ by ψ around the Y' axis, $\Delta a_1b_1c_1$ is obtained as shown in Figure 2b. $\Delta a_1b_1c_1$ is then rotated by ϕ around the X' axis to produce $\Delta a_2b_2c_2$ (Fig. 2c), followed by the final rotation by θ around the Z' axis to give $\Delta a_3b_3c_3$ which should overlap with $\Delta A_3B_3C_3$ (Fig. 2d). Our approach is to determine ψ, ϕ, and θ instead of λ. Notice that the Z' coordinates do not change after the rotation by θ ($Z'_{a_3} = Z'_{a_2}, \ldots$) and that the Z' coordinates of vertices A_1, B_1 and C_1 are the same as those of A_3, B_3, and C_3, respectively ($Z'_{A_3} = Z'_{A_1}, \ldots$). Based on the above, as will be shown in Appendix A, ψ and ϕ can be determined uniquely from the Z' coordinates of A_1, B_1, and C_1. That is, the positions of a_2, b_2, and c_2 can be obtained.

In the next step, θ can be calculated analytically by using the condition that constraint forces directed along the bond at time t_0 are of equal magnitudes and opposite orientations, and the detailed derivation will be presented in Appendix A. θ can be then used to obtain the coordinates of $\Delta a_3b_3c_3 \equiv \Delta A_3B_3C_3$, which can be used to calculate the constraint forces as appropriate. Although three Euler-like rotation angles are used in the above, the position of each apex is expressed explicitly in the way shown in Appendix A and, hence, this method is considered to be fully based on the Cartesian coordinate system.

In the case of velocity constraints, the constraint forces are directed along the bonds defined by $\Delta A_3B_3C_3$ at time $t_0 + \delta t$. Since only $\Delta A_3B_3C_3$ and time $= t_0 + \delta t$ are involved in the argument on the velocity constraints, subscripts of vertices and indications of time will be omitted in the following. Let $\mathbf{e}_{AB}$, $\mathbf{e}_{BC}$, and $\mathbf{e}_{CA}$ be the unit vectors of $\overrightarrow{AB}$, $\overrightarrow{BC}$, and $\overrightarrow{CA}$. Then, bond vectors and constraint forces are written as

$$\mathbf{r}_{AB} = r_{AB}\mathbf{e}_{AB},\ \mathbf{r}_{BC} = r_{BC}\mathbf{e}_{BC},\ \mathbf{r}_{CA} = r_{CA}\mathbf{e}_{CA}$$

$$\mathbf{g}_{AB} = -\mathbf{g}_{BA} = \lambda_{AB}\mathbf{r}_{AB} = \lambda_{AB}r_{AB}\mathbf{e}_{AB} = \tau_{AB}\mathbf{e}_{AB}$$

$$\mathbf{g}_{BC} = -\mathbf{g}_{CB} = \lambda_{BC}\mathbf{r}_{BC} = \lambda_{BC}r_{BC}\mathbf{e}_{BC} = \tau_{BC}\mathbf{e}_{BC}$$

$$\mathbf{g}_{CA} = -\mathbf{g}_{AC} = \lambda_{CA}\mathbf{r}_{CA} = \lambda_{CA}r_{CA}\mathbf{e}_{CA} = \tau_{CA}\mathbf{e}_{CA}$$

where r_{ij} is the length of side ij and $\tau_{AB} = \lambda_{AB}r_{AB}$, $\tau_{BC} = \lambda_{BC}r_{BC}$, and $\tau_{CA} = \lambda_{CA}r_{CA}$ are Lagrangian multipliers (Fig. 3). Equation (5) can be expressed as

$$\begin{aligned}\mathbf{v}_A &= \mathbf{v}_A^0 + \delta t/2m_a \cdot \mathbf{g}_A \\ &= \mathbf{v}_A^0 + \delta t/2m_a \cdot (\mathbf{g}_{AB} + \mathbf{g}_{AC}) \\ &= \mathbf{v}_A^0 + \delta t/2m_a \cdot (\tau_{AB}\mathbf{e}_{AB} - \tau_{CA}\mathbf{e}_{CA}) \quad (7\text{-}1)\end{aligned}$$

$$\begin{aligned}\mathbf{v}_B &= \mathbf{v}_B^0 + \delta t/2m_b \cdot \mathbf{g}_B \\ &= \mathbf{v}_B^0 + \delta t/2m_b \cdot (\mathbf{g}_{BC} + \mathbf{g}_{BA}) \\ &= \mathbf{v}_B^0 + \delta t/2m_b \cdot (\tau_{BC}\mathbf{e}_{BC} - \tau_{AB}\mathbf{e}_{AB}) \quad (7\text{-}2)\end{aligned}$$

$$\begin{aligned}\mathbf{v}_C &= \mathbf{v}_C^0 + \delta t/2m_c \cdot \mathbf{g}_C \\ &= \mathbf{v}_C^0 + \delta t/2m_c \cdot (\mathbf{g}_{CA} + \mathbf{g}_{CB}) \\ &= \mathbf{v}_C^0 + \delta t/2m_c \cdot (\tau_{CA}\mathbf{e}_{CA} - \tau_{BC}\mathbf{e}_{BC}). \quad (7\text{-}3)\end{aligned}$$

Substituting eq. (7) into eq. (2), we can write the constraint relation applied to the bond AB as

$$\begin{aligned}&\mathbf{r}_{AB} \cdot \mathbf{v}_{AB} \\ &= r_{AB}\mathbf{e}_{AB} \cdot [\mathbf{v}_B^0 + \delta t/2m_b \cdot (\tau_{BC}\mathbf{e}_{BC} - \tau_{AB}\mathbf{e}_{AB}) \\ &\quad - \mathbf{v}_A^0 - \delta t/2m_a \cdot (\tau_{AB}\mathbf{e}_{AB} - \tau_{CA}\mathbf{e}_{CA})] \\ &= r_{AB}\mathbf{e}_{AB} \cdot [\mathbf{v}_B^0 - \mathbf{v}_A^0 - \delta t/2 \cdot (1/m_a + 1/m_b)\tau_{AB}\mathbf{e}_{AB} \\ &\quad + \delta t/2m_b \cdot \tau_{BC}\mathbf{e}_{BC} + \delta t/2m_a \cdot \tau_{CA}\mathbf{e}_{CA}] \\ &= r_{AB}[\mathbf{e}_{AB} \cdot \mathbf{v}_B^0 - \delta t/2 \cdot (1/m_a + 1/m_b)\tau_{AB} \\ &\quad - \delta t/2m_b \cdot \tau_{BC}\cos B - \delta t/2m_a \cdot \tau_{CA}\cos A] \\ &= 0\end{aligned}$$

$$\therefore\ \delta t(m_a + m_b)\tau_{AB} + \delta t \cdot m_a\tau_{BC}\cos B + \delta t \cdot m_b\tau_{CA}\cos A = 2m_am_b\mathbf{e}_{AB} \cdot \mathbf{v}_{AB}^0 \quad (8\text{-}1)$$

where $\cos A$ and $\cos B$ are cosines of the apex angles of A and B. Similarly, denoting a cosine of the apex angle of C by $\cos C$ we obtain

$$\delta t(m_b + m_c)\tau_{BC} + \delta t \cdot m_b\tau_{CA}\cos C + \delta t \cdot m_c\tau_{AB}\cos B = 2m_bm_c\mathbf{e}_{BC} \cdot \mathbf{v}_{BC}^0 \quad (8\text{-}2)$$

$$\delta t(m_c + m_a)\tau_{CA} + \delta t \cdot m_c\tau_{AB}\cos A + \delta t \cdot m_a\tau_{BC}\cos C = 2m_cm_a\mathbf{e}_{CA} \cdot \mathbf{v}_{CA}^0 \quad (8\text{-}3)$$

Since the constraint relations of eq. (8) are simultaneous linear equations with respect to the variables, τ_{AB}, τ_{BC}, and τ_{CA}, they can be solved by use of

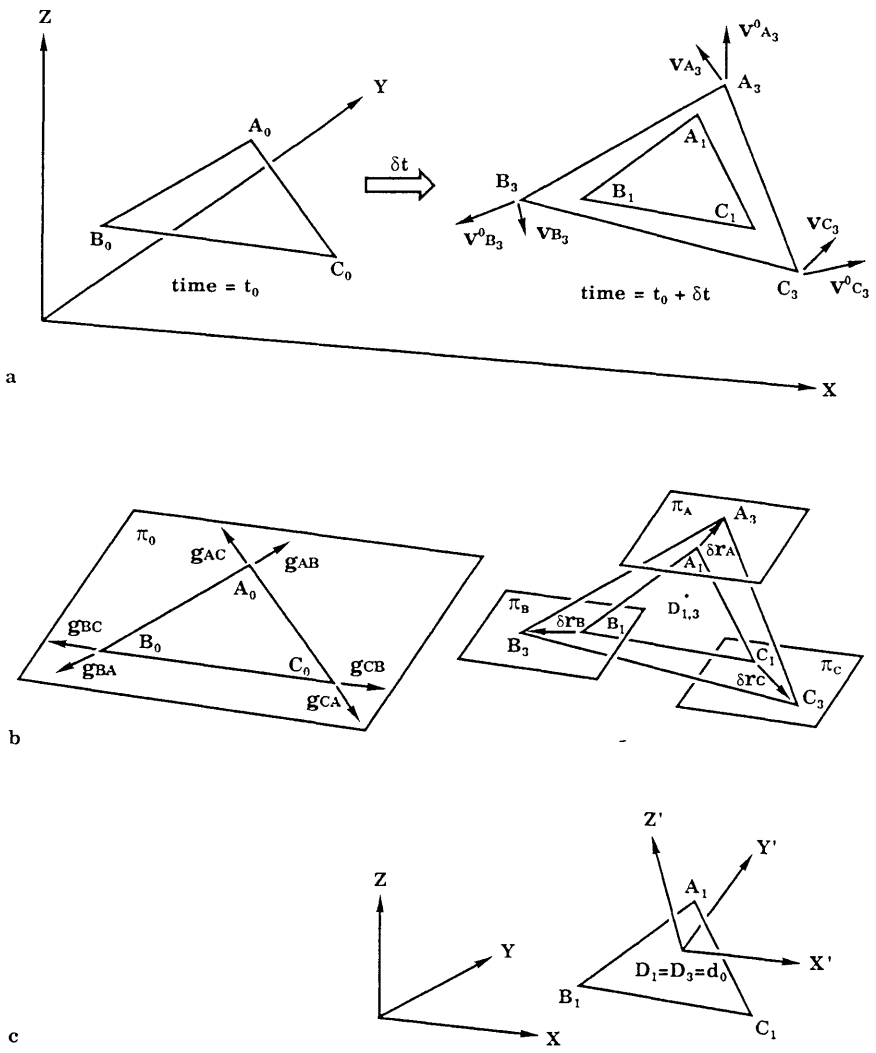

Figure 1. Schematic drawing of the geometric basis of the SETTLE method. (a) $\Delta A_0B_0C_0$ is a triangle at time t_0, $\Delta A_1B_1C_1$ is the triangle that would have been reached at time $t_0 + \delta t$ in the absence of any constraints, and $\Delta A_3B_3C_3$ is the corresponding one after applying constraints. $\mathbf{v}^0_{A_3}$, $\mathbf{v}^0_{B_3}$, and $\mathbf{v}^0_{C_3}$ and $\mathbf{v}_{A_3}$, $\mathbf{v}_{B_3}$, and $\mathbf{v}_{C_3}$ are velocities of vertices before and after adding correction velocities at time $t_0 + \delta t$, respectively. (b) The plane π_0 includes $\Delta A_0B_0C_0$. Three planes, π_A, π_B, and π_C, are parallel to π_0. Each of them includes one of the apexes of A_1, B_1, and C_1, respectively. The constraint forces, $\mathbf{g}_{ij}$, are directed along the bond at time t_0. Each of the displacement vectors, $\delta\mathbf{r}_i$, is therefore on π_A, π_B, or π_C, respectively, with the resulting positions of the vertices A_3, B_3, and C_3 lying on each of those planes. D_1 and D_3 are centers of mass of $\Delta A_1B_1C_1$ and $\Delta A_3B_3C_3$, respectively. They coincide with each other because the sum of the constraint forces is zero by definition. (c) An alternative orthogonal coordinate system $X'Y'Z'$ with its origin at $D_1 = D_3 = d_0$. The $X'Y'$ plane is parallel to π_0 and the $Y'Z'$ plane includes A_1. Since π_0 and A_1 are available after an unconstrained step, the matrix of orthogonal transformation is uniquely defined.

Cramer's rule, as shown in Appendix B. Substituting the τ_{AB}, τ_{BC}, and τ_{CA} into eq. (7), we get the constrained velocities, $\mathbf{v}_A$, $\mathbf{v}_B$, and $\mathbf{v}_C$.

RESULTS AND DISCUSSION

We incorporated the above described algorithm for rigid TIP3P water into the SPASMS package which uses RATTLE with the velocity Verlet scheme as its standard constraint algorithm. Since RATTLE includes SHAKE, a comparison between SETTLE and RATTLE were carried out. Several series of MD simulations mainly on a pure water system were performed to confirm the correct behavior of the algorithm, as well as to examine its performance in comparison with that of conventional RAT-

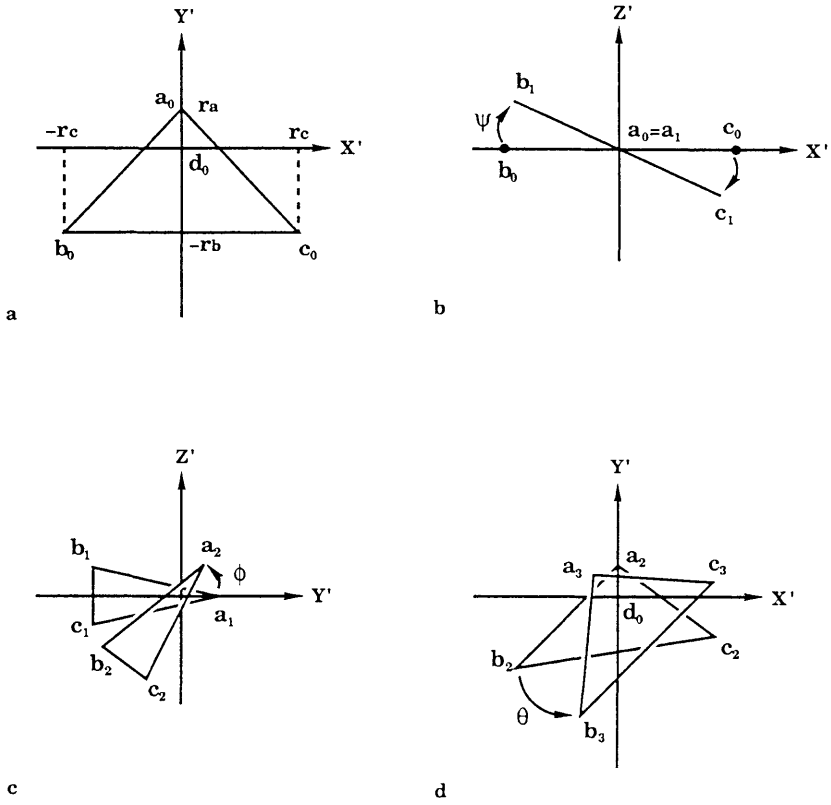

Figure 2. Pseudo-Euler angle definition for a canonical triangle. (a) $\Delta a_0 b_0 c_0$ indicates the initial location of a canonical triangle in the $X'Y'Z'$ system. $\Delta a_0 b_0 c_0$ lies on the $X'Y'$ plane with its center of mass, d_0 in the origin of $X'Y'Z'$ system, and a_0 in the positive part of the Y' axis. r_a, r_b, and $r_c > 0$. (b) The rotation of $\Delta a_0 b_0 c_0$ by ψ around the Y' axis gives $\Delta a_1 b_1 c_1$. (c) ϕ is a rotation angle about the X' axis. (d) θ gives a rotation of $\Delta a_2 b_2 c_2$ around the Z' axis, resulting in $\Delta a_3 b_3 c_3$.

TLE. In all the simulations a constant dielectric of 1.0 and a nonbonded cutoff of 8.0 Å were used.

First, a periodic box of 216 TIP3P water molecules was warmed up to 300 K for 5 ps at constant pressure (1 atm), followed by the equilibration for 5 ps at constant volume. Switching to the NVE simulation after warming was done to test energy conservation, as will be described below. In the above calculations the original RATTLE method was used with a value of 5×10^{-4} Å for both position and velocity tolerance. Additional equilibrations for 1 ps were performed with a variety of position and velocity tolerances from 5×10^{-4} to 1×10^{-7} Å as well as with SETTLE. An integration time step of 1 fs was used to obtain the set of equilibrated systems (E1). Table I lists the absolute accuracies of the constraints, i.e., bond length differences between the equilibrated structures (E1) and the canonical geometry, the components of relative velocities along the bond directions, and the angles between the net velocities and the normals of the corresponding bonds. As one can see in Table I, the SETTLE method gives very good accuracies, with less than 1.0×10^{-9} Å deviations for both position and velocity constraints, which correspond to those of a tolerance of less than 1×10^{-8} Å for the RATTLE approach. It seems that the errors in SETTLE are only dependent on machine accuracy. Al-

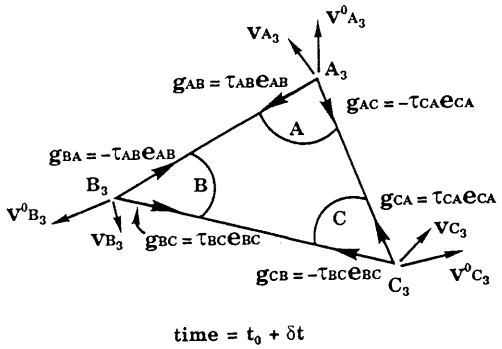

Figure 3. Schematic drawing for the velocity resetting. $\mathbf{v}^0_{A_3}$, $\mathbf{v}^0_{B_3}$, and $\mathbf{v}^0_{C_3}$ and $\mathbf{v}_{A_3}$, $\mathbf{v}_{B_3}$, and $\mathbf{v}_{C_3}$ are velocities of vertices before and after adding correction velocities at time $t_0 + \delta t$, respectively. In the case of velocity constraint, the constraint forces, $\mathbf{g}_{ij}$, are directed along the bonds defined by $\Delta A_3 B_3 C_3$ at time $t_0 + \delta t$. Let $\mathbf{e}_{AB}$, $\mathbf{e}_{BC}$, and $\mathbf{e}_{CA}$ be the unit vectors of $\overrightarrow{AB}$, $\overrightarrow{BC}$, and $\overrightarrow{CA}$. Then, constraint forces are written such that $\mathbf{g}_{AB} = \tau_{AB}\mathbf{e}_{AB}$ and $\mathbf{g}_{BA} = -\tau_{AB}\mathbf{e}_{AB}$.

Table I. Absolute accuracies obtained by SETTLE and RATTLE algorithms.

			RATTLE tolerance (Å)				
		SETTLE	1×10^{-7}	1×10^{-6}	1×10^{-5}	1×10^{-4}	5×10^{-4}
Position (Å)	Average[a]	$<1.0 \times 10^{-9}$	4.3×10^{-8}	4.8×10^{-7}	5.1×10^{-6}	5.0×10^{-5}	2.5×10^{-4}
	rms[b]	$<1.0 \times 10^{-9}$	5.6×10^{-8}	6.3×10^{-7}	6.4×10^{-6}	6.4×10^{-5}	3.2×10^{-4}
Velocity (Å)	Average[a]	$<1.0 \times 10^{-9}$	5.0×10^{-8}	5.0×10^{-7}	5.0×10^{-6}	5.1×10^{-5}	2.4×10^{-4}
(bond comp.)[c]	rms[b]	$<1.0 \times 10^{-9}$	6.4×10^{-8}	6.4×10^{-7}	6.5×10^{-6}	6.4×10^{-5}	3.2×10^{-4}
Velocity (°)	Average[a]	7.5×10^{-5}	2.5×10^{-4}	2.8×10^{-3}	2.6×10^{-2}	2.3×10^{-1}	1.2
(angle)[d]	rms[b]	7.5×10^{-5}	4.3×10^{-4}	4.9×10^{-3}	4.5×10^{-2}	3.8×10^{-1}	2.3

[a]Average = $[\Sigma(r_{ij} - d_{ij})]/n$.
[b]rms = $\{[\Sigma(r_{ij} - d_{ij})^2]/n\}^{1/2}$.
[c]The component of the relative velocity along the corresponding constraint bond multiplied by the time step of δt. It therefore has the dimension of Å.
[d]The angle between the relative velocity and the normal of the corresponding constraint bond.

though not shown in Table I, the conventional method results in the imbalance of the errors of bond distances. In the case of the tolerance of 5×10^{-4} Å, for example, the O—H bond length is longer than the equilibrium one by 3.8×10^{-4} Å on average while H—H distances are shorter than the target value by 1.5×10^{-5} Å in general. A similar imbalance is also observed for the velocity restraint.

Next, another equilibration was done using the same water box system. After warming to 300 K for 5 ps at 1 atm, the system was equilibrated for 5 ps under a constant volume condition using the new method with a step size of 1 fs. The SETTLE algorithm is naturally applicable to any type of simulation such as the NTP and NVE types. Employing the system thus obtained (E2) as the starting structure, two sets of MD simulations of the NVE type were carried out applying both the new method and the original approach with a variety of tolerances. One set consisted of 1 step MD runs while the other was constituted of 100 step runs. A time step of 1 fs was used in both cases. The comparison of the structure obtained by SETTLE to those of RATTLE gives another kind of accuracy, i.e., relative accuracy, which is shown in Table II. The tighter the tolerance is, the smaller the differences are. In other words, the configurations of the system based on both methods converge as the error tolerance becomes smaller. This indicates that the new algorithm works correctly and is equivalent to RATTLE. Interestingly, the magnitudes of velocity differences in 100 steps do not change at all compared to those of 1 step although the positions naturally deviate much more after 100 steps.

We then did NVE-type MD simulations of 200 steps with time increments of 1 and 2 fs based on the equilibrated structure (E2). The rms fluctuations of the total energy were derived from the last 100 steps; instabilities in the first several steps for the different tolerances might have biased the analysis. Since the residue-based switching function for the calculation of nonbonded interaction energies was used, the total energy should be conserved in the above simulations without coupling to the heat bath. The magnitude of the fluctuation, therefore, might be used as one indication of the accuracy of the trajectory, as it was used by van Gunsteren et al.[15] If the tolerance is smaller than a certain value, no significant change in the fluctuation of the total energy is observed, as shown in Figure 4, while the value itself is dependent on the time step. This feature had been reported by van Gunsteren et al. in their study on a small protein, bovine pancreatic trypsin inhibitor.[15] The new method naturally gives the smallest fluctuation values. Figure 4 suggests that the tolerance values of 1×10^{-5} and 1×10^{-4} Å are reasonable for use with the conventional approach for the step sizes of 1 and 2 fs, respectively. More conservative values such as 1×10^{-6} Å were used, however, in some studies.[7,15]

As the final analysis, the computation time spent for satisfying the constraints was examined. Using the initial structure of E1, 100 step MD runs were performed twice for each configuration with time steps of 1 and 2 fs, respectively. Table III reports the averaged timings. The computer time for RATTLE increases roughly proportional to the negative logarithm of the tolerance value. In the case of the VAX8650 scalar machine, SETTLE is at least three to seven times faster than RATTLE with the allowances of 5×10^{-4} and 1×10^{-4} Å, which are commonly used in conventional MD simulations. When the computation time is compared for the tolerances of 1×10^{-5} and 1×10^{-4} Å, which give reasonably small fluctuations of the total energy, the performance increase over the original version ranges from factors of seven to nine. As one can see from the data listed for the 1 fs time step, the speed-up ratios are almost the same for both position and velocity constraint routines. With respect to the FPS500 vector machine, the performance improvement for the time increment of 2 fs amounts to a factor of 98. If RATTLE is used with a

Table II. Relative accuracies (rms[a]) between SETTLE and RATTLE algorithms.

No. of steps		RATTLE tolerance (Å)				
		1×10^{-7}	1×10^{-6}	1×10^{-5}	1×10^{-4}	5×10^{-4}
1	Position (Å)	8.3×10^{-8}	9.4×10^{-7}	9.4×10^{-6}	9.5×10^{-5}	4.7×10^{-4}
	Velocity (Å) (bond comp.)[b]	9.4×10^{-8}	9.4×10^{-7}	9.4×10^{-6}	9.2×10^{-5}	4.6×10^{-4}
	Velocity (°) (angle)[c]	4.6×10^{-4}	4.4×10^{-3}	4.6×10^{-2}	4.3×10^{-1}	2.1
100	Position (Å)	3.2×10^{-7}	3.6×10^{-6}	3.6×10^{-5}	3.6×10^{-4}	1.8×10^{-3}
	Velocity (Å) (bond comp.)[b]	9.3×10^{-8}	9.5×10^{-7}	9.4×10^{-6}	9.5×10^{-5}	4.7×10^{-4}
	Velocity (°) (angle)[c]	4.1×10^{-4}	4.3×10^{-3}	4.2×10^{-2}	4.3×10^{-1}	2.2

[a]rms = $\{[\Sigma(r_{ij} - d_{ij})^2]/n\}^{1/2}$.
[b]The component of the relative velocity along the corresponding constraint bond multiplied by the time step of δt. It has therefore the dimension of Å.
[c]The angle between the relative velocity and the normal of the corresponding constraint bond.

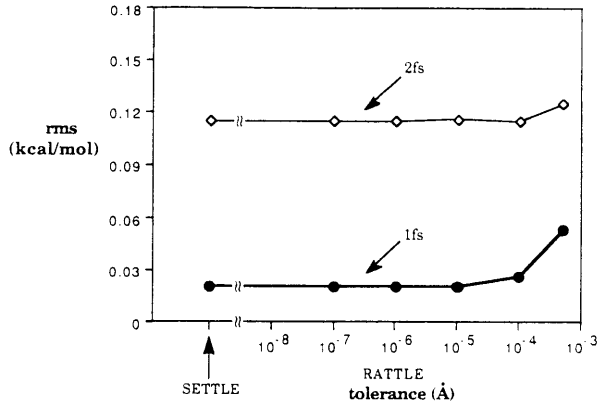

Figure 4. Relationship between the rms fluctuation of total energy and the tolerance. The rms fluctuation of the total energy obtained by the SETTLE method is shown in addition to those produced by the RATTLE approach with a variety of tolerances. Solid circles and open diamonds correspond to time steps of 1 and 2 fs, respectively.

reasonable tolerance of 1×10^{-4} Å, the speed-up ratio is 39. This is due to the noniterative nature of the new algorithm. Since the computation time spent on RATTLE amounts to nearly 20%, in the case of the conventional method with a typical tolerance, the reduction of the time obtained by SETTLE is important. In fact, when applied to practical systems we have studied, i.e., calixspherand (166 atoms) in a periodic box of 1110 TIP3P water molecules[16] and biotin (31 atoms) in a 502 TIP3P water box, the ratio spent on holonomic constraints during the MD simulations was reduced from 28.5 to 28.9% (tolerance of 1×10^{-5} Å) to 4.1 and 4.4%, respectively, on the scalar computer with a step size of 1 fs. Significant reductions of the computation time from 43.1 and 45.2 to 1.4 and 1.6% for the corresponding systems, respectively, were observed on the vector machine. In those simulations with SETTLE, no constraints were used for the solute molecules while the constraints on the TIP3P water solvent were applied automatically. This kind of strategy for the internal constraints seems to be suitable for coming out intensive simulations with the smaller time step of 1 fs.

As was shown in the previous section, the method presented here retains the simplicity of using Cartesian coordinates. It can be used in place of the standard SHAKE and RATTLE routines for rigid water molecules. In addition, the new algorithm is applicable to other three-point rigid

Table III. Comparison of computation time (s) spent for constraints during MD simulations of a periodic box of water.[a]

Time step	Machine type		SETTLE	RATTLE tolerance (Å)				
				1×10^{-7}	1×10^{-6}	1×10^{-5}	1×10^{-4}	5×10^{-4}
1 fs	Scalar[b]	Position	9.6 (2.9%)	153.1 (26.8%)				31.4 (8.4%)
		Velocity	5.2 (1.5%)	90.8 (15.9%)				14.8 (4.0%)
		Total[c]	14.8 (4.4%)	243.9 (42.7%)	187.6 (36.8%)	136.5 (29.7%)	78.4 (19.1%)	46.2 (12.4%)
		Ratio[d]	1.0	16.5	12.7	9.2	5.3	3.1
2 fs	Scalar[b]	Total[c]	14.9 (4.3%)	278.5 (45.8%)	222.3 (40.0%)	166.3 (33.6%)	110.9 (25.2%)	75.6 (18.6%)
		Ratio[d]	1.0	18.7	14.9	11.2	7.4	5.1
	Vector[e]	Total[c]	0.8 (1.8%)	78.6 (61.0%)	63.3 (57.6%)	46.9 (46.4%)	31.4 (40.7%)	21.1 (30.2%)
		Ratio[d]	1.0	98.3	79.1	58.6	39.3	26.4

[a]Values in parentheses show the percentage of computational efforts with respect to entire computation time.
[b]VAX8650 running under the VMS5.4 operating system.
[c]total = position + velocity.
[d]Ratio of total computation time spent for RATTLE to that of SETTLE.
[e]FPS500 running under UNIX operator system.

models by simple modifications of the equations. The MeOH model which employs the united atom approach for the methyl group is a good example. The SETTLE can be applied to a four-point water model like TIP4P[5] which has the fourth point with a certain charge and no mass if the force acting on the fourth point is distributed onto the other three points with masses in a reasonable manner.

Since SHAKE or RATTLE are widely employed in existing MD programs, SETTLE seems to be straightforward to implement in those programs, as we have done for the SPASMS package. Furthermore, this method is suitable for vector and parallel machines since it is not iterative and therefore the computation time spent is constant. Mertz et al. used a matrix inversion method for SHAKE in their study on vector and parallel algorithms for MD simulations.[7] Their approach, however, could not avoid the use of an iterative method, resulting in a decline of the speed-up ratio due to load imbalance while successful behaviors were attained for other parts of the MD calculation. Better results upon parallelization may be expected for water molecule constraints using SETTLE.

CONCLUSION

In this article we described an algorithm for satisfying constraints of the rigid water model and discussed its performance using the simulation package SPASMS. The main and subsidiary advantages of the method introduced here are:

1. SETTLE is quite accurate. The constraints are fulfilled exactly at each step of integration. This feature is ideal for the TIP3P and SPC water models, which have been parameterized using rigid geometries.
   1′. One need not worry about the choice of the tolerance value for rigid water, although reasonable numbers to be used with the conventional RATTLE are presented here.
2. SETTLE is fast. With respect to scalar machines, it is at least three to seven times faster than RATTLE. If used with reasonable tolerances mentioned above, the speed-up factor amounts to seven to nine. On vector machines, significant improvement of performance has been obtained, up to a factor of 98 over RATTLE.
   2′. SETTLE is also suitable for parallelization of the code because it is not iterative.
3. SETTLE can be easily implemented in standard MD packages. Since this algorithm is still based on Cartesian coordinates, it is straightforwardly incorporated into those packages in place of SHAKE or RATTLE on rigid water models.

Copies of FORTRAN subroutines SETTLE are available on request.

The authors acknowledge research support from the National Science Foundation (CHE-91 to P.A.K.).

APPENDIX A: DETAILS OF POSITION RESETTING OF SETTLE

Let us denote the coordinates of point A in $X'Y'Z'$ coordinate system by primes, i.e., $A' = (s, t, u)$ or $A = (X'_A, Y'_A, Z'_A)$. As is shown in Figure 2a, the coordinates of $\Delta a_0 b_0 c_0$ are given by

$$\begin{aligned} a'_0 &= (0, r_a, 0) \\ b'_0 &= (-r_c, -r_b, 0) \\ c'_0 &= (r_c, -r_b, 0) \end{aligned} \tag{A1}$$

where r_a, r_b, and $r_c > 0$.

$\Delta a_1 b_1 c_1$ is obtained by the rotation ψ ($-\pi/2 \leq \psi \leq \pi/2$) about Y' axis as illustrated in Figure 2b:

$$\begin{aligned} a'_1 &= a'_0 = (0, r_a, 0) \\ b'_1 &= (-r_c \cos\psi, -r_b, r_c \sin\psi) \\ c'_1 &= (r_c \cos\psi, -r_b, -r_c \sin\psi) \end{aligned} \tag{A2}$$

As indicated in Fig. 2c, ϕ ($-\pi < \phi \leq \pi$) gives a rotation of $\Delta a_1 b_1 c_1$ around X' into $\Delta a_2 b_2 c_2$

$$\begin{aligned} a'_2 &= (0, r_c \cos\phi, r_c \sin\phi) = (X'_{a_2}, Y'_{a_2}, Z'_{a_2}) \\ b'_2 &= (-r_c \cos\psi, -r_b \cos\phi - r_c \sin\psi \sin\phi, \\ &\quad -r_b \sin\phi + r_c \sin\psi \cos\phi) = (X'_{b_2}, Y'_{b_2}, Z'_{b_2}) \\ c'_2 &= (r_c \cos\psi, -r_b \cos\phi + r_c \sin\psi \sin\phi, \\ &\quad -r_b \sin\phi - r_c \sin\psi \cos\phi = (X'_{c_2}, Y'_{c_2}, Z'_{c_2}) \end{aligned} \tag{A3}$$

$\Delta a_3 b_3 c_3$ is produced by the rotation θ ($-\pi < \theta \leq \pi$) about Z' axis as shown in Figure 2d. Although the coordinates of $\Delta a_3 b_3 c_3$ might be expressed in the similar way as eq. (A3), the expression is rather complicated. In addition, θ could be determined separately from ψ and ϕ. Therefore we write the coordinates of $\Delta a_3 b_3 c_3$ based on those of $\Delta a_2 b_2 c_2$

$$\begin{aligned} a'_3 &= (X'_{a_2} \cos\theta - Y'_{a_2} \sin\theta, X'_{a_2} \sin\theta \\ &\quad + Y'_{a_2} \cos\theta, Z'_{a_2}) = (X'_{a_3}, Y'_{a_3}, Z'_{a_3}) \\ b'_3 &= (X'_{b_2} \cos\theta - Y'_{b_2} \sin\theta, X'_{b_2} \sin\theta \\ &\quad + Y'_{b_2} \cos\theta, Z'_{b_2}) = (X'_{b_3}, Y'_{b_3}, Z'_{b_3}) \\ c'_3 &= (X'_{c_2} \cos\theta - Y'_{c_2} \sin\theta, X'_{c_2} \sin\theta \\ &\quad + Y'_{c_2} \cos\theta, Z'_{c_2}) = (X'_{c_3}, Y'_{c_3}, Z'_{c_3}) \end{aligned} \tag{A4}$$

The Z' coordinates do not change after the rotation of θ and the Z' coordinates of $\Delta A_1 B_1 C_1$ are the same as those of $\Delta A_3 B_3 C_3$, as discussed in the explanation of the algorithm. Those relations as well as $\Delta A_3 B_3 C_3 \equiv \Delta a_3 b_3 c_3$ leads to

$$Z'_{A_1} = Z'_{A_3} = Z'_{a_3} = Z'_{a_2} = r_a \sin\phi \tag{A5}$$

$$Z'_{B_1} = Z'_{B_3} = Z'_{b_3} = Z'_{b_2} = -r_b \sin\phi + r_c \sin\psi \cos\phi \quad \text{(A6)}$$

$$Z'_{C_1} = Z'_{C_3} = Z'_{c_3} = Z'_{c_2} = -r_b \sin\phi - r_c \sin\psi \cos\phi \quad \text{(A7)}$$

Equation (A5) gives

$$\sin\phi = \frac{Z'_{A_1}}{r_a} \quad \text{(A8)}$$

where $|Z'_A| \leq r_a$. This is almost always true because $|Z'_A| \ll r_a$ in the practical MD simulations.* Adding eq. (A6) and eq. (A7), we also obtain the alternative expression

$$\sin\phi = \frac{Z'_{B_1} + Z'_{C_1}}{2r_b}$$

which is shown to be identical to eq. (A8) by a simple transformation. Subtracting eq. (A7) from eq. (A6), we find

$$2r_c \sin\psi \cos\phi = Z'_{B_1} - Z'_{C_1}$$

If $\cos\phi \neq 0$ (as is the case in general MD simulations)

$$\sin\psi = \frac{Z'_{B_1} - Z'_{C_1}}{2r_c \cos\phi} \quad \text{(A9)}$$

where $|Z'_{B_1} - Z'_{C_1}| < 2r_c|\cos\phi|$. By use of the basic relation of $\sin^2\omega + \cos^2\omega = 1$, we obtain

$$\cos\phi = \sqrt{1 - \sin^2\phi}, \qquad \cos\psi = \sqrt{1 - \sin^2\psi} \quad \text{(A10)}$$

where the positive sign of the square root is chosen for $\cos\phi$. Substituting eqs. (A8), (A9), and (A10) into eq. (A3), coordinates of $\Delta a_2 b_2 c_2$ are calculated.† Since the time consuming calculation of sine and cosine functions are not performed, this approach is fast as well as accurate.

In the next step θ can be calculated analytically by using the condition that constraint forces directed along the bond of $\Delta A_0 B_0 C_0$ are of equal magnitudes and opposite orientations as expressed in eq. (4) (Fig. 1b). Suppose $\Delta A_3 B_3 C_3$ is determined; then the displacement vectors are given by

$$\delta\mathbf{r}_A = \overrightarrow{A_1 A_3},\ \delta\mathbf{r}_B = \overrightarrow{B_1 B_3},\ \delta\mathbf{r}_C = \overrightarrow{C_1 C_3} \quad \text{(A11)}$$

According to eq. (4) the displacement vectors are given by

$$\begin{aligned}
\delta\mathbf{r}_A &= \frac{(\delta t)^2}{2m_a}(\mathbf{g}_{A_0B_0} + \mathbf{g}_{A_0C_0}) \\
&= \frac{(\delta t)^2}{2m_a}(\lambda_{AB}\overrightarrow{A_0B_0} + \lambda_{AC}\overrightarrow{A_0C_0}) \\
\delta\mathbf{r}_B &= \frac{(\delta t)^2}{2m_b}(\mathbf{g}_{B_0C_0} + \mathbf{g}_{B_0A_0}) \\
&= \frac{(\delta t)^2}{2m_b}(\lambda_{BC}\overrightarrow{B_0C_0} + \lambda_{BA}\overrightarrow{B_0A_0}) \\
\delta\mathbf{r}_C &= \frac{(\delta t)^2}{2m_c}(\mathbf{g}_{C_0A_0} + \mathbf{g}_{C_0B_0}) \\
&= \frac{(\delta t)^2}{2m_c}(\lambda_{CA}\overrightarrow{C_0A_0} + \lambda_{CB}\overrightarrow{C_0B_0}) \quad \text{(A12)}
\end{aligned}$$

Substituting eq. (A11) into (A12), we obtain the following expressions

$$\begin{aligned}
\frac{2m_a}{(\delta t)^2}\overrightarrow{A_1A_3} &= \lambda_{AB}\overrightarrow{A_0B_0} + \lambda_{AC}\overrightarrow{A_0C_0} \\
\frac{2m_b}{(\delta t)^2}\overrightarrow{B_1B_3} &= \lambda_{BC}\overrightarrow{B_0C_0} + \lambda_{BA}\overrightarrow{B_0A_0} \\
\frac{2m_c}{(\delta t)^2}\overrightarrow{C_1C_3} &= \lambda_{CA}\overrightarrow{C_0A_0} + \lambda_{CB}\overrightarrow{C_0B_0} \quad \text{(A13)}
\end{aligned}$$

Since $\overrightarrow{A_0B_0}$, $\overrightarrow{B_0C_0}$ and $\overrightarrow{C_0A_0}$ are not parallel to each other, λ_{ij} in eq. (A13) are uniquely defined. In the case of λ_{BC} and λ_{CB}, they can be written as

$$\lambda_{BC} = \frac{X'_{B_1B_3}Y'_{B_0A_0} - X'_{B_0A_0}Y'_{B_1B_3}}{X'_{B_0C_0}Y'_{B_0A_0} - X'_{B_0A_0}Y'_{B_0C_0}}$$

$$\lambda_{CB} = \frac{X'_{C_1C_3}Y'_{C_0A_0} - X'_{C_0A_0}Y'_{C_1C_3}}{X'_{B_0C_0}Y'_{C_0A_0} - X'_{C_0A_0}Y'_{B_0C_0}}$$

Since λ_{BC} should be equal to λ_{CB}, we obtain

$$\frac{(X'_{B_3} - X'_{B_1})(Y'_{A_0} - Y'_{B_0}) - (X'_{A_0} - X'_{B_0})(Y'_{B_3} - Y'_{B_1})}{(X'_{C_0} - X'_{B_0})(Y'_{A_0} - Y'_{B_0}) - (X'_{A_0} - X'_{B_0})(Y'_{C_0} - Y'_{B_0})} = \frac{(X'_{C_3} - X'_{C_1})(Y'_{A_0} - Y'_{C_0}) - (X'_{A_0} - X'_{C_0})(Y'_{C_3} - Y'_{C_1})}{(X'_{C_0} - X'_{B_0})(Y'_{A_0} - Y'_{C_0}) - (X'_{A_0} - X'_{C_0})(Y'_{C_0} - Y'_{B_0})}$$

After the rearrangement

$$(X'_{B_3} - X'_{B_1})(Y'_{B_0} - Y'_{A_0}) + (X'_{C_3} - X'_{C_1})(Y'_{C_0} - Y'_{A_0}) = (Y'_{B_3} - Y'_{B_1})(X'_{B_0} - X'_{A_0}) + (Y'_{C_3} - Y'_{C_1})(X'_{C_0} - X'_{A_0})$$

Substitutions of eq. (A4) and $X'_{c_2} = X'_{b_2}$ into the above relation gives

$$\begin{aligned}
&[X'_{b_2}(X'_{B_0} - X'_{C_0}) + (Y'_{B_0} - Y'_{A_0})Y'_{b_2} \\
&\quad + (Y'_{C_0} - Y'_{A_0})Y'_{c_2}]\sin\theta \\
&\quad + [X'_{b_2}(Y'_{C_0} - Y'_{B_0}) + (X'_{B_0} - X'_{A_0})Y'_{b_2} \\
&\quad + (X'_{C_0} - X'_{A_0})Y'_{c_2}]\cos\theta \\
&= (X'_{B_0} - X'_{A_0})Y'_{B_1} - X'_{B_1}(Y'_{B_0} - Y'_{A_0}) \\
&\quad + (X'_{C_0} - X'_{A_0})Y'_{C_1} - X'_{C_1}(Y'_{C_0} - Y'_{A_0}) \quad \text{(A14)}
\end{aligned}$$

The other relations of $\lambda_{AB} = \lambda_{BA}$ and $\lambda_{CA} = \lambda_{AC}$ provide the more complicated equations. Since these equations are equivalent to eq. (A14), we develop the discussion of only eq. (A14). Defining

*In the typical simulations with a time step of 1 to 2 fs at 300 K, the maximum values of ψ and ϕ are about 7°. They are small enough to guarantee $\cos\phi > 0$, $\cos\psi > 0$, $|Z'_A| \ll r_a$ and $|Z'_{B_1} - Z'_{C_1}| \ll 2r_c|\cos\phi|$. In the case of a longer step size of 5 fs and a higher temperature of 1500 K, maximum values are about 38°. Still, they are small enough to satisfy $\cos\phi > 0$, $\cos\psi > 0$, $|Z'_A| \ll r_a$ and $|Z'_{B_1} - Z'_{C_1}| \ll 2r_c|\cos\phi|$.

†Because of the numerical error in the calculation of $\cos\psi$, the coordinates of hydrogens (b_2 and c_2) might be adjusted to obtain the canonical geometry of the triangle.

$$\alpha = X'_{b_2}(X'_{B_0} - X'_{C_0}) + (Y'_{B_0} - Y'_{A_0})Y'_{b_2} + (Y'_{C_0} - Y'_{A_0})Y'_{c_2}$$

$$\beta = X'_{b_2}(Y'_{C_0} - Y'_{B_0}) + (X'_{B_0} - X'_{A_0})Y'_{b_2} + (X'_{C_0} - X'_{A_0})Y'_{c_2}$$

$$\gamma = (X'_{B_0} - X'_{A_0})Y'_{B_1} - X'_{B_1}(Y'_{B_0} - Y'_{A_0}) + (X'_{C_0} - X'_{A_0})Y'_{C_1} - X'_{C_1}(Y'_{C_0} - Y'_{A_0})$$

we can write eq. (A14) as

$$\alpha \sin\theta + \beta \cos\theta = \gamma \qquad \text{(A15)}$$

This is transformed into

$$\sqrt{\alpha^2 + \beta^2}\sin(\theta + \varepsilon) = \gamma, \qquad \tan\varepsilon = \frac{\beta}{\alpha}$$

The solution of the above trigonometric equation is given by*

$$\theta = \sin^{-1}\frac{\gamma}{\sqrt{\alpha^2 + \beta^2}} - \tan^{-1}\frac{\beta}{\alpha} \qquad \text{(A16)}$$

On the other hand, applying $\sin^2\theta + \cos^2\theta = 1$, eq. (A15) is expressed as

$$(\alpha^2 + \beta^2)\sin^2\theta - 2\alpha\gamma\sin\theta + \gamma^2 - \beta^2 = 0$$

The solution of this quadratic equation is given by the root's rule

$$\sin\theta = \frac{\alpha\gamma - \beta\sqrt{\alpha^2 + \beta^2 - \gamma^2}}{\alpha^2 + \beta^2} \qquad \text{(A17)}$$

where the sign of the square root is chosen so that eq. (A15) is satisfied. From either eq. (A16) or (A17), $\sin\theta$ and then $\cos\theta$ can be calculated while the latter might be preferable due to avoiding the calculations of trigonometric functions. Finally, the coordinates of $\Delta A_3B_3C_3$ are calculated by substitutions of $\sin\theta$ and $\cos\theta$ with the numbers obtained above into eq. (A4).

APPENDIX B: DETAILS OF VELOCITY RESETTING OF SETTLE

Equation (8) can be written as

$$\delta t(m_a + m_b)\cdot\tau_{AB} + \delta t\cdot m_a\cos B\cdot\tau_{BC} + \delta t\cdot m_b\cos A\cdot\tau_{CA} = 2m_am_bv^0_{AB}$$

$$\delta t\cdot m_c\cos B\cdot\tau_{AB} + \delta t(m_b + m_c)\cdot\tau_{BC} + \delta t\cdot m_b\cos C\cdot\tau_{CA} = 2m_bm_cv^0_{BC}$$

$$\delta t\cdot m_c\cos A\cdot\tau_{AB} + \delta t\cdot m_a\cos C\cdot\tau_{BC} + \delta t(m_c + m_a)\cdot\tau_{CA} = 2m_cm_av^0_{CA} \qquad \text{(B1)}$$

where $v^0_{AB} = \mathbf{e}_{AB}\cdot\mathbf{v}^0_{AB}$, $v^0_{BC} = \mathbf{e}_{BC}\cdot\mathbf{v}^0_{BC}$, and $v^0_{CA} = \mathbf{e}_{CA}\cdot\mathbf{v}^0_{CA}$ are the components of relative velocities along the bonds. The solution of the above simultaneous linear equations is given by

$$\tau_{AB} = m_a\{v^0_{AB}[2(m_a + m_b) - m_a\cos^2 C] + v^0_{BC}[m_b\cos C\cos A - (m_a + m_b)\cos B] + v^0_{CA}(m_a\cos B\cos C - 2m_b\cos A)\}/d$$

$$\tau_{BC} = \{v^0_{BC}[(m_a + m_b)^2 - m_b^2\cos^2 A] + v^0_{CA}m_a[m_b\cos A\cos B - (m_a + m_b)\cos C] + v^0_{AB}m_a[m_b\cos C\cos A - (m_a + m_b)\cos B]\}/d$$

$$\tau_{CA} = m_a\{v^0_{CA}[2(m_a + m_b) - m_a\cos^2 B] + v^0_{AB}(m_a\cos B\cos C - 2m_b\cos A) + v^0_{BC}[m_b\cos A\cos B - (m_a + m_b)\cos C]\}/d$$

$$d = \delta t[2(m_a + m_b)^2 + 2m_am_b\cos A\cos B\cos C - 2m_b^2\cos^2 A - m_a(m_a + m_b)\times(\cos^2 B + \cos^2 C)]/2m_b \qquad \text{(B2)}$$

based on the Cramer's rule

$$\tau_i = \frac{D_i}{D}$$

where D is the system determinant and D_i is the determinant obtained on replacing the respective coefficients of ith column [$\delta t(m_a + m_b)$, $\delta tm_c\cos B$ . . .] of D by $2m_am_bv^0_{AB}$, $2m_bm_cv^0_{BC}$ . . . . Substituting thus calculated τ_{AB}, τ_{BC}, and τ_{CA} into eq. (7), we get the constrained velocities, $\mathbf{v}_A$, $\mathbf{v}_B$, and $\mathbf{v}_C$.

*In the typical simulations with a time step of 1 to 2 fs at 300 K, the maximum value of θ is about 1°. In the case of a longer step size of 5 fs and a higher temperature of 1500 K, the maximum value is about 9°. These values are small enough to satisfy $-1/2\pi < \theta < 1/2\pi$ or $\sin\theta \ll 1$.

References

1. W.F. Van Gunsteren and H.J.C. Berendsen, *Angew. Chem. Int. Ed. Engl.*, **29**, 992 (1990); D.L. Beveridge and F.M. DiCapura, *Annu. Rev. Biophys. Biophys. Chem.*, **18**, 431 (1989).
2. W.L. Jorgensen and M. Briggs, *J. Am. Chem. Soc.*, **111**, 4190 (1989); T.P. Straatsma and H.J.C. Berendsen, *J. Chem. Phys.*, **89**, 5879 (1988).
3. C.L. Brooks and J.J. McDonald, *J. Am. Chem. Soc.*, **113**, 2295 (1991); S. Ha, J. Geo, B. Tidor, J.W. Brady, and M. Karplus, *J. Am. Chem. Soc.*, **113**, 1553 (1991).
4. D.M. Ferguson and P.A. Kollman, *Antisense Res. Dev.*, **1**, 243 (1991); D.A. Pearlman and P.A. Kollman, *Biopolymers*, **29**, 1193 (1990).
5. W.L. Jorgensen, J. Chandrasekhar, J.D. Madura, R.W. Impey, and M.L. Klein, *J. Chem. Phys.*, **79**, 926 (1983).
6. H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, and J. Hermans, *Intermolecular Forces*, B. Pullman, Ed., Reidel, Dordrecht, The Netherlands, 1981, pp 331-342.
7. J.E. Mertz, D.J. Tobias, C.L. Brooks III, and U.C. Singh, *J. Comp. Chem.*, **12**, 1270 (1991).
8. A. Rahman and F.H. Stillinger, *J. Chem. Phys.*, **55**, 3336 (1971).
9. D.J. Evans and S. Murad, *Mol. Phys.*, **34**, 327 (1977).
10. J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, *J. Comp. Phys.*, **23**, 327 (1977).
11. H. Andersen, *J. Comp. Phys.*, **52**, 24 (1983).
12. L. Verlet, *Phys. Rev.*, **159**, 98 (1967).

13. W.C. Swope, H.C. Andersen, P.H. Berens, and K.R. Wilson, *J. Chem. Phys.*, **76,** 637 (1982).
14. D.C. Spellmeyer, W.C. Swope, E.R. Evensen, P.A. Kollman, and R. Langridge, work in progress.
15. W.F. van Gunsteren and H.J.C. Berendsen, *Mol. Phys.*, **34,** 1311 (1977).
16. S. Miyamoto and P.A. Kollman, *J. Am. Chem. Soc.*, **114,** 3668 (1992).