# Algorithm for numerical integration of the rigid-body equations of motion

Igor P. Omelyan*
*Institute for Condensed Matter Physics, National Ukrainian Academy of Sciences,*
*1 Svientsitsky Street, UA-290011 Lviv, Ukraine*

(Received 22 December 1997)

An algorithm for numerical integration of the rigid-body equations of motion is proposed. The algorithm uses the leapfrog scheme and the quantities involved are angular velocities and orientational variables that can be expressed in terms of either principal axes or quaternions. Due to specific features of the algorithm, orthonormality and unit norms of the orientational variables are integrals of motion, despite an approximate character of the produced trajectories. It is shown that the method presented appears to be the most efficient among all such algorithms known. [S1063-651X(98)00107-X]

PACS number(s): 02.60.Cb, 04.25.−g, 95.75.Pq

The method of molecular dynamics (MD) plays a prominent role in studying molecular liquids. All existing techniques appropriate to simulate such systems can be categorized depending on what types of parameters are chosen to represent the rotational degrees of freedom and what kind of numerical algorithm is applied to integrate the corresponding equations of motion.

In the molecular approach, the phase trajectories are considered in view of translational and rotational motions. The translational dynamics is defined by the motion of molecular centers of masses, whereas the orientational motion can be determined in terms of Eulerian angles [1,2], quaternions [3–8], or principal-axis vectors [4]. The numerical integration within Eulerian angles is very inefficient due to singularities of the equations of motion [3,5]. If the quaternions or principal-axis vectors are involved, additional effort must be made to conserve their unit norms or orthonormality.

The atomic approach [9] treats the dynamics of the system in terms of translational motion of individual atoms that move under the potential-energy forces plus forces of constraints introduced to hold interatomic distances constant. This approach is believed to have good stability properties because the usual Verlet algorithm can be applied here. Nevertheless, the atomic approach is sophisticated to implement for point molecules and when there are more than two, three, or four atoms in the cases of linear, planar, and three-dimensional molecules, respectively. Moreover, to reproduce the rigid molecular structure it is necessary to solve complicated systems of nonlinear (in general, six per molecule) equations at each time step of the integration [10].

It is a common practice to integrate orientational motion with the Gear predictor-corrector algorithm of a high order [11]. Such an algorithm, being accurate at very small time steps, quickly becomes unstable with increasing step size [10]. Translational motion is usually integrated with lower-order Verlet [12], velocity Verlet [13], and leapfrog [14] integrators, because of their simplicity and exceptional numerical stability. However, the original versions of these integrators were constructed assuming that acceleration is velocity independent and therefore they cannot be applied directly to rotational dynamics. Analogous problems arise with translational motion in the presence of magnetic fields.

In order to remedy that situation, Fincham [15] has derived a rotational-motion version of the leapfrog algorithm in which systems of four nonlinear equations per molecule for quaternion components are solved by iteration. Ahlrichs and Brode have introduced a method [16] in which principal axes are considered as pseudoparticles and constraint forces are introduced to maintain their orthonormality. However, the algorithm is within the Verlet framework and does not contain angular velocities explicitly. The quaternion dynamics with constraints was also formulated [17]. As a result, a different algorithm within the velocity Verlet framework has been generated. Recently, the principal-axis scheme has been adapted to this framework as well [18]. Nevertheless, it was concluded that the best numerical stability can be achieved in the atomic-constraint approach.

In this paper we propose a leapfrog integrator of the rigid-body equations of motion. The main idea consists in involving angular velocities, instead of angular momenta, into the integration. This leads to significant simplifications with respect to angular-momentum versions [15]. The algorithm seems to be the most efficient and simple, exhibiting excellent stability properties that are similar to those observed within the cumbersome atomic-constraint technique.

Consider a classical system with $N$ rigid molecules composed of $M$ point atoms. Translational motion of the system is described in the usual way, applying Newton's law, whereas two first-order equations per molecule of the rotational dynamics can be obtained as follows. According to Euler equations [1], the rate of change in time of principal components $(\Omega_X^i,\Omega_Y^i,\Omega_Z^i)=\mathbf{\Omega}_i$ of the angular velocity is

$$J_\alpha \frac{d\Omega_\alpha^i}{dt}=K_\alpha^i(t)+(J_\beta-J_\gamma)\Omega_\beta^i(t)\Omega_\gamma^i(t). \qquad (1)$$

Here $(\alpha,\beta,\gamma)=(X,Y,Z)$, $(Y,Z,X)$, and $(Z,X,Y)$, $K_\alpha^i$ are principal components, $\mathbf{K}_i=\mathbf{A}_i\mathbf{k}_i^+$, of the torque $\mathbf{k}_i=\sum_{j;a,b}^{N;M}(\mathbf{r}_i^a-\mathbf{r}_i)\times\mathbf{f}_{ij}^{ab}$ exerted on molecule $i$ with respect to its center of mass $\mathbf{r}_i$ due to the site-site interactions $\mathbf{f}_{ij}^{ab}\equiv\mathbf{f}(\mathbf{r}_i^a-\mathbf{r}_j^b)$ with the other molecules, and $J_\alpha$ denote the principal moments of inertia. The orientational variables were

*Electronic address: nep@icmp.lviv.ua

collected in the square orthonormal matrices $\mathbf{A}_i$. The nine elements of each matrix $(i=1,\ldots,N)$ present coordinates of three principal axes $(X,Y,Z)$ of the molecule in the laboratory frame. The position of atom $a$ within molecule $i$ in the same frame is $\mathbf{r}_i^a(t)=\mathbf{r}_i(t)+\mathbf{A}_i^+(t)\mathbf{\Delta}^a$, where $\mathbf{\Delta}^a=(\Delta_1^a,\Delta_2^a,\Delta_3^a)^+$ is a vector column of these positions in the body frame and $\mathbf{A}^+$ the matrix transposed to $\mathbf{A}$.

The second equation follows from the definition of angular velocity

$$\frac{d\mathbf{A}_i}{dt}=\begin{pmatrix} 0 & \Omega_Z^i & -\Omega_Y^i \\ -\Omega_Z^i & 0 & \Omega_X^i \\ \Omega_Y^i & -\Omega_X^i & 0 \end{pmatrix}\mathbf{A}_i\equiv\mathbf{W}(\mathbf{\Omega}_i)\mathbf{A}_i, \qquad (2)$$

where the property $\mathbf{A}\mathbf{A}^+=\mathbf{I}$ of orthonormal matrices has been used, $\mathbf{W}(\mathbf{\Omega}_i)$ is a skew symmetric matrix associated with angular velocity, i.e., $\mathbf{W}^+(\mathbf{\Omega}_i)=-\mathbf{W}(\mathbf{\Omega}_i)$, and $\mathbf{I}$ designates the unit matrix. In an alternative representation the matrix $\mathbf{A}_i\equiv\mathbf{A}(\mathbf{q}_i)$ is a function of the four-component quaternion $\mathbf{q}_i\equiv(\xi_i,\eta_i,\zeta_i,\chi_i)^+$ [4,5]. The time derivatives of quaternions can be cast in the form

$$\frac{d\mathbf{q}_i}{dt}=\frac{1}{2}\begin{pmatrix} 0 & \Omega_Z^i & -\Omega_X^i & -\Omega_Y^i \\ -\Omega_Z^i & 0 & -\Omega_Y^i & \Omega_X^i \\ \Omega_X^i & \Omega_Y^i & 0 & \Omega_Z^i \\ \Omega_Y^i & -\Omega_X^i & -\Omega_Z^i & 0 \end{pmatrix}\mathbf{q}_i\equiv\mathbf{Q}(\mathbf{\Omega}_i)\mathbf{q}_i, \qquad (3)$$

where $\mathbf{Q}(\mathbf{\Omega}_i)$ is a skew symmetric matrix again and the unit quaternion norm $\xi_i^2+\eta_i^2+\zeta_i^2+\chi_i^2=1$, which follows from the orthonormality of $\mathbf{A}_i$, has been used.

In the case of translational motion, it is easy to derive the leapfrog algorithm [14]: $\mathbf{v}_i(t+h/2)=\mathbf{v}_i(t-h/2)+h\mathbf{a}_i(t)$, $\mathbf{r}_i(t+h)=\mathbf{r}_i(t)+h\mathbf{v}_i(t+h/2)$, where $h$ denotes the time increment, $\mathbf{v}_i=d\mathbf{r}_i/dt$ is the center-of-mass velocity, $\mathbf{a}_i(t)=(1/m)\Sigma_{j;a,b}^{N;M}\mathbf{f}_{ij}^{ab}(t)$ is the molecular acceleration, and $m$ is the mass of a separate molecule. Recently, it has been shown that, contrary to the conventional point of view, the order of truncation errors for this leapfrog is 4 rather than 3 for both coordinates and velocities due to a fortunate cancellation of uncertainties [19].

One problem with deriving a leapfrog algorithm for rotational motion is that angular accelerations (1) depend explicitly not only on spatial coordinates via molecular torques but also on angular velocities. Moreover, the time derivatives of orientational variables do not define angular velocities directly [see Eqs. (2) and (3)]. These difficulties cannot be handled with a simple leapfrog algorithm in which the position and velocity are known at different times. It is worth emphasizing that similar problems (even much more difficult) arise in the angular-momentum approach [15] and the Verlet and velocity Verlet frameworks [17,18].

The basic idea of our approach lies in involving principal angular velocities in the integration process. Then, acting in the spirit of the leapfrog scheme and using the Euler equation (1), one obtains

$$\Omega_\alpha^{i\,(n+1)}\left(t+\frac{h}{2}\right)=\Omega_\alpha^i\left(t-\frac{h}{2}\right)+\frac{h}{J_\alpha}[K_\alpha^i(t)+(J_\beta-J_\gamma)\times\Omega_\beta^{i\,(n)}(t)\Omega_\gamma^{i\,(n)}(t)]. \qquad (4)$$

While the molecular torques $K_\alpha^i(t)$ can easily be evaluated via the coordinates $\mathbf{r}_i(t)$ and $\mathbf{A}_i(t)$ or $\mathbf{q}_i(t)$, a propagation of the products of angular velocities in Eq. (4) to on-step levels of time is necessary. The obvious choice for this is

$$\Omega_\beta^{i\,(n)}(t)\Omega_\gamma^{i\,(n)}(t)=\frac{1}{2}\left[\Omega_\beta^i\left(t-\frac{h}{2}\right)\Omega_\gamma^i\left(t-\frac{h}{2}\right)+\Omega_\beta^{i\,(n)}\left(t+\frac{h}{2}\right)\Omega_\gamma^{i\,(n)}\left(t+\frac{h}{2}\right)\right]. \qquad (5)$$

In view of Eq. (5), Eq. (4) constitutes a system of a maximum of three nonlinear equations per molecule for the unknowns $\Omega_\alpha^i(t+h/2)$. The system is simple and can be solved in a quite efficient way by iteration, $n=0,1,\ldots,$ taking $\Omega_\alpha^{i\,(0)}(t+h/2)=\Omega_\alpha^i(t-h/2)$ as an initial guess. We note that the order of truncation errors for the angular-velocity evaluation (4) reduces to 3, because the approximation (5) is only second order accurate for $h$.

The evaluation of orientational variables can be realized by writing

$$\mathbf{S}_i(t+h)=\mathbf{S}_i(t)+h\mathbf{H}_i\mathbf{S}_i\left(t+\frac{h}{2}\right) \qquad (6)$$

for principal-axis vectors $(\mathbf{S}_i\equiv\mathbf{A}_i,\mathbf{H}_i\equiv\mathbf{W}_i)$ and quaternion $(\mathbf{S}_i\equiv\mathbf{q}_i,\mathbf{H}_i\equiv\mathbf{Q}_i)$ representations, where Eqs. (2) and (3) have been used. The matrices $\mathbf{W}_i\equiv\mathbf{W}(\mathbf{\Omega}_i)$ and $\mathbf{Q}_i\equiv\mathbf{Q}(\mathbf{\Omega}_i)$ are calculated using already defined angular velocities $\mathbf{\Omega}_i(t+h/2)$, whereas orientational variables can be propagated to midstep levels of time as

$$\mathbf{S}_i\left(t+\frac{h}{2}\right)=\frac{1}{2}[\mathbf{S}_i(t)+\mathbf{S}_i(t+h)]. \qquad (7)$$

Equation (6) together with Eq. (7) is in fact a system of linear equations with respect to elements of $\mathbf{A}_i(t+h)$ and $\mathbf{q}_i(t+h)$, which therefore can be solved analytically. The result is

$$\mathbf{S}_i(t+h)=\left(\mathbf{I}-\frac{h}{2}\mathbf{H}_i\right)^{-1}\left(\mathbf{I}+\frac{h}{2}\mathbf{H}_i\right)\mathbf{S}_i(t)\equiv\mathbf{\Theta}_i(t,h)\mathbf{S}_i(t). \qquad (8)$$

More explicit expressions for the set $\mathbf{\Theta}_i\equiv\{\mathbf{D}_i,\mathbf{G}_i\}$ of evolution matrices are $\mathbf{D}_i=\{\mathbf{I}\,[1-(h^2/4)\Omega_i^2]+h\mathbf{W}_i+(h^2/2)\mathbf{P}_i\}/[1+(h^2/4)\Omega_i^2]$ and $\mathbf{G}_i=\{\mathbf{I}\,[1-(h^2/16)\Omega_i^2]+h\mathbf{Q}_i\}/[1+(h^2/16)\Omega_i^2]$ in the cases of principal axes and quaternion representations, respectively, where $\mathbf{P}_i$ is a symmetric matrix with the elements $\Omega_\alpha^i(t+h/2)\Omega_\beta^i(t+h/2)$ and $\Omega_i^2\equiv\mathbf{\Omega}_i^2(t+h/2)$. This completes the algorithm. It is interesting to remark that the evaluation (8) exhibits the same fourth-order local accuracy for $h$ as in the case of translational coordinates, despite the second order of the interpola-

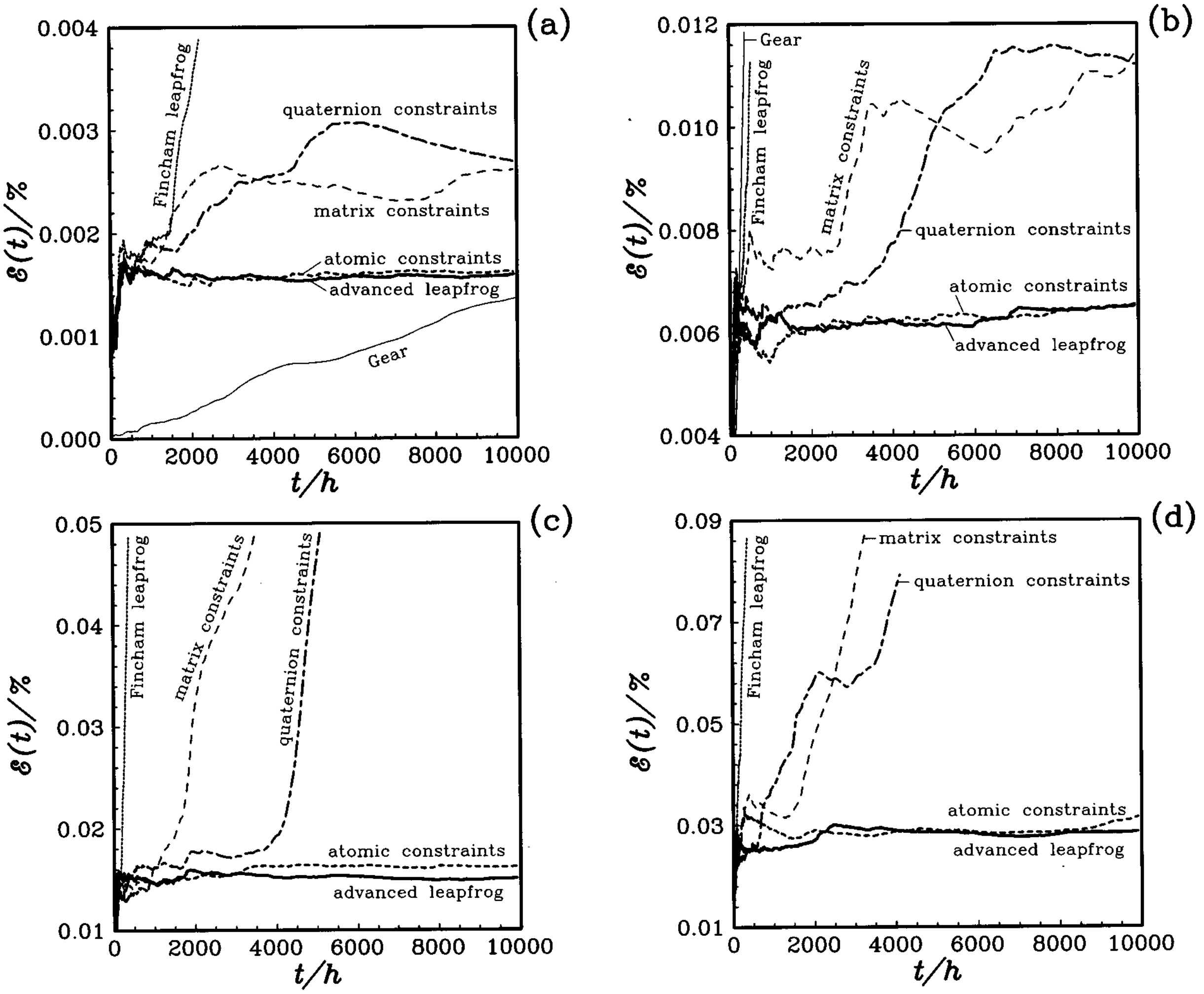

FIG. 1. Total energy fluctuations as functions of the length of the simulations on liquid water, performed by various techniques at four fixed time steps: (a) 1 fs, (b) 2 fs, (c) 3 fs, and (d) 4 fs.

tion (7). The reason for this results again from a cancellation of errors arising in coordinates and velocities during two neighboring time steps.

It can be verified easily that the matrix $(\mathbf{I}-\lambda\mathbf{H})^{-1}(\mathbf{I}+\lambda\mathbf{H})$ is orthonormal at arbitrary values of $\lambda$, provided $\mathbf{H}^{+}=-\mathbf{H}$. Then, as follows from the construction (8), the evolution matrices $\mathbf{D}_i$ and $\mathbf{G}_i$ are orthonormal as well. Therefore, if initially the orthonormality of $\mathbf{A}_i$ and unit norms of $\mathbf{q}_i$ are satisfied, they will be fulfilled perfectly at arbitrary times in the future, despite the approximate character of the trajectories produced. This can be considered as the main advantage of the algorithm derived that distinguishes it from all other singularity free algorithms because no additional effort is needed to preserve the rigid structure of molecules.

We now test our approach on the basis of MD simulations on liquid water. The simulations were performed in an *NVE* ensemble with $N=256$ molecules at a density of m$N/V=1$ g/cm$^3$ and at a temperature of 298 K using the potential of Jorgensen *et al.* [20] and reaction field geometry [21]. All runs were started from an identical well equilibrated configuration. The numerical stability was identified in terms of fluctuations of the total energy $\mathcal{E}=[\langle(E-\langle E\rangle)^2\rangle]^{1/2}/|\langle E\rangle|$. The kinetic part of the energy was calculated at time $t$ by setting $\mathbf{V}(t)=\frac{1}{2}[\mathbf{V}(t-h/2)+\mathbf{V}(t+h/2)]+O(h^2)$ for $\mathbf{V}\equiv\{\mathbf{v}_i,\mathbf{\Omega}_i\}$, where the main term $O(h^2)$ of uncertainties is self-consistent with the second order of global errors for our algorithm (one order lower than the minimal order of truncation errors for coordinates and velocities).

As the atomic-constraint algorithm [9,10] is intensively exploited and its performances are generally recognized, we have made comparative tests using this method and our advanced leapfrog algorithm within quaternion and principal-axes variables, as well as all known other approaches, namely, the fifth-order Gear algorithm [11], the implicit leapfrog algorithm of Fincham [15], the pseudoparticle formalism [16], and quaternion- and matrix-constraint methods [17,18]. Samples of $\mathcal{E}(t)$ as a function of the length of the simulations at four fixed values of $h=1$, 2, 3, and 4 fs are shown in Fig. 1. The usual value of the step size for studying such a system is 2 fs [22].

Despite the fact that the Gear algorithm integrates the equations of motion very well at $h=1$ fs, it has a very small region of stability and cannot be used for greater time steps [see Fig. 1(b)]. Small step sizes are impractical in calculations because too much expensive computer time is required to cover sufficient phase space. At the same time, the quaternion- and matrix-constraint methods as well as the pseudoparticle approach produce much more stable trajectories and exhibit a similar equivalence in the energy conservation. Worse results are observed for Fincham's leapfrog method. Finally, the best numerical stability is achieved in the atomic-constraint technique and our leapfrog scheme within both quaternion and principal axis representations, which conserve the energy approximately with the same accuracy (the results for principal-axis variables and the pseudoparticle formalism are not included in the figure to simplify the graph). Very few iterations (the mean number of iterations varied from 3 to 5 at $h=1-4$ fs) were sufficient to find solutions to the system of nonlinear equations (4) with a precision of $10^{-12}$. This contributes additional, negligibly small computation time to the total time.

No shift of the total energy was observed for the atomic-constraint and our leapfrog techniques at $h\leqslant 4$ fs over a length of 10 000 steps. To reproduce features of an $NVE$ ensemble quantitatively, it is necessary for the ratio $\Gamma=\mathcal{E}/\Upsilon$ of the total energy fluctuations to the fluctuations $\Upsilon$ of the potential energy to be no more than a few percent. We have obtained the following levels of $\mathcal{E}$ at the end of the runs in our leapfrog approach: 0.0016%, 0.0065%, 0.015%, and 0.029%, corresponding to $\Gamma\approx 0.29\%$, 1.2%, 2.7%, and 5.2% at $h=1$, 2, 3, and 4 fs, respectively (for the system under consideration $\Upsilon\approx 0.56\%$). Therefore, the greatest time step considered (4 fs) is still suitable for precise calculations. The ratio $\Gamma$ can be fitted with great accuracy to the function $Ch^2$ with a coefficient of $C\approx 0.29\%\;\;\mathrm{fs}^{-2}$. This is completely in line with our theoretical prediction about a characteristic square growth of global errors and, as a consequence, $\mathcal{E}(t)$ at $t\gg h$. The square growth was observed in all other approaches, except for the Gear algorithm. However, only the advanced leapfrog algorithm provides a minimum of $C$ and total energy fluctuations.

The algorithm presented might become popular because of its great stability, simplicity to implement for arbitrary rigid bodies, and its intrinsic conservation of rigid structures. These features should be considered as significant benefits of the algorithm with respect to all the other approaches. It can easily be substituted into existing MD programs on rigid polyatomic molecules. Moreover, since velocities appear explicitly, the algorithm can be extended to a thermostat version and to integration in the presence of magnetic fields. We plan to discuss these problems elsewhere.

The author thanks the president of Ukraine for financial support.

[1] A. Rahman and F.H. Stillinger, J. Chem. Phys. **55**, 3336 (1971); **60**, 1545 (1974).

[2] J.P. Ryckaert and A. Bellemans, Chem. Phys. Lett. **30**, 123 (1975).

[3] H. Goldstein, *Classical Mechanics*, 2nd ed. (Addison-Wesley, Reading, MA, 1967).

[4] D.J. Evans, Mol. Phys. **34**, 317 (1977).

[5] D.J. Evans and S. Murad, Mol. Phys. **34**, 327 (1977).

[6] D.J. Tildesley and P.A. Maden, Mol. Phys. **42**, 1137 (1981).

[7] M.P. Allen and D.J. Tildesley, *Computer Simulation of Liquids* (Clarendon, Oxford, 1987).

[8] D.C. Rapaport, *The Art of Molecular Dynamics Simulation* (Cambridge University Press, Cambridge, 1995).

[9] J.P. Ryckaert, G. Ciccotti, and H.J.C. Berendsen, J. Comput. Phys. **23**, 327 (1977).

[10] G. Ciccotti, J.P. Ryckaert, and M. Ferrario, Mol. Phys. **47**, 1253 (1982).

[11] C.W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Engelwood Cliffs, NJ, 1971).

[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] R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles* (McGraw-Hill, New York, 1981).

[15] D. Fincham, Mol. Simul. **8**, 165 (1992).

[16] R. Ahlrichs and S. Brode, Comput. Phys. Commun. **42**, 59 (1986).

[17] I.P. Omelyan, Comput. Phys. **12**, 97 (1998).

[18] I.P. Omelyan, Comput. Phys. Commun. **109**, 171 (1998).

[19] A.K. Mazur (unpublished).

[20] W. L. Jorgensen, J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein, J. Chem. Phys. **79**, 926 (1983).

[21] I.P. Omelyan, Phys. Lett. A **223**, 295 (1996).

[22] D. Bertolini and A. Tani, Phys. Rev. E **51**, 1091 (1995).