JOURNAL OF COMPUTATIONAL PHYSICS **136,** 354–365 (1997)
ARTICLE NO. CP975740

Common Molecular Dynamics Algorithms Revisited: Accuracy and Optimal Time Steps of Störmer–Leapfrog Integrators

Alexey K. Mazur

Laboratoire de Biochimie Théorique, CNRS UPR9080 Institue de Biologie Physico-Chimique 13, rue Pierre et Marie Curie, Paris, 75005, France

Received November 12, 1996; revised April 21, 1997

The Störmer–Verlet–leapfrog group of integrators commonly used in molecular dynamics simulations has long become a textbook subject and seems to have been studied exhaustively. There are, however, a few striking effects in performance of algorithms which are well known but have not received adequate attention in the literature. A closer view of these unclear observations results in unexpected conclusions. It is shown here that contrary to the conventional point of view, the leapfrog scheme is distinguished in this group both in terms of the order of truncation errors and the conservation of the total energy. In this case the characteristic square growth of fluctuations of the total energy with the step size, commonly measured in numerical tests, results from additional interpolation errors with no relation to the accuracy of the computed trajectory. An alternative procedure is described for checking energy conservation of leapfrog-like algorithms which is free from interpolation errors. Preliminary tests on a representative model system suggest that standard step size values used at present are lower than necessary for accurate sampling.

I. INTRODUCTION

The well-known group of integrators for the Newton's equations comprising Verlet [1], leapfrog [2], velocity Verlet [3], and Beeman [4] methods play a central role in the classical methodology of molecular dynamics. Due to their simplicity and exceptional stability they proved to be the best choice for long time, large step size calculations and they are now employed in the great majority of simulations of large systems. All or some of these methods are always described in detail and compared in any modern textbook [2, 5, 6]. It is well known that they are based upon a ninety-year-old Störmer time-centered difference approximation of accelerations [7, 8], that given appropriate initial conditions they produce the same trajectory in coordinate space and that, therefore, they all represent variations of a single algorithm.

Despite this long prehistory, the performance of these algorithms still attracts remarkable attention, first, because of their practical importance and, second, because understanding of the origin of their exceptional properties may help to develop even better algorithms. In particular, their stability has been attributed to the correspondence between the order of the finite difference equation and its analytical analog [2], to their time reversibility [2, 9, 10], to oscillating fundamental solutions [11] and to their symplectic property [9, 12, 13]. A hallmark of these method is a square power growth of global errors with the step size, which is invariably observed with different testing techniques and can even be used for debugging computer codes [5]. Because of the low apparent order of approximation they usually appear less accurate than other methods in comparative studies, but, with large step sizes, when other algorithms loose stability, the leapfrog scheme and its analogs still produce trajectories with very low total energy drift and correct average static and dynamic properties [14–16]. These observations are known and exploited for such a long time that it seems to have been forgotten that actually they are quite unusual and that this behavior still has no clear explanation.

The observed persistence of averages and a low drift of the total energy might mean that the numerical trajectory computed with a large step size, although inaccurate in a strict sense, holds well to the correct constant energy hypersurface in phase space. It is well known, however, that compared with other integrators the Störmer-equivalent algorithms conserve the instantaneous total energy rather poorly, its value already strongly fluctuates at time steps well below the theoretical limit of stability. Thus, it appears that the trajectory constantly deviates from the initial hypersurface, but always finds a way back, which is rather striking because, for example, the velocity Verlet algorithm is self-starting and, consequently, keeps no memory of the preceding part of the trajectory. One explanation to these observations follows from the general property of symplectic integrators which, with a sufficiently small time step, generate discretizations of exact trajectories corresponding to perturbed Hamiltonians [12]. This property, however, holds for small time steps only and it does not explain why the leapfrog-equivalent algorithms are distinguished among symplectic integrators as well.

The above contradiction may be settled if one assumes that the total energy is actually conserved better than it appears. For a Störmer or a leapfrog trajectory this is

0021-9991/97 $25.00

possible, in principle, because in these cases the total energy is not a well-defined quantity. The original Störmer formula [1, 5] needs only coordinates and accelerations for computing a trajectory and so, strictly speaking, velocities are not defined. There are many methods to evaluate velocities and kinetic energies but they all employ some finite difference approximations and, usually, interpolations, which is often done implicitly, as in the case of the velocity Verlet and Beeman algorithms [3, 4]. It is understood that these computations may introduce additional errors into the computed total energy, but these errors are difficult to evaluate or to get rid of and one cannot generally tell how large is their relative contribution.

The initial goal of the present study was to check more accurately how well a leapfrog trajectory holds to a constant-energy hypersurface, when only approximations intrinsic to the algorithm play a role and there is no influence of additional approximations and interpolations. As happens sometimes, the solution of a practical task has led to a comprehensive " *ab initio*" analysis of several related problems which have been unrecognized or simply ignored in the literature. It is shown here that contrary to the conventional point of view, the leapfrog scheme is more accurate than other algorithms of this group, both in terms of the order of the truncation errors and the conservation of the total energy. The observed square growth of fluctuations of the total energy with the step size appears to result from interpolations only and has no relation to the accuracy of the computed trajectory. An alternative procedure is proposed for checking energy conservation of leapfrog-like algorithms which is free from interpolation errors. Testing on an example of protein dynamics confirms that the leapfrog trajectory really manages to sample from a correct hypersurface in phase space with larger step sizes than commonly recommended.

When a classical text-book subject is discussed it is difficult to maintain the logic of argumentation without reproducing some well-known general results. The author hopes, therefore, to be excused by experts if some of the derivations in the text appear trivial.

II. RESULTS AND DISCUSSION

We will consider Newton's equation for a particle of a unit mass

$$\ddot{x} = f(x), \tag{1}$$

where the dot notation denotes time derivatives. The Störmer fourth order algorithm (the order of the algorithm here and below refers to the order of the truncation error) first introduced in this field by Verlet [1] is

$$x_{n+1} = 2x_n - x_{n-1} + a_n h^2, \tag{2}$$

where x_n and a_n are the coordinate and the acceleration at the nth time step and h is the step size. The leapfrog scheme popularized by Hockney and Eastwood [2] is

$$v_{n+1/2} = v_{n-1/2} + a_n h \tag{3a}$$

$$x_{n+1} = x_n + v_{n+1/2} h, \tag{3b}$$

where $v_{n+1/2}$ is the half-step velocity. The Swope *et al.* [3] formulation of the algorithm referred to as velocity Verlet is

$$x_{n+1} = x_n + \left(v_n + a_n \frac{h}{2}\right) h \tag{4a}$$

$$v_{n+1} = v_n + (a_n + a_{n+1}) \frac{h}{2}. \tag{4b}$$

Finally, the Beeman method [4],

$$x_{n+1} = x_n + v_n h + \left(\frac{2}{3} a_n - \frac{1}{6} a_{n-1}\right) h^2 \tag{5a}$$

$$v_{n+1} = v_n + \left(\frac{1}{3} a_{n+1} + \frac{5}{6} a_n - \frac{1}{6} a_{n-1}\right) h. \tag{5b}$$

Instructive discussions of the relationships between these methods can be found elsewhere [2, 5, 10, 17, 18]. The algorithms defined by Eqs. (2)–(5) significantly differ in the implicitly adopted point of view upon the definition of the on-step velocities and, consequently, the total energy. For the velocity Verlet and Beeman integrators the on-step velocity v_n, kinetic energy K_n and the total energy E_n are well defined quantities. In contrast, for both the leapfrog and the original Störmer algorithms no complete trajectory in phase space is computed and one can choose between different interpolation formulae for the calculation of on-step velocities or kinetic energies. Because of the reasons given below it is most convenient for us to take the second choice. Let us consider that our trajectory is computed by the leapfrog integrator (3a), (3b). The coordinates thus obtained automatically satisfy Eq. (2). The corresponding solution of Eqs. (4a), (4b) is obtained if we employ the interpolation formula

$$v_n = \frac{1}{2}(v_{n-1/2} + v_{n+1/2}) \tag{6}$$

and the solution of Eqs. (5a), (5b) with

$$v_n = \frac{1}{6}(2v_{n+1/2} + 5v_{n-1/2} - v_{n-3/2}). \tag{7}$$

It should be noted that since the Beeman algorithm employs coordinates and velocities from several time steps it is equivalent to the leapfrog algorithm with interpolation (7) only if its initial conditions match Eqs. (3a), (3b).

When analytical trajectories are described it is common to use the term "state" to refer to a point in phase space. From any state the analytical trajectory can be continued in a unique way in both time directions. For the following discussion, it will be convenient to use a similar terminology referring to numerical trajectories. Thus, a leapfrog state is defined as a pair $(x_n, v_{n-1/2})$, a velocity Verlet state as (x_n, v_n), and a Störmer state as (x_n, x_{n-1}). In each of these three cases there exists a unique analytical solution of Eq. (1) to which these states belong although the corresponding exact trajectories are not identical.

A. *The Order of Approximation of the Leapfrog Scheme*

This very basic and apparently simple question appears to be very confused and persistently misinterpreted in the literature. Fundamental textbooks always demonstrate that algorithm (2) is accurate to fourth order but add that velocities are usually computed with a lower accuracy. Equivalence between all integrators of this group also appears only when low-order approximations are used for velocities [18]. At the same time, Eqs. (3a), (3b) and the velocity Verlet and Beeman algorithms are only $O(h^3)$ accurate. In various numerical tests all these integrators exhibit an $O(h^2)$ growth of global errors in agreement with $O(h^3)$ order of the local truncation error [5, 6]. This occurs even when no lower order approximations are involved, for instance, if Eq. (2) is used and the deviation of coordinates from an analytical solution is checked [7]. Although these observations are not contradictory they are rather perplexing, but, however surprising, they are essentially ignored in the literature. The confusion clearly results from approximations of velocities which are not included in Eq. (2), but since all these algorithms can produce the same trajectory the difficulty is only formal and seems to present no practical interest. We will now see that this reasoning involves a conceptual error with important consequences.

Consider the standard derivation of Eq. (2), which starts from two Taylor series from time t in opposite directions

$$x(t+h) = x(t) + \dot{x}(t)h + \frac{1}{2}\ddot{x}(t)h^2 + \frac{1}{6}\dddot{x}(t)h^3 + O(h^4) \quad (8a)$$

$$x(t-h) = x(t) - \dot{x}(t)h + \frac{1}{2}\ddot{x}(t)h^2 - \frac{1}{6}\dddot{x}(t)h^3 + O(h^4). \quad (8b)$$

Adding these two expansions gives Eq. (2), with all odd-order terms eliminated, and a resultant truncation error of $O(h^4)$. Now let us perform a similar derivation for half-step velocities, which gives

$$v\left(t+\frac{h}{2}\right) = 2v\left(t-\frac{h}{2}\right) - v\left(t-\frac{3h}{2}\right) + \ddot{v}\left(t-\frac{h}{2}\right)h^2 + O(h^4) \quad (9)$$

and by substituting

$$\ddot{v}\left(t-\frac{h}{2}\right) = \frac{1}{h}[\ddot{x}(t) - \ddot{x}(t-h)] + O(h^2) \quad (10)$$

we get a fourth-order finite difference equation

$$v_{n+1/2} = 2v_{n-1/2} - v_{n-3/2} + (a_{n-1} - a_n)h^2. \quad (11)$$

It is readily seen that any trajectory produced by the leapfrog scheme (3a), (3b) satisfies this equation, so it is an exact analog of the Störmer formula (2) for velocities. We see, therefore, that the order of approximation of the leapfrog scheme is four rather than three, both for coordinates and velocities. Equations (2) and (11) are common fourth-order predictors which use values from several time steps to reduce the truncation error. The fact that the same trajectory can be generated by lower order Eqs. (3a), (3b) results from a fortunate cancelation of errors.

Note, however, that Eqs. (2), (11) and Eqs. (3a), (3b) are not equivalent. Equation (11) provides a leapfrog solution only if initial conditions match Eq. (3a). Note also, that Eqs. (2) and (11) are coupled only via accelerations—no relationship between coordinates and velocities is implied *a priori.* Therefore, starting from arbitrary initial conditions a completely nonleapfrog trajectory may be produced. The relationship between the general solutions of Eqs. (2), (11), and (3a), (3b) presents an interesting question, but it is beyond the scope of the present paper.

B. *Accumulation of Errors Along a Leapfrog Trajectory*

The truncation error is in fact only a part of the story because single-step errors can accumulate thus producing so-called global errors [5, 6], which are relevant for assessment of an algorithm and which, as noted above, exhibit a characteristic $O(h^2)$ growth. Let us check how, in the case of a leapfrog trajectory, truncation errors accumulate for conventional testing procedures.

Suppose we have a trajectory of duration T; we compute some parameter $Q[x(t_i), v(t_i)]$ at M time steps $i = 1, ..., M$ and evaluate

$$\overline{\Delta Q^2} = \frac{1}{M}\sum_{i=1}^{M}\Delta Q_i^2 = \frac{1}{M}\sum_{i=1}^{M}[Q(t_i) - Q_0^a(t_i)]^2. \quad (12)$$

Here $Q_0^a(t_i)$ is the corresponding analytical value computed for an exact trajectory starting from the initial state. An overbar here and below denotes time averaging. Now we have

$$\Delta Q_i = \sum_{j=0}^{i} \Delta\Delta Q_j, \tag{13}$$

where

$$\begin{aligned}\Delta\Delta Q_j &= \Delta Q_{j+1} - \Delta Q_j = Q(t_{j+1}) - Q(t_j) - Q_0^a(t_{j+1}) + Q_0^a(t_j)\\ &= [Q(t_{j+1}) - Q_j^a(t_{j+1})] + [Q_j^a(t_{j+1}) - Q(t_j)]\\ &\quad - [Q_0^a(t_{j+1}) - Q_0^a(t_j)],\end{aligned} \tag{14}$$

where $Q_j^a(t)$ refers to the analytical trajectory starting from jth state in the numerical trajectory. Here only the first bracket in the r.h.s. is contributed by the local truncation error while the rest results from deviations of analytical solutions corresponding to different initial conditions. Discarding higher order terms one can rewrite Eq. (14) as

$$\Delta\Delta Q_j \approx \alpha_j h^m + [\dot{Q}_j^a(t_j) - \dot{Q}_0^a(t_j)]h, \tag{15}$$

where α_j are constant coefficients and, for the leapfrog scheme, $m = 4$.

In the general case both terms in the r.h.s. of Eq. (15) must be taken into consideration. Let us check the case when Q is just the coordinate and so Eq. (12) estimates deviation from an analytical trajectory. Let $x_j^a(t)$ and $x_{j+1}^a(t)$ be two analytical solutions of Eq. (1) with boundary conditions

$$x_j^a(t-h) = x_{j-1}, \qquad x_j^a(t) = x_j, \tag{16a}$$

$$x_{j+1}^a(t) = x_j, \qquad x_{j+1}^a(t+h) = x_{j+1}, \tag{16b}$$

where x_{j-1}, x_j, and x_{j+1} are successive points on a numerical trajectory. We have

$$x_{j+1}^a(t+h) = x_j + \dot{x}_{j+1}^a(t)h + f(x_j)\frac{h^2}{2} + \ddot{x}_{j+1}^a(t)\frac{h^3}{6} + \dddot{x}_{j+1}^a(t)\frac{h^4}{24} + O(h^5) \tag{17a}$$

$$x_j^a(t-h) = x_j - \dot{x}_j^a(t)h + f(x_j)\frac{h^2}{2} - \ddot{x}_j^a(t)\frac{h^3}{6} + \dddot{x}_j^a(t)\frac{h^4}{24} + O(h^5). \tag{17b}$$

Taking into account that x_{j-1}, x_j, and x_{j+1} are related by Eq. (2) summation of Eqs. (17a), (17b) yields

$$\begin{aligned}&[\dot{x}_{j+1}^a(t) - \dot{x}_j^a(t)]h + [\ddot{x}_{j+1}^a(t) - \ddot{x}_j^a(t)]\frac{h^3}{6}\\ &+ [\dddot{x}_{j+1}^a(t) + \dddot{x}_j^a(t)]\frac{h^4}{24} + O(h^5) = 0.\end{aligned} \tag{18}$$

Thus, except for the special case $\dot{x}_{j+1}^a(t) \equiv \dot{x}_j^a(t)$ we have

$$\dot{x}_{j+1}^a(t) - \dot{x}_j^a(t) = O(h^3). \tag{19}$$

Assuming that $x^a(t)$ are smooth enough, relation (19) can be continued over a finite time interval and we may write

$$\begin{aligned}\dot{x}_j^a(t_j) - \dot{x}_0^a(t_j) &= \sum_{k=0}^{j} [\dot{x}_{k+1}^a(t_j) - \dot{x}_k^a(t_j)] \approx \sum_{k=0}^{j} \beta_k h^3\\ &= t_j \beta(t_j) h^2,\end{aligned} \tag{20}$$

where β_k are constant coefficients and $\beta(t_j)$ results from averaging over interval $(0, t_j)$ for a sufficiently small step size. Thus in Eq. (15) the second term appears to be $O(h^3)$ and it dominates over the truncation error given by the first term. Now, by performing the summation in Eq. (13) similarly to Eq. (20), we see that the deviation of the leapfrog trajectory from an analytical solution must be of the order of $O(h^2)$ in agreement with usual conclusions [7]. We see, therefore, that the square growth of the deviation of the coordinates from an analytical trajectory has a more complicated origin than the simple accumulation of truncation errors and it does not contradict the $O(h^4)$ truncation error of the algorithm.

Now let us consider the case when Q is the total energy. For any analytical solution its time derivative is zero and $\Delta\Delta E_j$ in Eq. (15) appears to be $O(h^4)$. The same result is obtained with the average total energy instead of the analytical value in Eq. (12). Summation in Eq. (13) now gives the global error on the order of $O(h^3)$, that is one order higher than the common conclusion [5, 14]. There is one complication, however, which was missed in the above derivations. We tentatively assumed in Eq. (14) that for any state on a numerical trajectory one can find a reference analytical solution passing through this state. For the next-step total energy to deviate as $O(h^4)$ both coordinates and velocities must be within $O(h^4)$ of this reference trajectory, which, in turn, requires that it pass through both (x_n, x_{n-1}) and $(v_{n-1/2}, v_{n-3/2})$, according to Eqs. (2) and (11), respectively. However, since Eq. (1) is only of second order, such an analytical solution does not necessarily exist. In the general case, we have two different analytical trajectories corresponding to coordinates and velocities and the overall single-step error in the total en-

ergy has a contribution from incoherence between these trajectories. On the other hand, if we use as a reference an analytical trajectory passing through a leapfrog state or a velocity Verlet state we obtain $O(h^3)$ single-step deviation and $O(h^2)$ global error, but this is only an upper estimate because it does not take into account the cancelation of local errors which allows solutions of Eqs. (3a), (3b) to fit higher order equations (2) and (11). We conclude, therefore, that the global error in the total energy for a leapfrog trajectory must be between $O(h^2)$ and $O(h^3)$, but it may depend upon the specific properties of Eq. (1). That is why the order of global errors in this case cannot be estimated analytically *a priori*, and should rather be measured numerically. We will see in the next section, however, that this appears difficult.

C. *Effects of Interpolations of Kinetic Energy*

We are going to show here that the common approach to testing energy conservation in molecular dynamics in the case of the leapfrog scheme in fact fails to evaluate the true error of the algorithm because of a dominating contribution of additional interpolations necessary to compute on-step kinetic energies. Following many previous studies, we first consider a simple harmonic oscillator as a model case suitable for analytical treatment. This simple model appears to have a significant predictive power because the fastest motions in real systems are nearly harmonic. We will check this on a more realistic example of protein dynamics.

Consider an oscillator with the Hamiltonian

$$H = \frac{1}{2}(\dot{x}^2 + \omega^2 x^2) \tag{21}$$

and the leapfrog equations of motion

$$v_{n+1/2} - v_{n-1/2} = -\omega^2 x_n h \tag{22a}$$

$$x_{n+1} - x_n = v_{n+1/2} h. \tag{22b}$$

It is known that these finite difference equations have an analog of the total energy [2]. By multiplying Eq. (22a) by $(v_{n+1/2} + v_{n-1/2})$ with simple algebra one obtains

$$\frac{1}{2}(v_{n+1/2}^2 + \omega^2 x_n x_{n+1}) = \frac{1}{2}(v_{n-1/2}^2 + \omega^2 x_{n-1} x_n) = \varepsilon_1. \tag{23}$$

Similarly, by multiplying Eq. (22b) by $(x_{n+1} + x_n)$ we obtain

$$\frac{1}{2}(v_{n-1/2} v_{n+1/2} + \omega^2 x_n^2) = \frac{1}{2}(v_{n-3/2} v_{n-1/2} + \omega^2 x_{n-1}^2) = \varepsilon_2 \tag{24}$$

and it is easy to check that $\varepsilon_1 = \varepsilon_2 = \varepsilon$. Thus, there is a quantity with the dimension of energy which is exactly conserved along the numerical trajectory. In the general case a numerical "first integral" analogous to ε does not exist, but it is very useful in analytical derivations and gives the possibility to reveal certain qualitative features intrinsic in the common approaches to the estimation of the on-step total energy. Consider the most common approximation of the on-step total energy for the leapfrog scheme

$$E_{lf}^n = \frac{1}{2}\left[\frac{1}{2}(v_{n-1/2}^2 + v_{n+1/2}^2) + \omega^2 x_n^2\right] = \varepsilon + \frac{\tau^2}{2} U_n, \tag{25}$$

where $\tau = \omega h$ is the reduced step size. Here we made a substitution of Eqs. (22a) and (24) and denoted potential energy as U_n. As expected, the total energy is not constant and it fluctuates with a small amplitude on the order of $O(\tau^2)$. The relative fluctuation, sometimes used as a convenient indicator of the accuracy of the trajectory, is simply

$$q_{lf} = \frac{D[E_{lf}]}{D[U]} = \frac{\tau^2}{2}, \tag{26}$$

where $D[\ ]$ denotes the operator of variance. The fluctuating term in Eq. (25) is usually used to check for the accuracy and stability of a computed trajectory. One can note, however, that the analytical form of this term is somewhat suspicious. Namely, it could have been expected that a numerical fluctuation of an analytically constant value has a more complicated form than just a scaled oscillation of the potential energy. In order to make clear the origin of this oscillation let us consider the related value for an exact analytical trajectory of an oscillator with $x(0) = 0$ and $v(0) = \omega$. We have

$$\begin{aligned} E_{lf}^n &= \frac{\omega^2}{2}\left\{\frac{1}{2}\left[\cos^2\omega\left(t - \frac{h}{2}\right) + \cos^2\omega\left(t + \frac{h}{2}\right)\right] + \sin^2\omega t\right\} \\ &= E_a\left(1 - \frac{\tau^2}{4}\right) + \frac{\tau^2}{2} U(t) + O(h^4), \end{aligned} \tag{27}$$

where E_a and $U(t)$ are analytical total and potential energies, respectively. Comparison of this result with Eq. (25) shows that the $O(\tau^2)$ oscillation is exactly same for the numerical trajectory as for the analytical one suggesting that it is introduced by the interpolation and has no relation to the accuracy of the trajectory. In order to validate this suggestion let us check what one gets when more accurate interpolation formulas are employed. Consider the interpolation

$$K_n = \frac{1}{8}(3K_{n+1/2} + 6K_{n-1/2} - K_{n-3/2}) + O(h^3), \tag{28}$$

where $K_{n-1/2}$ denotes the half-step kinetic energy. It is not difficult to check that when applied to an exact trajectory it gives an $O(\tau^3)$ oscillation around the correct total energy. For the leapfrog solution, derivations similar to that used for Eq. (25) result in

$$E_3^n = \varepsilon\left(1+\frac{\tau^2}{4}\right)+\frac{\tau^3}{2}\omega x_n v_{n-1/2}-\frac{\tau^4}{8}K_{n-1/2}. \quad (29)$$

Again we see that the amplitude of the oscillation scales as the order of the interpolation. Note that its phase is shifted by $\pi/2$ from that of E_{lf}. For the following discussion we need one more interpolation

$$K_n = \frac{1}{48}(15K_{n+1/2}+45K_{n-1/2}-15K_{n-3/2}+3K_{n-5/2})+O(h^4) \quad (30)$$

which gives

$$E_4^n = \varepsilon\left(1+\frac{\tau^2}{2}\right)+\frac{\tau^4}{16}(5K_{n-1/2}+K_{n-3/2}-4U_{n-1}) \quad (31)$$

with $O(\tau^4)$ oscillation, as expected. E_4 oscillates in phase with the kinetic rather than with potential energy, which means that it is shifted by π from E_{lf}.

One might conclude that in this special case the total energy is conserved exactly with any time step size. We will later see, however, that actually the leapfrog trajectory deviates from an exact constant energy hypersurface, but these deviations are regular and they appear to be compensated by the approximation errors of the interpolation formulas. This occasional cancelation certainly results from the fundamental solutions of the finite difference equation (2) being sine and cosine functions [2], but it is rather instructive because it demonstrates that interpolations can produce very misleading effects. The most interesting for us, however, is the form of the oscillations produced by the interpolation formulas. We will now see that qualitatively similar behavior is observed in real simulations.

Figures 1–3 present a 50 fs interval of a molecular dynamics trajectory computed with three different time steps: 0.5 fs, 0.1 fs, and 0.05 fs, respectively. The system modeled consists of an immunoglobulin binding domain of strepto-

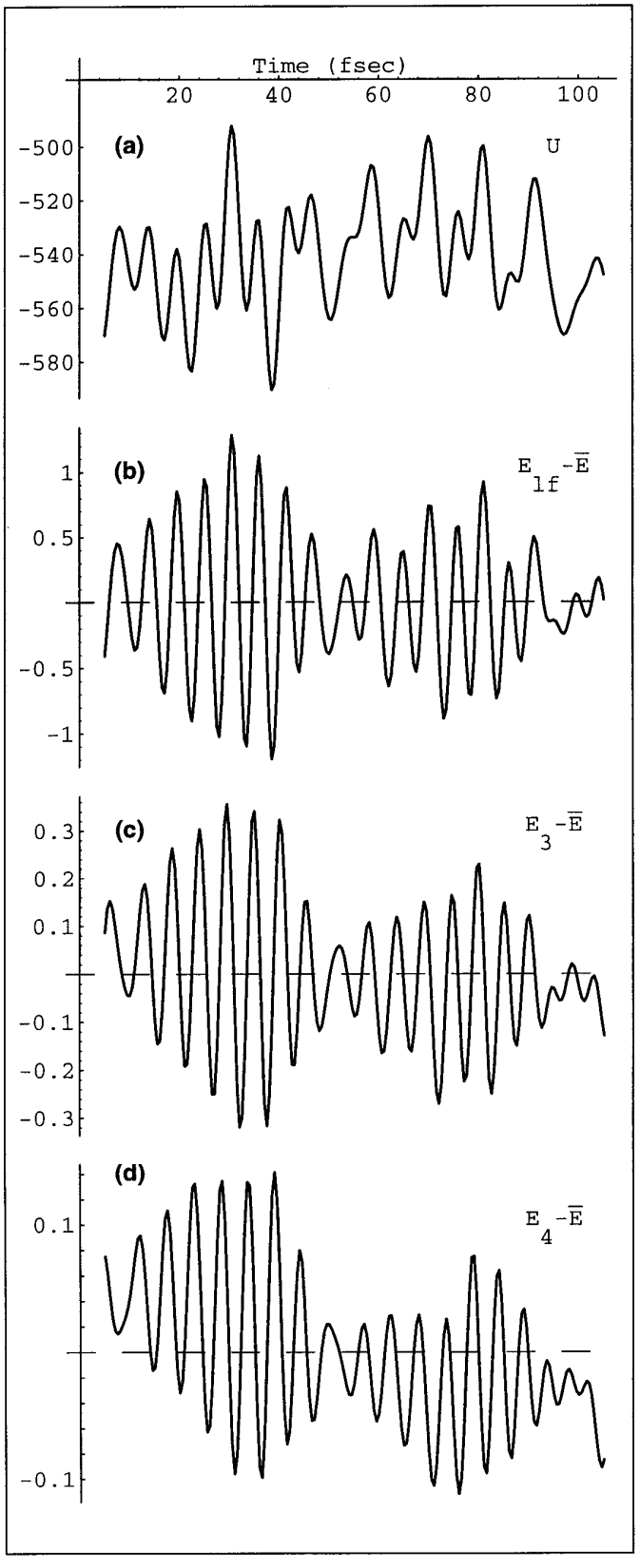

FIG. 1. Time dependence of the potential energy (a) and the total energy (c)–(d) for an unconstrained protein molecule. All energies are in kcal/mole. The molecular dynamics trajectory was computed with the leapfrog algorithm and a time step of 0.5 fsec. Three different estimates of the total energy for figures (c)–(d) were obtained with interpolations of second, third and fourth order, respectively, applied to half-step kinetic energies. The average total energy subtracted from the instantaneous values in figures (c)–(d) was 254.68 kcal/mole.

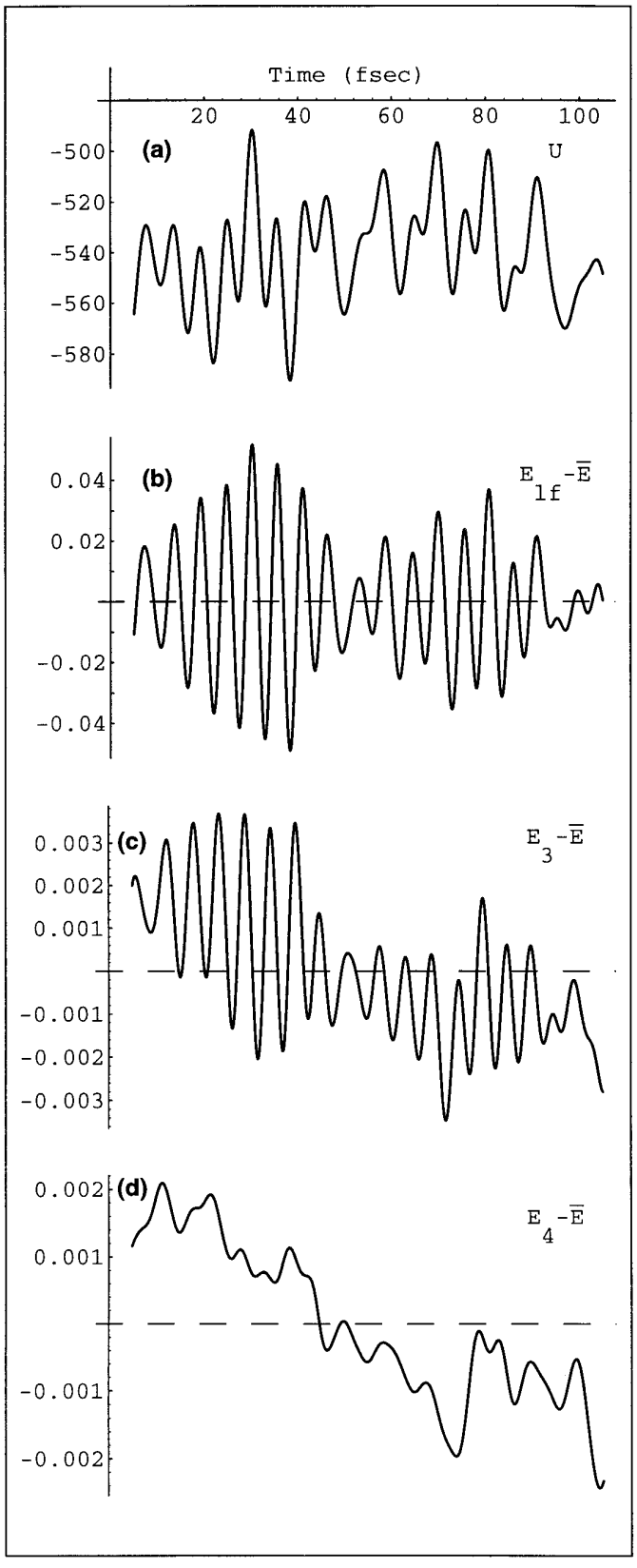

FIG. 2. Same as Fig. 1, but with the molecular dynamics trajectory computed with a time step of 0.1 fs. The average total energy in Figs. (c)–(d) was 255.42 kcal/mole.

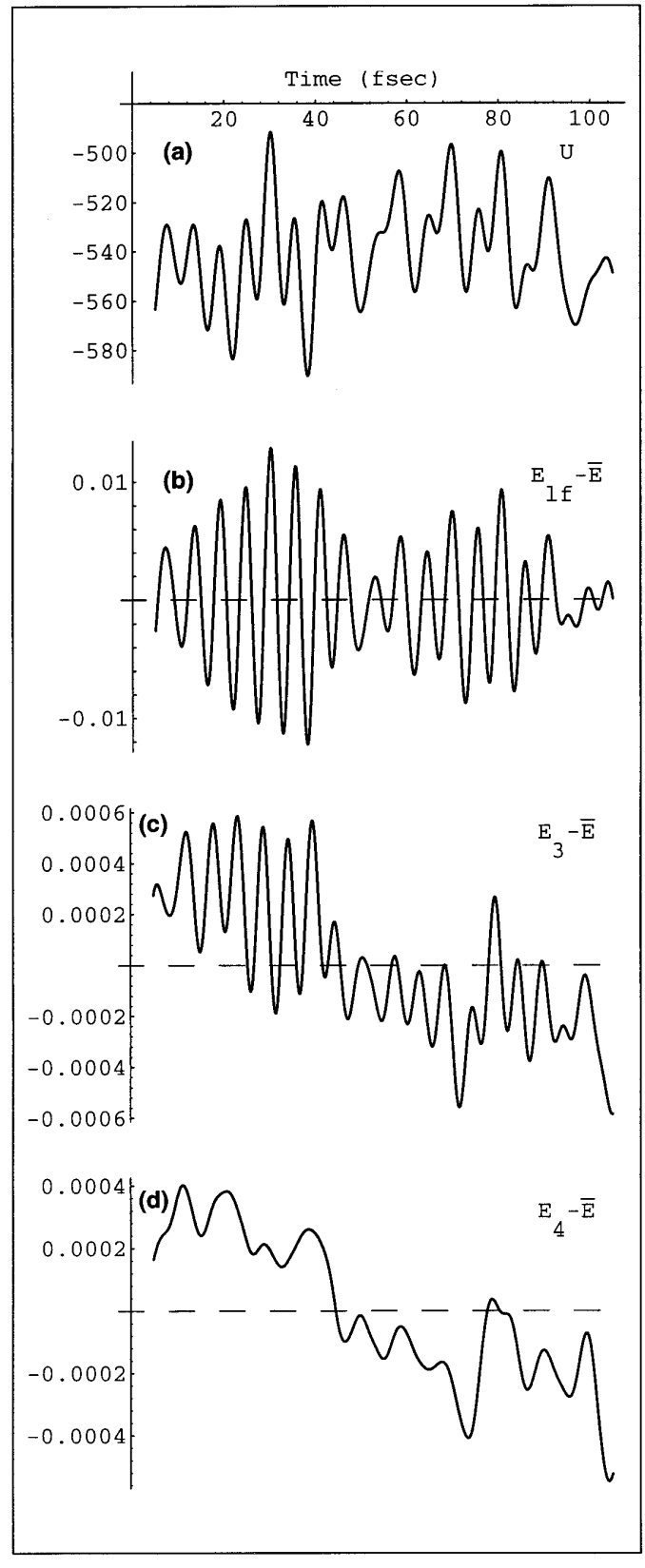

FIG. 3. Same as Fig. 1, but with molecular dynamics trajectory computed with a time step of 0.05 fs. The average total energy in Figs. (c)–(d) was 255.51 kcal/mole.

coccal protein G [19] which is a 56 residue α/β protein subunit (file 1pgb in the Protein Database [20]) with 24 bound water molecules presented in the crystal structure. All hydrogens are considered explicitly thus giving 927 atoms. The trajectory was computed without constraints upon the protein structure and with TIP3P water molecules held rigid. Other details of the simulation protocol are given in Section IID below. In such a system the fastest oscillations are due to bond stretching of hydrogens with a period of about 10 fs, which is clear in the time profiles of the potential energy shown in Figs. 1a–3a. Figures 1b–3b show the time profile of the total energy computed by Eq. (25). Note that according to Eqs. (25), (29), and (31) the amplitude of the oscillation of the total energy grows with the frequency, which means that, in systems of many oscillators, interpolations tend to filter lower frequencies. It is not surprising therefore that in Figs. 1b–3b the highest frequency strongly dominates compared with Figs. 1a–3a. Note, however, that in all three figures E_{lf} oscillation has exactly the same phase as that of the potential energy.

The profiles of the potential energy in Figs. 1a–3a are indistinguishable, which means that with a step size of 0.5 fs the trajectory is perfectly accurate. In agreement with the simple case considered above the shapes of the profiles of the total energy in Figs. 1b–3b also appear to be indistinguishable, with the amplitude of the oscillation scaled as the square of the step size. The identity of the three profiles in this case is not evident *a priori* and it is difficult to explain unless the above observations for an oscillator are not invoked. Thus, despite the simplicity of the model employed, Eq. (25) very accurately predicts both the growth rate with the step size and the shape of the fluctuation of the total energy in a much more complex system.

Now consider the time profiles of the total energy computed with higher order interpolations. Figures 1c and 1d show that both E_3 and E_4 behave similarly to E_{lf} and as predicted by Eqs. (29) and (31). Note that the oscillations of E_3 and E_4 are shifted by $\pi//2$ and π, respectively, from E_{lf} in Fig. 1b. With a reduced step size, however, both E_3 and E_4 reveal some new features. Similarly to E_{lf} the E_3 oscillation in Figs. 2c and 3c remains dominated by high-frequency oscillation, although its overall profile slightly changes. It is easy to see that the amplitude of the high-frequency oscillation scales as $O(h^3)$ and that the change in the profile is likely to be due to an underlying fluctuation which scales slower than $O(h^3)$. Finally, for E_4 oscillation in Figs. 2d and 3d we observe a profile of the fluctuation qualitatively different from those in Figs. 1a–d. The amplitude of the oscillation in Fig. 1d reduced by factors of 625 and 10^4 in Figs. 2d and 3d, respectively, would present a negligible part of the remaining fluctuation and that is why this hidden profile is revealed. This residual deviation of the total energy can be attributed to the interpolation-free error of the trajectory itself. Comparison of Figs. 2d and 3d shows that the amplitude of the remaining fluctuation scales faster than $O(h^2)$ but slower than $O(h^3)$, as we expected. If scaled back to the step size of 0.5 fs used in Fig. 1 this fluctuation can account for only a few percentages of the whole amplitude, which indicates that the total energy is really much better conserved than it appears to be. It is interesting that the high-frequency harmonic oscillation that dominates in Figs. 1b–d essentially disappears in Figs. 2d and 3d, suggesting that harmonic motions in the system really do not contribute to the apparent fluctuation of the total energy, which is produced by slower anharmonic motions.

Let us now summarize the above results. In the case of an oscillator, fluctuations are produced by interpolations only, their phases and amplitudes are related to those of the kinetic and potential energies in a simple way. For a realistic microcanonical ensemble with all common types of interactions, the total energy behaves similarly for the second- and third-order interpolations, as well as with the fourth-order one at a moderately small step size. For an extremely small step size, however, a qualitative difference is observed between the fourth and lower order interpolations. All this agrees very well with the global error in the total energy between $O(\tau^2)$ and $O(\tau^3)$ corresponding to $O(\tau^4)$ local error in the trajectory. Although the trajectory contribution can be revealed by interpolations of the fourth and higher orders, with practical time step values, the interpolation errors always dominate overwhelmingly. Taking into account that, because of the rapid growth of higher derivatives on repulsive walls of non-bonded potentials, higher order interpolations tend to be less accurate in ''less harmonic'' systems, we are obliged to conclude that straightforward interpolations in principle cannot adequately evaluate instantaneous on-step velocities and kinetic energies in the case of the leapfrog scheme. This approach, therefore, should not be used for assessing the accuracy of the algorithm.

D. *Alternative Strategy for Checking Energy Conservation*

In this section we consider how numerical tests with microcanonical ensembles can be arranged so that the real accuracy of conservation of the total energy can be assessed. The main idea is simple. Note that all interpolations of the kinetic energy such as Eqs. (28) and (30) give the same average kinetic energy. For any sufficiently long analytical trajectory these interpolations, upon averaging, result in a correct average kinetic energy and, consequently, exact total energy. This property does not depend upon the specific form of the Hamiltonian or the number of degrees of freedom. Note, for example, that in Figs. 1–3 with the same step size, E_{lf}, E_3, and E_4 oscillate around the same average. At the same time there is a distinguishable

difference between the averages for the three given values of the step size. We see, therefore, that unlike instantaneous energies, their time averages appear to be free from interpolation errors and they represent adequate indicators of the accuracy of the energy conservation.

Suppose we know the analytical value of the total energy. We could then repeatedly calculate a long enough test trajectory with gradually growing time steps and compare the average total energy with the analytical one. As long as these two values are close to each other, the computed trajectory remains close to a correct hypersurface in phase space. Instead of the unknown analytical total energy, we can use that obtained for the same trajectory with a very small step size. It is important to make sure, however, that the test trajectory starts from the same state on a fixed constant-energy hypersurface with each step size, which is not automatic in the case of the leapfrog scheme. Let us consider the implementation of this testing procedure for a harmonic oscillator.

By using Eqs. (22a) and (24) one can derive an identity

$$U_n \equiv \frac{\varepsilon}{2} + \frac{1}{2}\left[U_n - \frac{1}{2}(K_{n-1/2} + K_{n+1/2}) \right] + \frac{\tau^4}{4} U_n. \quad (32)$$

Denoting

$$\delta_n = U_n - \frac{1}{2}(K_{n-1/2} + K_{n+1/2}), \quad (33)$$

we obtain

$$U_n = \frac{1}{2(1 - \tau^2/4)}(\varepsilon + \delta_n). \quad (34)$$

It can be shown by a straightforward solution of the finite difference Eqs. (22a), (22b), that within the time step range of stability of an oscillator $\overline{U} = \overline{K}$. Therefore δ_n presents an oscillation around a zero average and we have

$$\overline{E}_{lf} = 2\overline{U} = \frac{\varepsilon}{(1 - \tau^2/4)}. \quad (35)$$

Let us now check how accurately the numerical average total energy approximates exact values, that is total energies corresponding to constant-energy hypersurfaces sampled by a leapfrog trajectory. Consider an analytical trajectory passing through a leapfrog state $(v_{n-1/2}, x_n)$. We have

$$\varepsilon = E_a \left\{ \sin^2 \omega t + \cos \omega \left(t - \frac{h}{2} \right) \left[\cos \omega \left(t - \frac{h}{2} \right) - \tau \sin \omega t \right] \right\}. \quad (36)$$

From Eqs. (35) and (36) by using a Taylor series expansion, we obtain

$$\frac{E_a}{\overline{E}_{lf}} = 1 + \left[\frac{\tau^3}{48} + O(\tau^5) \right] \sin 2\omega t - \left[\frac{\tau^4}{96} + O(\tau^6) \right] \cos 2\omega t - \frac{\tau^4}{96} - \frac{\tau^6}{64} + O(\tau^8). \quad (37)$$

The r.h.s. of Eq. (37) involves two qualitatively different types of deviations. The first depends upon the current phase of the oscillation and it in fact presents the true measure of the energy conservation along the trajectory: as long as the phase-dependent terms in Eq. (37) are small, all states on the trajectory belong, or are close, to the same exact constant energy hypersurface. When these terms become too large, successive leapfrog states correspond to different hypersurfaces, but the corresponding analytical total energies oscillate around a certain value. The second type of deviation is phase-independent and it is presented by the fourth- and sixth-order terms. Because of this deviation, with large step size all leapfrog states appear to belong to hypersurfaces of a systematically lower energy than the computed average value, which means that they are no longer sampled ergodically. The magnitudes of both these types of deviations can be easily evaluated, and it turns out that for $\tau \lesssim 1.1$ the fluctuating part dominates, while above this level the phase-independent contribution rapidly becomes overwhelming due to its sixth-order growth.

Now consider what one would observe in the numerical tests outlined above. It can be seen from Eq. (35) that by simply increasing the step size one obtains an $O(\tau)$ regular growth of $\overline{U}$ and $\overline{E}_{lf}$. It is clear, however, that, in this case, the starting leapfrog state in phase space is effectively moved from one constant energy hypersurface to another, which results in a regular drift of averages. In order to remove this drift, the trajectory should start from the same hypersurface, that is from x_0 and $v_{-1/2}$ corresponding to the same analytical trajectory. It is easy to see that this case is also described by Eq. (37) if it is read in an opposite sense. Namely, now E_a and the phase of the oscillating terms on the right are constant because they correspond to the initial constant energy hypersurface and the phase of the starting state. Equation (37) therefore describes the step size dependence of E_{lf} which beyond $\tau \approx 1.1$ should rapidly deviate upward from its zero step size limit. It turns out that this estimate holds quite well for representative molecular models as well, which is illustrated by Fig. 4.

This figure presents an example of application of the proposed test to protein dynamics. The simulations were made by AMBER molecular modeling program [21] with AMBER94 force field [22]. The system considered was same as in Figs. 1–3, that is with only water bond lengths and bond angles constrained by the SHAKE algorithm

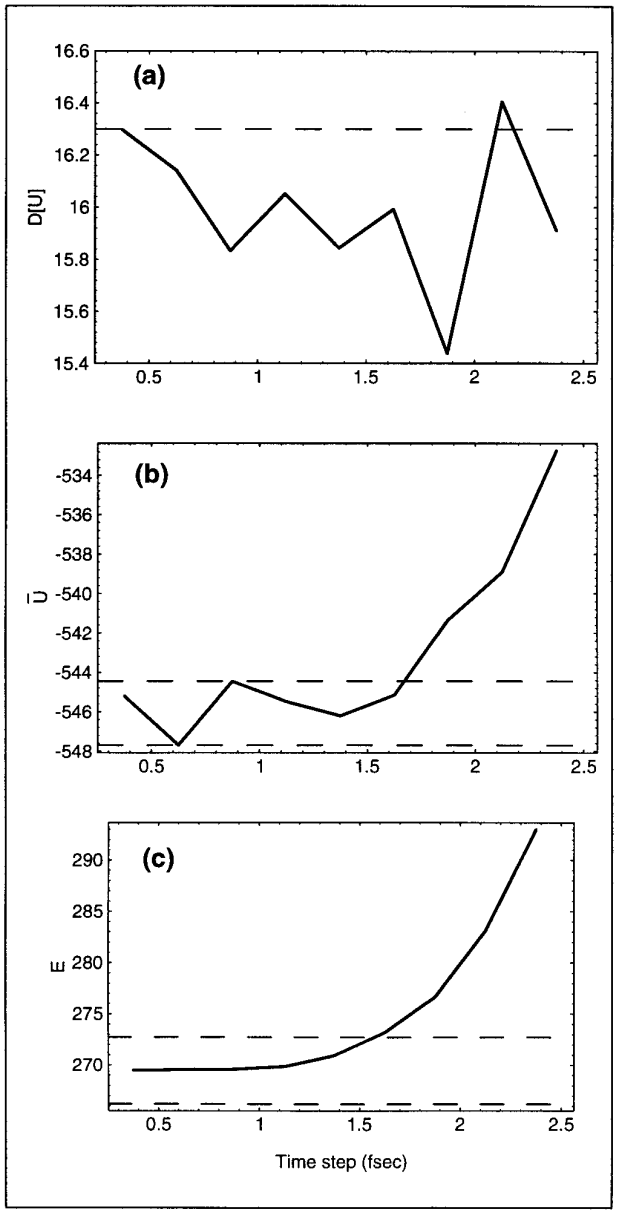

FIG. 4. Time step dependencies of the potential energy (b), its time variance (a) and the total energy (c) computed over a 10 ps molecular dynamics trajectory. All energies are in kcal/mole. The dotted lines in (b) and (c) show the corresponding bands of acceptable deviation defined as described in the text from the low time step limit of variance indicated by the dotted line in (a).

[23]. Initial data for these tests were prepared as follows. The equilibration was more or less standard and included minimization of the crystal structure followed by a 12.5 ps run starting from Maxwell distribution at 300K with periodic temperature control and a step size of 0.5 fs. After that the step size was reduced to 0.25 fs and a short trajectory of 150 fs was calculated, with the final part stored and used in place of an analytical trajectory for generating initial data. The last point was used as the starting state for a 10 ps test trajectory which was computed with gradually growing step sizes and half-step velocities taken at appropriate time intervals from coordinates.

In order to apply our test to a real molecular system we should also take into account that, since in common empirical potentials the absolute value of the potential energy involves many nonharmonic contributions which always remain far from zero, it is no longer sensible to compare the deviations with the absolute energy values. An appropriate natural scale, however, is given by the variance of the instantaneous potential energy over the test trajectory. One may reasonably consider for $\overline{U}$ the deviation of $\pm 0.1 D[U]$, for instance, as an acceptable level of accuracy. Note that in the oscillator case this gives approximately same criterion as above. The deviation of E_{lf} should be two times larger because it involves a similar contribution from the kinetic energy.

Figure 4a shows the step size dependence of $D[U]$. The dotted line in this figure shows the low step size level used to plot the corresponding bands of acceptable accuracy in Figs. 4 (b) and (c). As seen in Fig. 4(b) the fluctuation of $\overline{U}$ remains within the band $\pm 0.1 D[U]$ up to a time step of 1.7 fs. These fluctuations are relatively large because they are affected by rare conformational transitions which cannot be averaged during a test trajectory. Oppositely, E_{lf} shown in Fig. 4c grows steadily, but its deviation also remains within the band of acceptable accuracy up to the same step size. Assuming that this value corresponds to $\tau \approx 1.1$ one obtains a frequency of 3400 cm^{-1}, that is exactly that of bond stretching of peptide hydrogens. The whole spectrum of bond stretching modes of hydrogens covers the range 3000–3700 cm^{-1}, which corresponds to characteristic step sizes in the range 1.6–1.9 fs. We may conclude, therefore, that Fig. 4 demonstrates a good agreement with the single oscillator model considered above.

At this point it is convenient to compare our step size estimates with the common recommendations. Although no strict rules exist, it is usually considered that the computed trajectory provides an acceptably accurate sampling when the relative fluctuation q_{lf} given by Eq. (26) is less than 0.1, which gives $\tau \lesssim 0.4$, i.e., at least 14 time steps per single cycle of the oscillation. Our estimates show that actually this level of accuracy is reached with an almost three times larger time step, although in this case q_{lf} is already so large that the energy conservation seems to be lost. In practice, however, the above recommendations are almost never respected and most often molecular dynamics trajectories are computed with less than 10, or even less than 5 steps, per the shortest cycle [24]. This practice thus appears to be well justified and it is clear from Fig. 4 that the leapfrog trajectory really manages to hold to a constant energy hypersurface with somewhat larger time steps than commonly recommended.

E. *Leapfrog Scheme versus Its Relatives*

In this section we will demonstrate that, contrary to the conventional point of view, the leapfrog scheme exhibits exceptional properties compared with the other algorithms of this group.

The results obtained in Section IIC above obviously are not applicable to the velocity Verlet and Beeman algorithms, Eqs. (4)–(5). In these cases, interpolations appear to be built into the algorithms, which makes them only $O(h^3)$ accurate in velocities. (Note that interpolation (7) has the same order of truncation as that in Eq. (6).) Oscillations of the instantaneous total energy result from the algorithms themselves and they grow simply as $O(h^2)$ in agreement with the order of the truncation error. Let us look more thoroughly at the total energy computed by the velocity Verlet integrator. Similarly to Eq. (25) we obtain for the oscillator

$$E_{vv}^n = \frac{1}{2}\left[\frac{1}{4}(v_{n-1/2} + v_{n+1/2})^2 + \omega^2 x_n^2\right] = \varepsilon + \frac{\tau^2}{4} U_n. \quad (38)$$

We see that E_{vv} behaves similarly to E_{lf} with two times smaller amplitude of the oscillation and always $E_{vv} < E_{lf}$, which means that, in the case of velocity Verlet, the temperature estimated for the same trajectory is somewhat lower. In real simulations q_{vv} is normally smaller by a factor of 0.5–0.7 and the corresponding temperature can be lower by as much as a few degrees. The origin of these differences is clear from the previous discussion and it is not very interesting.

Since $\overline{E}_{lf}$ and $\overline{E}_{vv}$ are $O(\tau^2)$ different it is clear that $\overline{E}_{vv}$ gives one order less accurate estimate of the total energy of the analytical trajectory approximated by leapfrog states. The same order of approximation is obtained if we consider the trajectories passing through velocity Verlet states. In this case the analytical energies are given simply by Eq. (38), and we have

$$E_a - \overline{E}_{vv} = \frac{\tau^2}{4}(U_n - \bar{U}) \quad (39)$$

which shows that successive states belong to hypersurfaces of $O(\tau^2)$ different energies. If we consider E_a and U_n in Eq. (39) as constants and $\overline{E}_{vv}$ and τ as variables we obtain the rate of growth of the deviation of $\overline{E}_{vv}$ with the step size. We see that it is on the order of $O(\tau^2)$, i.e. similar to the leapfrog scheme with fixed ε according to Eq. (35). All these derivations can be readily reproduced for Beeman and the original Störmer algorithms which both deviate from their starting hypersurfaces as $O(\tau^2)$.

The above results can be summarized as follows. Consider a numerical trajectory of an oscillator in phase space. It may be represented by Störmer–Verlet states (x_n, x_{n-1}), velocity Verlet states (x_n, v_n) or by leapfrog states $(x_n, v_{n-1/2})$. According to Eq. (37), the leapfrog states sample from constant energy hypersurfaces which are $O(\tau^3)$ close to each other, while in the former two cases a broader $O(\tau^2)$ spectrum of energies is covered. In this sense one can say that the leapfrog trajectory presents the most accurate sampling among the three representations.

III. CONCLUSIONS

The results presented in the paper shed some light upon the long-standing confusion involved in the conventional interpretation of the common algorithms of the classical molecular dynamics. The main practical result is a new simple testing scheme for energy conservation, which gives a better estimate of the quality of leapfrog trajectories. It is shown that the leapfrog trajectories provide accurate sampling in phase space with large time step values, when the apparent total energy obtained by routine interpolations is no longer conserved. This explains the recognized quality of thermodynamic averages at large step size. The new testing scheme should be particularly useful for studying the time step limitations in various molecular models appearing in internal coordinate molecular dynamics simulations of polymers [25].

REFERENCES

1. L. Verlet, Comupter ''experiments'' on classical fluids. 1: Thermodynamic properties of Lennard-Jones molecules, *Phys. Rev.* **159**, 98 (1967).
2. R. W. Hockney and J. W. Eastwood, *Computer Simulation Using Particles* (McGraw–Hill, New York, 1981).
3. W. C. Swope, H. C. Andersen, P. H. Berens, and K. R. Wilson, A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters, *J. Chem. Phys.* **76,** 637 (1982).
4. D. Beeman, Some multistep methods for use in molecular dynamics calculations, *J. Comput. Phys.* **20,** 130 (1976).
5. M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids* (Clarendon Press, Oxford, 1987).
6. J. M. Haile, *Molecular Dynamics Simulations: Elementary Methods* (Wiley-Interscience, New York, 1992).
7. G. D. Venneri and W. G. Hoover, Simple exact test for well-known molecular dynamics algorithms, *J. Comput. Phys.* **73,** 486 (1987).
8. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Engelwood Cliffs, NJ, 1971).
9. M. E. Tuckerman, B. J. Berne, and G. J. Martyna, Reversible multiple time schale molecular dynamics, *J. Chem. Phys.* **97,** 1990 (1992).
10. S. Toxvaerd, Comment on: Reversible multiple time scale molecular dynamics, *J. Chem. Phys.* **99,** 2277 (1993).
11. A. R. Janzen and J. W. Leech, Periodic multistep methods in molecular dynamics, *Comput. Phys. Commun.* **32,** 349 (1984).
12. S. P. Auerbach and A. Friedman, Long-time behaviour of numerically computed orbits: Small and intermediate timestep analysis of one-dimensional systems, *J. Comput. Phys.* **93,** 171 (1991).

13. S. Toxvaerd, Molecular dynamics at costant temperature and pressure, *Phys. Rev. E* **47,** 343 (1993).
14. W. F. van Gunsteren and H. J. C. Berendsen, Algorithms for macromolecular dynamics and constraint dynamics, *Mol. Phys.* **34,** 1311 (1977).
15. D. MacGowan and D. M. Heyes, Large timesteps in molecular dynamics simulations, *Mol. Simul.* **1,** 277 (1988).
16. D. Fincham, Choice of time step in molecular dynamics simulations, *Comput. Phys. Commun.* **40,** 263 (1986).
17. M. E. Tuckerman, B. J. Berne, and G. J. Martyna, Reply to comment on: Reversible multiple time scale molecular dynamics, *J. Chem. Phys.* **99,** 2278 (1993).
18. P. M. Rodger, On the accuracy of some common molecular dynamics algorithms, *Mol. Simul.* **3,** 263 (1989).
19. T. Gallagher, P. Alexander, P. Bryan, and G. L. Gilliland, Two crystal structures of the B1 immunoglobulin-binding domain of streptococcal protein G and comparison with NMR, *Biochemistry* **33,** 4721 (1994).
20. F. C. Bernstein *et al.*, The protein data bank: A computer-based archival file for macromolecular structures, *J. Mol. Biol.* **112,** 535 (1977).
21. D. A. Pearlman *et al.*, *AMBER 4.1* (University of California, San Francisco, 1995).
22. W. D. Cornell *et al.*, A second generation force field for the simulation of proteins, nucleic acids and organic molecules, *J. Amer. Chem. Soc.* **117,** 5179 (1995).
23. J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen, Numerical integration of the cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes, *J. Comput. Phys.* **23,** 327 (1977).
24. M. Levitt, Molecular dynamics of native protein. 1: Computer simulation of trajectories, *J. Mol. Biol.* **168,** 595 (1983).
25. A. K. Mazur, results prepared for publication.