/ /3

Reprinted from JOURNAL OF COMPUTATIONAL PHYSICS
All Rights Reserved by Academic Press, New York and London

Vol. 73, No. 2, December 1987
Printed in Belgium

Note

Simple Exact Test for Well-Known Molecular Dynamics Algorithms*

Computer simulation via molecular dynamics is now a widely used approach to the study of condensed matter physics. It consists in the solution of the equations of motion for a large number of particles, interacting with forces suited to model fluids, solids, clusters, surfaces, etc.

A number of well-known algorithms are commonly used for the numerical solution of the equations of motion $m\ddot{x} = F(x)$. They are still based on methods nearly 100 years old. (A clear and comprehensive review was written by Milne in 1953 [1]). The choice of a particular algorithm is generally done on the basis of the accuracy of the solution versus computer cost (time and memory) and difficulty of programming. The accuracy for a given amount of computer time varies with the number of coupled equations and the form of the force law $F(x)$. In a typical system studied with molecular dynamics the number of particles N ranges from 2 to 160,000 and the force F is highly nonlinear. The evaluation of the accuracy becomes difficult and imprecise. A check for a lower bound on the acceptable accuracy is the conservation of the constants of motion (typically the energy). It is difficult to formulate a simple useful criterion to evaluate the performance of the various algorithms.

Recently Berendsen and Van Gunsteren [2] compared six numerical methods commonly used in molecular dynamics, in a simple and more useful way than previous workers [3]. The comparison was made by applying *all* the algorithms to the same one-dimensional harmonic oscillator. Because it provides an exact picture of the relative accuracies and efficiencies of the different methods (the *exact* solution being available as a reference), this idealized special case is a useful caricature of the more realistic many-body problems to which molecular dynamics is generally applied. Indeed, although the simple linear force is far from being a complete model for the particle interactions in fluids and solids, the corresponding particle trajectories in many-body systems do closely resemble the simple harmonic oscillations (see, for example, Fig. 3 in the review by Hoover and Ashurst [4]). This is a consequence of the fact that in solids and dense fluids the particles are seldom far from the local potential minimum.

For the simple one-dimensional harmonic case, the difference equations

* Work performed under the auspices of the U. S. Department of Energy at Lawrence Livermore National Laboratory, under Contract W-7405-ENG-48.

associated with various algorithms can be exactly solved analytically (solutions can be found scattered in numerical analysis textbooks). The comparison can therefore be made even more precise and the behavior of the various numerical solutions can be understood in terms of phase shifts and decaying rates. In what follows we discuss these exact solutions of the difference equations associated with the algorithms considered by Berendsen and Van Gunsteren. We also add for comparison the Runge–Kutta method, because it is one of the most widely used in molecular dynamics.

The purpose of this note is twofold: (i) to gather in one place (see Fig. 1) the most commonly used algorithms in traditional molecular dynamics and (ii) to suggest a simple criterion to compare their performances (see Fig. 2).

The results reviewed here are strictly true only for the case of the one-dimensional harmonic oscillator. Many other techniques, not discussed here, could be applied to that problem. We feel that the exact unified picture available for this simple system, using many-body techniques, serves as a useful reference for the more complicated many-body systems of molecular dynamics. We consider approximate

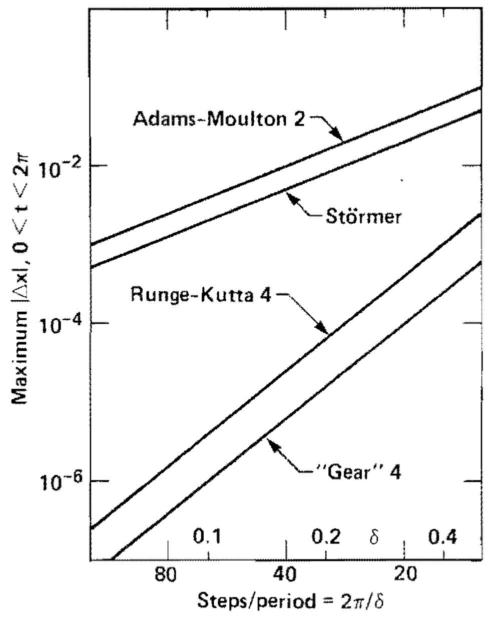

FIG. 1. Relative accuracies of approximate solutions of the one-dimensional harmonic oscillator. The largest deviation $\Delta x = x_{\text{approximate}} - x_{\text{exact}}$ during the first oscillator period is compared as a function of step length δ for various algorithms. (The solution of Beeman's method coincides with the one of Störmer's method.) For the implicit methods, Adams–Moulton and "Gear" 4, the results for the converged corrector schemes are plotted.

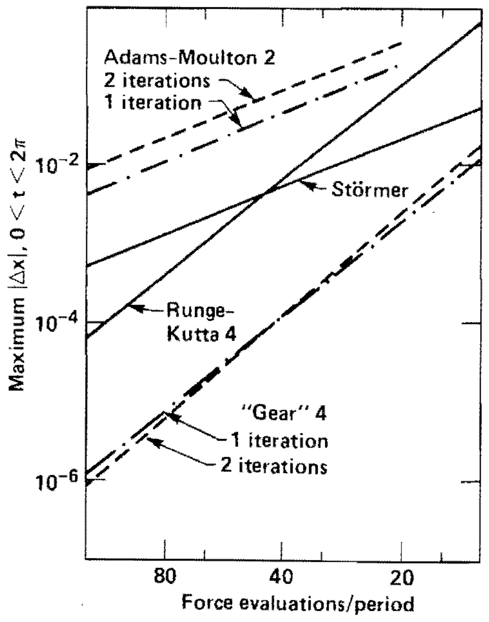

FIG. 2. Relative efficiencies of the algorithms of Fig. 1. The largest deviation Δx is here plotted as function of the number of force-evaluations (as a measure of computer time). For the implicit methods, Adams-Moulton and "Gear" 4, the results for 1 and 2 iterations are plotted.

solutions of the second-order ordinary differential equation of motion for a one-dimensional harmonic oscillator (with unit mass and unit spring constant)

$$\ddot{x} = -x,$$

which is equivalent to the pair of first-order differential equations

$$\dot{x} = v, \qquad \dot{v} = -x.$$

With the initial conditions $x_0 = 1$ and $v_0 = 0$ the analytic solution is

$$x = \cos(t).$$

The "Verlet" centered-difference algorithm [5] was used by Cowell and Crommelin [6] 77 years ago to calculate the orbit of Halley's comet, and is attributed by Milne to the work of Störmer in 1907. It is equivalent to a "leap-frog" version of the two first-order equations given above. It can be used for the particular case of Newton's second-order differential equation, which has no explicit dependence on the first derivative. The centered-difference Störmer equation

$$x_{n+1} - 2x_n + x_{n-1} = -x_n\,\delta^2, \qquad t = n\delta \tag{1.0}$$

is symmetric and therefore preserves the time-reversible character of the original second-order differential equation. The solution satisfying our initial condition $x_0 = 1$ and $x_{+1} = x_{-1}$ is a cosine function with a scaled frequency,

$$x_n = \cos(n\,\delta\lambda) \tag{1.1}$$

where λ is

$$\lambda = \frac{1}{\delta}\cos^{-1}\left(1 - \frac{\delta^2}{2}\right) \sim 1 + \frac{1}{24}\delta^2 + \frac{3}{640}\delta^4. \tag{1.2}$$

For a fixed value of the time step δ the deviation Δx from the true solution $\Delta x = \cos(\lambda t) - \cos(t)$ can be expressed in terms of a time-dependent phase shift, $\Phi = (\lambda - 1)\,t$. The phase shift increases linearly with time, so that the two solutions first become 180° out of phase, and then in phase again, at integral multiples of the recurrence time $\rho = 2\pi/(\lambda - 1)$ ($= 2400$ periods of the oscillator for $\delta = 0.1$). The corresponding trajectory deviation varies, for λ close to 1, as $\Delta x \sim -\Phi \sin(t) - (\Phi^2/2)\cos(t)$, where the first term $\Delta x \sim (-\delta^2/24)t\sin(t)$ dominates, at times small compared to $2/(\lambda - 1)$ (764 oscillator periods for $\delta = 0.1$). We can then expect the amplitude of the deviation to increase in time, initially linearly, up to a maximum value of 2, when the phase shift is exactly 180°, and then decrease again to 0 as the phase shift approaches 360°. On the other hand, for a fixed value of the time (still small compared to $2/(\lambda - 1)$), this amplitude is a linear function of $(\lambda - 1)$, and, from Eq. (1.2), a quadratic function of δ. This can also be deduced from Fig. 1, where the largest deviation of the trajectory during the first oscillator period is plotted as a function δ: the slope of the line corresponding to the Störmer algorithm is 2. We follow Berendsen and Van Gunsteren in defining the "apparent order" of the algorithm as the smallest power of δ in the trajectory deviation at a fixed time. We then assign to Störmer's algorithm an apparent order of 2. More often, however, the "order" of algorithm refers to the local accuracy, defined as the last correct term in the truncated series expansion for a single time step. With such a definition Störmer's scheme would be a third-order algorithm, rather than second, because the first neglected term in the expansion is $-x_n(\delta^4/12)$.

The other commonly used algorithms for molecular dynamics are methods (of second- and higher order) for solving first-order differential equations and can be grouped into two categories: those derived from Adams implicit methods and the Runge–Kutta explicit methods. In implicit methods a prediction for the position is made; velocity and force are next evaluated and used for correcting the position. The procedure can then be iterated, using the new position (and velocity) as a new prediction. Because convergence is rapid, the solution of the algorithm is essentially defined by the corrector equations, to which the solution typically converges closely in one or two iterations.

There are a number of proposed second-order schemes, of which we will consider three predictor-corrector implicit methods: the "modified-Euler" method mentioned in [2], the method used by Rahman [7], and its version suggested by Beeman [8].

The "modified-Euler" and Rahman methods use an Adams–Moulton second-order corrector, but the predictors differ slightly. The predictors are respectively: $x_{n+1} = x_n + v_n\,\delta + a_n(\delta^2/2)$ in the "modified-Euler" method and $x_{n+1} = x_{n-1} + 2v_n\,\delta$ in the method used by Rahman. The second-order Adams–Moulton corrector has the form of the well-known trapezoidal formula:

$$\begin{aligned} v_{n+1} &= v_n + (a_{n+1} + a_n)\frac{\delta}{2} \\ x_{n+1} &= x_n + (v_{n+1} + v_n)\frac{\delta}{2}. \end{aligned} \tag{2.0}$$

For the harmonic oscillator $a_n = -x_n$ so that the system of two equations for the corrector can be combined into one equation,

$$x_{n+1} - 2x_n + x_{n-1} = -x_n \frac{\delta^2}{1 + \delta^2/m}, \qquad m = 4, \tag{2.1}$$

which is identical to Eq. (1.0), but with a scaled time step. The solution is of the same form as Eq. (1.1), but the multiplicative factor λ is now given by

$$\lambda = \frac{1}{\delta}\cos^{-1}\left[1 - \frac{\delta^2}{2(1 + \delta^2/4)}\right]. \tag{2.2}$$

Scaling of the time step does not change the apparent order of the algorithm, but produces a trajectory deviation, $\Delta x \sim (\delta^2/12)\,t\sin(t)$, twice as large as the Störmer method and with the opposite sign. Both Rahman's and the "modified-Euler" methods correspond therefore, in the limit of an infinite number of iterations, to the Störmer scheme, with a slightly worse accuracy. It is interesting to note that the "modified-Euler" predictor has the same form as the second-order Runge–Kutta method. For an application see Lorenz [9].

Beeman [8] proposed a combination of explicit predictor (P) and implicit corrector (C) formulas that appears to increase the accuracy in the evaluation (E) of the velocities. His algorithm can be used in two different ways:

(i) PEC(v): the prediction is used to evaluate the force and correct the velocity, but no correction is performed on the position. This scheme is identical to Störmer's [2];

(ii) PEC(r) EC(v): the prediction is used to evaluate the force, the position is corrected, the force is evaluated again and used to correct the velocity.

The second scheme can be iterated. But for the harmonic oscillator it converges in one step to an equation of type (2.1) with $m = 6$. The dominant term at small times in the trajectory deviation, $\Delta x \sim (\delta^2/24)\,t\sin(t)$, is as large as in Störmer's case, but with the opposite sign. This second version of the Beeman algorithm is then again equivalent to Störmer's. Therefore the first version, which requires half as many force evaluations, is preferred.

Simplicity can be sacrificed for a 100-fold gain in accuracy by using higher order methods. As a representative higher order implicit algorithm we will consider here the algorithm referred to as "Gear" 4 by Berendsen and Van Gunsteren [2]. It is a generalization of the Nordsieck formulation to second-order differential equations and essentially solves the differential equations not only for the position and the velocity but also for the second and the third derivatives of the position. We refer to the original work for details [10]. What is particularly interesting is the simple form assumed by the corrector in the so-called force representation [8]:

$$\begin{aligned} v_{n+1} &= v_n + (5a_{n+1} + 8a_n - a_{n-1})\frac{\delta}{12} \\ x_{n+1} &= x_n + v_n\,\delta + (a_{n+1} + 6a_n - a_{n-1})\frac{\delta^2}{12}. \end{aligned} \tag{3.0}$$

Although the equations *appear* irreversible, for the harmonic oscillator they can be reduced again to one single reversible equation identical to (2.1), with $m = 12$ in the scaled time step. To show this, substitute $v_n - v_{n-1}$, from the first of Eq. (3.0) into the expression for $(x_{n+1} - x_n) - (x_n - x_{n-1})$ obtained by applying the second of Eq. (3.0) at two successive time steps. In this case the scaling of the time step produces an extra term in δ^2 in the trajectory deviation that exactly cancels the term in δ^2 present in Störmer's solution. The dominant term in the trajectory deviation becomes $\Delta x \sim -(\delta^4/480)\,t\sin(t)$ and therefore the algorithm has an apparent order of 4. The time dependence of the error is nevertheless the same as in the other reversible examples, namely a phase shift that increases linearly in time, causing the true and the numerical solutions to be exactly 180° out of phase and then in phase again and so on, with a period much longer than was the case for our second-order algorithms (three orders of magnitude larger than for the Störmer's algorithm with $\delta = 0.1$).

The last example we will consider is the explicit Runge–Kutta algorithm, which involves calculating the forces at intermediate steps (four in the most commonly used fourth-order version). The numerical fourth-order equations for the velocity and the position of the harmonic oscillator assume a simple form:

$$\begin{aligned} v_{n+1} &= \left(1 - \frac{\delta^2}{2} + \frac{\delta^4}{24}\right) v_n - \left(\delta - \frac{\delta^3}{6}\right) x_n \\ x_{n+1} &= \left(1 - \frac{\delta^2}{2} + \frac{\delta^4}{24}\right) x_n + \left(\delta - \frac{\delta^3}{6}\right) v_n. \end{aligned} \tag{4.0}$$

Again they can be rewritten as one single equation:

$$x_{n+1} - 2x_n + x_{n-1} = -x_n\,\delta^2\left(1 - \frac{\delta^2}{12}\right) + x_{n-1}\frac{\delta^6}{72}\left(1 - \frac{\delta^2}{8}\right). \tag{4.1}$$

This algorithm is irreversible. The solution satisfying the initial conditions $x_0 = 1$ and $v_0 = 0$ is in this case a cosine function with a time dependent amplitude,

$$x_n = \left[1 - \frac{\delta^6}{72}\left(1 - \frac{\delta^2}{8}\right)\right]^{n/2} \cos(n\,\delta\lambda),$$

where the factor λ becomes

$$\lambda = \frac{1}{\delta}\tan^{-1}\left[\frac{\delta(1-\delta^2/6)}{1-\delta^2/2+\delta^4/24}\right] \sim 1 - \frac{\delta^4}{120}.$$

For any useful value of $\delta(\delta < 2\sqrt{2})$ the amplitude of the solution slowly decays to zero.

The accuracies of the algorithms are compared as functions of the time step δ in Fig. 1. We plot there the maximum deviation from the true trajectory $\Delta x = x_{\text{approximate}} - x_{\text{exact}}$ during the first oscillator period, according to the analytical solutions of our difference equations. The slopes of the plotted lines are well described by the apparent order of the algorithms, within 1% even for the largest δ shown in the figure. In Fig. 2 a different comparison is made by considering the accuracies for a given amount of computer time rather than for a given time step. We plot here the maximum trajectory deviation versus the number of force evaluations, as these are usually the most time-consuming operations. The trajectory deviation of the fourth-order Runge–Kutta method is increased by about two orders of magnitude; although worse than Störmer's solution for a large time step, it becomes more efficient already at about 40 force evaluations per period, corresponding to $\delta = 0.16$ for Störmer, $\delta = 0.63$ for Runge–Kutta, and a deviation of less than 0.5% in the first period. The Runge–Kutta method is therefore superior, even in efficiency, to the lower order schemes. For the implicit methods we plot here the solutions iterated only once and twice: the deviation is larger than the exact solution (limit of an infinite number of iterations), and the efficiency is smaller and smaller. The second-order Adams–Moulton method, for example, loses about a factor of n in efficiency if it is iterated n times. Although not true in general, a similar behavior is found for the Newton's equations of motion for a large number of particles interacting with a Lennard–Jones $(r^{-12} - r^{-6})$ potential (see, for example, [7]).

Higher order methods are obviously more accurate but, also, more expensive. The best performance, as suggested by Fig. 2 for our model, is still exhibited by the "Gear" 4 algorithm. The Runge–Kutta method, although less efficient than "Gear" 4, is more efficient than the lower order algorithms in a useful range of values for the time step. Its relative simplicity often makes it preferable to other fourth-order algorithms.

A second interesting aspect of the numerical solutions is their behavior in time (for a fixed value of the time step) and how this is connected with the reversibility of the difference equations. The trajectory deviation in our simple model can be described in terms of a phase shift, which increases quadratically in time and makes the true and the analytic solutions out of phase and in phase again periodically, with a period determined by the time step δ and the order of the algorithm. The amplitude of the solution remains constant for reversible equations, and changes (with damping in the forward direction) for irreversible equations. The amplitude of the Runge–Kutta solution in the forward direction, for example, is decreased by 10% in 241 periods of the oscillator, when $\delta = 0.4$. Of the six schemes that we have considered, only Runge–Kutta is irreversible. The other five are reversible. When the four implicit reversible schemes are used with only a few iterations, however, they produce irreversible, damped solutions. For example, the solution of the Adams–Moulton algorithm iterated only once, x_1, is damped by 10% after fewer than 20 oscillator periods when $\delta = 0.2$.

Acknowledgments

We thank the National Science Foundation for a generous travel grant, S. Bestiale for valuable discussions, and Risa Weis for technical assistance.

References

1. W. E. Milne, *Numerical Solution of Differential Equations* (Dover, New York, 1970).
2. H. J. C. Berendsen and W. F. van Gunsteren, in *Proceedings, Enrico Fermi School on "Molecular-Dynamics Simulation of Statistical-Mechanical Systems"* Varenna, 1985, edited by G. P. F. Ciccotti and W. G. Hoover (North-Holland, Amsterdam, 1986).
3. T. E. Hull, W. H. Enright, B. M. Fellen, and A. E. Sedgwick, *SIAM J. Numer. Anal.* **9**, 603 (1972).
4. W. G. Hoover and W. T. Ashurst, *Theor. Chem.* **1**, 1 (1975).
5. L. Verlet, *Phys. Rev.* **159**, 98 (1967).
6. P. H. Cowell and A. D. C. Crommelin, *Essay on the Return of Halley's Comet* (Greenwich Observations, 1909).
7. A. Rahman, *Phys. Rev. A* **136**, 405 (1964).
8. D. Beeman, *J. Comput. Phys.* **20**, 130 (1976).
9. E. N. Lorenz, *J. Atmos. Sci.* **20**, 130 (1963).
10. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice–Hall, Princeton, NJ, 1971).

Received: February 11, 1986; revised: February 6, 1987

G. D. Venneri
William G. Hoover
Department of Applied Science,
University of California at Davis-Livermore and
Lawrence Livermore National Laboratory,
Livermore, California 94550

Printed by Catherine Press, Ltd., Tempelhof 41, B-8000 Brugge, Belgium