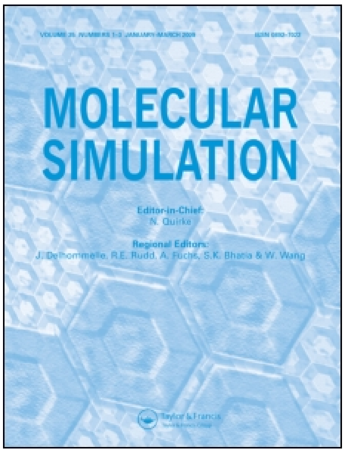

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

On the Accuracy Of Some Common Molecular Dynamics Algorithms

P. Mark Rodger[a][b]
[a] British Gas plc, London Research Station, London [b] Department of Chemistry, The University, Reading

To cite this Article Rodger, P. Mark(1989) 'On the Accuracy Of Some Common Molecular Dynamics Algorithms', Molecular Simulation, 3: 5, 263 — 269
To link to this Article: DOI: 10.1080/08927028908031379
URL: http://dx.doi.org/10.1080/08927028908031379

Molecular Simulation, 1989, Vol. 3, pp. 263–269
Reprints available directly from the publisher
Photocopying permitted by license only

ON THE ACCURACY OF SOME COMMON MOLECULAR DYNAMICS ALGORITHMS

P. MARK RODGER*

British Gas plc, London Research Station, Michael Rd., London SW6 2AD

(*Received November, 1988*)

Attention is drawn to the fact that some of the algorithms used in the simulation of molecular dynamics are less accurate than is commonly believed. In particular, we show that many of the "Verlet-equivalent" integration schemes are not equivalent to the Verlet algorithm, and consequently are not necessarily third order schemes which exhibit exact time-reversal symmetry. Of this class of algorithms, only Beeman's technique is found to generate the optimal positions and velocities for a third order technique. It is also pointed out that the method of constraints introduces errors of $O(\tau^3)$ into the calculated position, and hence limits the accuracy of simulations that employ this method to second order.

KEY WORDS: Molecular Dynamics, algorithms, Verlet, Beeman.

INTRODUCTION

Over the last twenty years a variety of algorithms for the computer simulation of molecular dynamics have been proposed. In particular there is wide support for the use of low order direct finite difference methods such as those based on the Verlet algorithm [1]. During this period there has also been extensive discussion of the relative merits and accuracy of the various techniques, and there appears to be a concensus that algorithms in which the error is of lower than fourth order in the integration time step (*i.e.* lower than third order algorithms) are not efficient methods of performing molecular dynamics (MD) calculations [2,3]. The purpose of this note is to point out that many of the commonly (and successfully) used techniques are in fact only second order algorithms. In the rest of this paper we shall consider two general classes of algorithm: the "Verlet-equivalent" algorithms, and the method of constraints for fixed intramolecular degress of freedom.

VERLET-EQUIVALENT ALGORITHMS

One of the most widely used algorithms for MD is the one introduced by Verlet [1], which uses the position of a particle at two successive time steps to predict the position at the next time step:

$$r_{n+1} = 2r_n - r_{n-1} + \tau^2 a_n, \tag{1}$$

where $r_m (= r(t_m))$ and a_m are the position and acceleration of the particle at time $t_m (= m\tau)$, and τ is the integration time step. The advantages of this method are that it is simple and efficient to program, it produces a configuration-space trajectory that is correct to $O(\tau^4)$, and it exactly satisfies time-reversal symmetry. Its major disadvan-

[*] Present address: Department of Chemistry, The University, Whiteknights, P.O. Box 224, Reading RG6 2AD.

tage, on the other hand, is that it does not treat the velocities explicitly and so, for example, it is not amenable to constant temperature simulations. The exact velocity, v_n, is usually approximated by the relationship

$$u_n = (r_{n+1} - r_{n-1})/(2\tau) = v_n + O(\tau^2) \tag{2}$$

which, being correct only to O(τ), will lead to large fluctuations in the kinetic energy; note, however, that because equation (1) does not depend on the velocity, these errors will not affect the configuration space trajectory that it generates.

As a result of these difficulties a number of other algorithms that supposedly combine the advantages of equation (1) with an explicit and more accurate treatment of the velocity have been proposed; these include the leap-frog scheme [4],

$$\begin{aligned} r_{n+1} &= r_n + \tau z_n \\ z_{n+1} &= z_n + \tau a_{n+1}, \end{aligned} \tag{3}$$

where z_n is defined by

$$z_n = (r_{n+1} - r_n)/\tau = v((n + 1/2)\tau) + O(\tau^2)$$

the velocity-Verlet algorithm [5],

$$\begin{aligned} r_{n+1} &= r_n + \tau u_n + (1/2)\tau^2 a_n \\ u_{n+1} &= u_n + (1/2)\tau(a_{n+1} + a_n) \end{aligned} \tag{4}$$

where u_n is given by equation (2), and the Beeman algorithm [6]

$$\begin{aligned} r_{n+1} &= r_n + \tau v_n + (1/6)\tau^2(4a_n - a_{n-1}) \\ v_{n+1} &= v_n + (1/6)\tau(2a_{n+1} + 5a_n - a_{n-1}). \end{aligned} \tag{5}$$

In addition, eqs (3) and (4) are often used with the approximate velocities, z and u, replaced by the exact velocities [7,8]:

$$\begin{aligned} r_{n+1} &= r_n + \tau v_{n+1/2} \\ v_{n+1/2} &= v_{n-1/2} + \tau a_n \end{aligned} \tag{6}$$

and

$$\begin{aligned} r_{n+1} &= r_n + \tau v_n + (1/2)\tau^2 a_n \\ v_{n+1} &= v_n + (1/2)\tau(a_{n+1} + a_n). \end{aligned} \tag{7}$$

In all of these cases the resultant trajectory satisfies equation (1), and because of this they are usually claimed to be equivalent to the Verlet algorithm and to produce exactly the same configuration space trajectory as equation (1) (see for example [8]).

Unfortunately, reduction of an algorithm to equation (1) is a necessary but not sufficient condition for it to be equivalent with the Verlet algorithm, and hence it does not necessarily follow that any two methods that may be reduced to equation (1) will produce exactly the same trajectory from given initial conditions. That this is so can be seen by considering two sets of three points, r_{n-1}, r_n, and r_{n+1} generated from equation (1), and R_{n-1}, R_n, and R_{n+1} generated from some other algorithm, such that

$$\begin{aligned} R_m &= r_m + mf(t_m), \\ R_{-1} &= r_{-1}. \end{aligned} \tag{8}$$

Then the points R_m will also satisfy equation (1) to within the accuracy of the Verlet algorithm if

$$2\tau \partial f/\partial t|_{t=t_m} = O(\tau^4),$$

which, together with equation (8), implies that

$$f(t) = \tau^3 g(t) + O(\tau^4) \tag{9}$$

where g is an arbitrary function of time. The significance of equations (8) and (9) is that an integration technique can satisfy equation (1) but still produce a trajectory that steadily diverges from the Verlet result by an amount of $0(\tau^3)$, so that not all techniques that can be reduced to equation (1) will have the properties of the Verlet algorithm; the nature of this divergence can be determined by using the given algorithm to express the estimate of the phase point (r_{n+1}, v_{n+1}) as a Taylor series in τ, expanded about the point (r_n, v_n).

A closer look at the "Verlet-equivalent" algorithms listed in equations (3)–(7) reveals that only equations (3) are really equivalent to the Verlet scheme, and being equivalent, involve exactly the same first order approximation to the velocity. In fact, while equations (6) and (7) do produce a more accurate prediction of the velocity (with $O(\tau^3)$ errors), they diverge from the true position by terms of $O(\tau^3)$; hence the improvement in the velocity will be lost in the cumulative effect of a less accurate prediction of the position, so that these two techniques are only correct to second order in τ. Both of the remaining algorithms, the velocity-Verlet and Beeman methods, do give configuration trajectories that are correct to fourth order, but produce different fourth order errors from the Verlet trajectory (the fourth order coefficients in r_{n+1} are $-(1/24)\ddot{a}_n$, 0, and $-(1/12)\ddot{a}_n$ for equations (1), (4) and (5) respectively, which should be compared with $+(1/24)\ddot{a}_n$ for the Taylor series — $\ddot{a}$ denotes $\partial^2 a/\partial t^2$). However equations (4) still give just a first order approximation to the correct velocity, so that Beeman's method is the only one to give the optimal accuracy for a third order technique, *viz.* a trajectory in which errors in r are $O(\tau^4)$, and those in v are $O(\tau^3)$. It is for this reason that MacGowan and Hayes recently found that the Beeman algorithm gave the most satisfactory results when used with large time steps [9].

We note, finally, that the inequivalence the Verlet and "Verlet-equivalent" algorithms means that it is no longer obvious that the latter techniques will exhibit time reversal symmetry; in fact, neither the Beeman algorithm (equation (5)) nor the velocity-Verlet algorithm (equations (4)) are exactly symmetric to time reversal, with the deviations again being $O(\tau^4)$.

CONSTRAINT TECHNIQUES

The use of constraint techniques has become a popular means of treating very high frequency and small amplitude motions such as bond vibrations. Since such motion has almost no effect on many of the properties of the system it is convenient to hold these variables fixed throughout the duration of a simulation, thus allowing one to use a longer time step in the calculations. The method of constraints achieves this by introducing a force acting in the appropriate degree of freedom, and choosing the magnitude of this force at the end of each time step in such a way that the relevant degree of freedom is unchanged at the end of the time step. The magnitude of these constraint forces may then be found either by inverting the resulting matrix of constraint conditions, or by applying some sort of iterative scheme of cycling through the various constraints, as occurs in the SHAKE algorithm [10]. For example, if one or more bond lengths are to be kept fixed then one introduces forces along the bonds whose magnitudes are chosen to ensure that the bond length is unchanged at the end of each time step.

The advantage of this scheme over a direct numerical integration of the constraint forces is that it ensures that the constraints are always satisfied exactly, so that no spurious numerical errors are introduced into the constrained degrees of freedom [11]. What is not generally realized, however, is that this procedure introduces new errors into the resulting trajectory, and that these errors are typically $O(\tau^3)$ (where again τ is the integration time step). The reason for this lies in the assumption that the constraint force acts entirely in the constrained degree of freedom, for example entirely in the direction of the bond. While this is true instantaneously, it is not true of the net force acting over the entire step, and so the coupling of a finite difference scheme with this constraint technique will ensure that the constraint force is applied in the wrong direction.

To illustrate this we consider the motion of a particle constrained to lie on a circle, choosing the Cartesian axes so that initially the particle is on the x axis. Initially, the method of constraints involves solving the equations of motion without the constraint force, which, in the absence of an external potential, may be done exactly and results in the unconstrained position

$$\begin{aligned} x_u(\tau) &= R \\ y_u(\tau) &= V\tau, \end{aligned} \tag{10}$$

where R is the radius of the circle and V is the speed of the particle. This is then followed by the application of a constraining force in the x direction (the direction of the constraint at the start of the time step), the magnitude of which is chosen retrospectively to ensure that $x(\tau)^2 + y(\tau)^2$ is exactly R. This results in the constrained prediction of the position:

$$\begin{aligned} x_c(\tau) &= (R^2 - V^2\tau^2)^{1/2} = R(1 - V^2\tau^2/2) + O(\tau^4) \\ y_c(\tau) &= V\tau. \end{aligned} \tag{11}$$

In reality, however, the constraint force will also influence the y coordinate during the time interval τ, as is clear from the exact solution for circular motion

$$\begin{aligned} x(\tau) &= R\,cos(V\tau/R) = R(1 - V^2\tau^2/2) + O(\tau^4) \\ y(\tau) &= R\,sin(V\tau/R) = V\tau - V^3\tau^3/6 + O(\tau^4). \end{aligned} \tag{12}$$

Note that the third order term in the (exact) Taylor series expansion for $y(\tau)$ reflects the fact that the derivative of the y component of the constraint force is non-vanishing at $t = 0$; this is precisely the term that is neglected by the method of constraints. Comparison of equation (11) and (12) confirms that the constraint technique is only accurate to $O(\tau^2)$.

This result is in fact quite general. Since the implementation of the constraint force completely neglects the components of the derivative of this force in the directions perpendicular to the constraint, it will also always neglect the third order term in the Taylor series expansion for the motion in these degrees of freedom, and so will generate a trajectory that is only accurate to $O(\tau^2)$. In principle it would be possible to correct for this effect by taking the constraint force to act in a slightly different direction from the constraint at time t (in fact, in a direction somewhere in between the directions defined by the constraint at times t and $t + \tau$), but the required direction will vary with the curvature of the constrained degree of freedom, the velocity of the system in all other degrees of freedom, and the size of the time step; it is therefore unlikely that this direction can be specified simply for an arbitrary system. In view of these observations, it is interesting to recall the investigation into the relative merits of various MD algorithms undertaken by van Gunsteren and Berendsen [12]. These authors compared the Verlet and predictor-corrector algorithms both in the presence and the absence of Cartesian constraints, and found that the use of constraints reversed their findings: in the absence of constraints the high order predictor-corrector algorithms were better, whereas in the presence of constraints it was the lower order Verlet algorithm that was preferred. From the above discussion it is apparent that the magnitude of the errors introduced by the method of constraints makes the use of a high order algorithm redundant, and so it is not surprising that it is preferable to use a lower order algorithm in conjunction with the method of constraints.

It should be noted that the constraint interactions are purely intramolecular forces and so can change only the orientation and internal configuration of the molecule, leaving its centre of mass unaltered. As a result, it is likely that many of the bulk properties of the simulated system, such as energy and pressure, will be relatively unaffected by the numerical errors introduced by the method of constraints; this is because such properties tend to be dominated by intermolecular interactions, which are usually insensitive to small changes in the internal configuration of the molecule. Consequently, since tests of the accuracy of MD programs rely on properties such as energy conservation, it may be difficult to ensure that the integration step is small enough to give an accurate description of the intramolecular degrees of freedom. This will be compounded by the fact that the use of rigid constraints also modifies the unconstrained intramolecular degrees of freedom [13,14], and although it is known how to describe this effect, it is impractical to do so even for moderately sized systems. Hence it is likely that a realistic assessment of the accuracy of simulated intramolecular properties of rigidly constrained molecules is possible only for small molecules (say up to about four or five constituent particles).

A number of computer simulations of small molecules have been reported, and these indicate that there is good agreement between the simulated and exact intramolecular properties; however these studies have used either curvillinear coordinates (instead of the rectilinear coordinates employed by the method of constraints) [15,16], or implemented the method of constraints in stochastic simulations [17,18]; since the latter method makes use of a random force on the molecule it is likely to mask the effect of small errors in the prescription of the systematic force. We have therefore tried to repeat some of these simulations by using Cartesian constraints in a MD simulation. The particular system considered was a Kramers freely jointed three-bead chain, which has two rigid bonds but free movement of the bond angle. It was found that with a time step that gave good energy conservation (fluctuations of about 0.01% of kT) the equilibrium distribution of the bond angle was very different from the correct distribution [13,15]; we note that other workers have found similar difficulties [19]. Although reducing the time step did substantially improve the calculated distribution, we found that the duration of the trajectory required to get a representative sample of the angular distribution made it impractical to find the limiting distribution. This suggests that there may be at least some internal degrees of freedom for which the method of constraints is not an efficient computational probe.

CONCLUSIONS

In conclusion, we note that many of the integration schemes commonly employed in molecular dynamics simulations are in fact only accurate to second order in the integration time step. Of the class of Verlet-equivalent algorithms, only the Verlet (equation (1)), velocity-Verlet (equation (4)) and Beeman (equation (5)) algorithms generate a configurational trajectory that is correct to $O(\tau^3)$, where τ is the integration time step, and of these, only Beeman's method gives a velocity that is correct to $0(\tau^2)$. Use of the Beeman algorithm is therefore recommended in studies of velocity-dependent properties.

Further, the accuracy of the integration scheme will probably be limited by the accuracy of the scheme used to implement holonomic constraints such as fixed bondlengths, since the method of constraints (which is presently the only practical method of implementing constraints in large molecules) is inherently a second order scheme. It is probable that many bulk properties will be fairly insensitive to the errors introduced by the method of constraints, but these errors are likely to affect intramolecular properties such as the equilibrium distribution of dihedral angles in n-alkanes and polymer chains.

Acknowledgements

The author wishes to thank British Gas plc for supporting this work.

References

[1] L. Verlet, "computer 'experiments' on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules", *Phys. Rev.*, **159**, 98 (1967).

[2] H.J.C. Berendsen and W.F. van Gunsteren, "Practical algorithms for dynamic simulations", in Molecular dynamics simulation of statistical mechanical systems, G. Ciccotti and W.G. Hoover, eds, Proceedings of the International School of Physics "Enrico Fermi", course 97, North-Holland, Amsterdam, 1986.

[3] W.F. van Gunsteren and H.J.C. Berendsen, "Algorithms for Brownian dynamics", *Molec. Phys.*, **45**, 637 (1982).

[4] R.W. Hockney, "The potential calculation and some applications", *Methods comput. Phys.*, **9**, 136 (1970).

[5] 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).

[6] D. Beeman, "Some multistep methods for use in molecular dynamics calculations", *J. comput. Phys.*, **20**, 130 (1976).

[7] M.P. Allen and D.J. Tildesley, Computer simulation of liquids, Clarendon Press, Oxford, 1987.

[8] W.F. van Gunsteren and H.J.C. Berendsen, "A leap-frog algorithm for stochastic dynamics", *Molec. Sim.*, **1**, 173 (1988).

[9] D. MacGowan and D.M. Hayes, "Large time steps in molecular dynamics simulations", *Molec. Sim.*, **1**, 277 (1988).

[10] 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).

[11] J.-P. Ryckaert, "Special geometrical constraints in the molecular dynamics of chain molecules", *Molec. Phys.*, **55**, 549 (1985).

[12] W.F. van Gunsteren and H.J.C. Berendsen, "Algorithms for macromolecular dynamics and constraint dynamics", *Molec. Phys.*, **34**, 1311 (1977).

[13] M. Fixman, "Classical statistical mechanics of constraints: a theorem and application to polymers", *Proc. Nat. Acad. Sci. USA*, **71**, 3050 (1974).

[14] P.M. Rodger and M.G. Sceats, "Phase space considerations for interacting particles obeying the Fokker-Planck equation", *J. Chem. Phys.*, **83**, 3358 (1985).

[15] M. Gottlieb and R.B. Bird, "A molecular dynamics calculation to confirm the incorrectness of the random-walk distribution for describing the Kramers freely jointed bead-rod chain", *J. Chem. Phys.*, **65**, 2467 (1976).

[16] M.R. Pear and J.H. Weiner, "Brownian dynamics study of a polymer chain of linked rigid bodies", *J. Chem. Phys.*, **71**, 212 (1979).

[17] W.F. van Gunsteren, "Constrained dynamics of flexible molecules", *Molec. Phys.*, **40**, 1015 (1980).

[18] J.A. Montgomery, S.L. Holmgren and D. Chandler, "Stochastic molecular dynamics study of the trans-gauche isomerization process in simple chain molecules", *J. Chem. Phys.*, **73**, 3688 (1980).

[19] D.J. Tildesley, private communications.