JOURNAL OF COMPUTATIONAL PHYSICS **47**, 444–451 (1982)

A New Algorithm for Molecular Dynamics Calculations

SØREN TOXVAERD

Institute for Chemistry, University of Copenhagen,
DK-2200 Copenhagen N, Denmark

Received January 6, 1982

The Newtonian differential equations for N particles can be solved numerically by the molecular dynamics technique using different algorithms. An algorithm which makes use of the derivatives of the forces $a(t)$ at the time when the new positions are calculated is found to be superior to the algorithms which extrapolate using time behaviour of the forces in the previous time period.

1. INTRODUCTION

In *molecular dynamics* (MD) calculations, the Newtonian equations of motion for N particles are solved numerically [1–3]. Simulation of macroscopic systems requires that N and the integration time Δt be large; however, the technical limit at present is $N \lesssim 10{,}000$ and $\Delta t \lesssim 10^{-8}$ sec. The upper limit of Δt depends on how accurately one needs to calculate the trajectories; e.g., the thermodynamics of bulk systems far from phase transitions depends only on the lower-order particle distribution functions and is rather insensitive to the precise structure. Therefore, a large integration step h can be applied in the MD calculations. Near a phase transition, this is no longer the case.

A number of MD algorithms have been proposed. The simplest and most commonly used algorithm is a simple third-order predictor (Verlet algorithm [3])

$$\mathbf{r}_i(t+h) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-h) + h^2\mathbf{a}_i(\mathbf{r}_i(t)) + O(h^4), \tag{1}$$

by which the position of the ith particle at time $t+h$ is obtained from its position at t and $t-h$ and the acceleration $\mathbf{a}_i$ which is calculated from the potential energy, $U(\mathbf{r}^N)$, of the system at time t

$$\mathbf{a}_i = -\nabla_i U(\mathbf{r}^N). \tag{2}$$

In general, the predicted positions can be corrected, but it might require a calculation of the accelerations at the predicted positions and since most of the time in an MD calculation is spent evaluating $\mathbf{a}$, one is usually content with a predictor provided it conserves the total energy of the system. The conservation of energy rather than other criteria is used as a test for how large h can be chosen and how simple an algorithm one can apply. Another criterion is the time reversibility of the trajectories. This test,

0021-9991/82/090444–08$02.00/0

however, can only be applied in connection with the criterion of conservation of energy, which can clearly be seen by considering the simple predictor algorithm (1) which is symmetric in time and thus time reversible [4] even for a time increment h for which the energy is no longer conserved.

The MD calculation technique depends on the nature of the potential energy $U(\mathbf{r}^N)$ which is often approximated by a sum of pair-potentials

$$U(\mathbf{r}^N) = \sum_{i<j}^{N} u(\mathbf{r}_i, \mathbf{r}_j). \tag{3}$$

A typical intermolecular pair-potential is strongly repulsive at short distances r_{ij} with a weak long-range attraction. This results in rapid variation in time for the short-range forces [5] and makes the differential equations stiff [6], with the consequence that one is forced to choose a small time increment in order not to predict unrealistic collision positions. Even for small h, the simple algorithm (1) predicts collisions that are too energetic, but these terms are compensated by corresponding attractive and kinetic terms, so that the total energy $E = K + U$ is conserved over many time steps, although it fluctuates. For larger h, the fluctuations are drastically increased and the mean energy $\langle E \rangle = \langle K \rangle + \langle U \rangle$ is no longer conserved. Thus, the criterion of constant energy is not a sufficient test for the accuracy of an MD calculation scheme, since a conserved mean energy $\langle E \rangle$ can be obtained from correlated fluctuations in its two components which cancel each other. On the other hand, this circumstance gives an additional test for the MD calculations, which should provide correct fluctuations in K (and in the pressure), or what is equivalent, correct derivatives of the thermodynamic functions.

Near a phase change, e.g., melting, the high energy collisions might be the trigger of the phase transition by producing (artificial) fluctuations necessary to reach the new state. It is therefore especially important to apply an accurate MD algorithm at phase transitions, and this is the motivation for the present article, in which we derive a much more accurate MD predictor. Recently, a melting theory for two-dimensional (2-D) systems has been formulated. The melting is driven by dislocations and disclimations which are easy to identify in a MD simulation and thus give us an additional test, namely, that the density of crystal defects near melting should not depend on the actual algorithm used in the MD calculation.

2. MD ALGORITHMS

For simplicity, we consider a system of N Lennard–Jones (LJ) particles (see the Appendix). The second-order differential equations can be written as

$$\mathbf{r}_i''(t) = \mathbf{a}_i(t) = -\nabla_i \left(\sum_{i<j}^{N} u_{\mathrm{LJ}}(r_{ij}) \right), \tag{4}$$

where $\mathbf{a}_i$ is given in units of the mass. If the positions and the higher order derivatives are known at time t, the positions after a time increment h can be evaluated from a Taylor expansion. In even powers of h, which is convenient due to relation (4), we obtain

$$\mathbf{r}_i(t+h) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-h) + h^2\mathbf{a}_i(t) + \tfrac{1}{12}h^4\mathbf{a}_i''(t) + O(h^6). \tag{5}$$

The velocities $\mathbf{r}_i'(t)$ to third order in h are

$$\mathbf{r}_i'(t) = (\mathbf{r}_i(t+h) - \mathbf{r}_i(t-h))/2h - \tfrac{1}{6}h^2\mathbf{r}_i'''(t) + O(h^4). \tag{6}$$

These equations can be rewritten by expressing $\mathbf{r}_i''(t)$ (and $\mathbf{r}_i'''(t)$) as a sum of accelerations at previous time steps. To the same order in h as (5),

$$\begin{aligned}\mathbf{r}_i(t+h) = 2\mathbf{r}_i(t) - \mathbf{r}_i(t-h) + \frac{h^2}{12}\,&[14\mathbf{a}_i(t) - 5\mathbf{a}_i(t-h)\\ &+ 4\mathbf{a}_i(t-2h) - \mathbf{a}_i(t-3h)] + O(h^6).\end{aligned} \tag{7}$$

Predictors (5) and (7) can be followed up by correctors, e.g., (7) is the predictor in a fifth-order Nordsieck–Gear method, which uses a fifth-order Adam–Moulton corrector [6, 7]. In [7], the predictor (7) is followed by a fifth-order corrector

$$\begin{aligned}\mathbf{r}_{i,\mathrm{corr}}(t+h) = \mathbf{r}_i(t) + h\mathbf{r}_i'(t) + (h^2/360)[&38\mathbf{a}_i(t+h) + 171\mathbf{a}_i(t)\\ &- 36\mathbf{a}_i(t-h) + 7\mathbf{a}_i(t-2h)] + O(h^6),\end{aligned} \tag{8}$$

and the velocities are calculated to fourth order as

$$\begin{aligned}\mathbf{r}_i'(t+h) = (\mathbf{r}_i(t+h) - \mathbf{r}_i(t))/h + (h/360)[&97\mathbf{a}_i(t+h)\\ &+ 114\mathbf{a}_i(t) - 39\mathbf{a}_i(t-h) + 8\mathbf{a}_i(t-2h] + O(h^5).\end{aligned} \tag{9}$$

Equations (5) and (7) are in principle different, and this makes it possible to calculate the trajectories in two ways: one can either extrapolate to the new positions using the derivatives of $\mathbf{a}_i(t)$ at time t, or alternatively, one can extrapolate using the history of the trajectories. Traditionally, one makes use of the latter possibility because of its simplicity, since one only needs to store three sets of accelerations during the MD calculations. The evaluation of $\mathbf{a}_i''(t)$ in (5) is more complicated. Differentiation of (4) with respect to time gives [5]

$$\begin{aligned}\mathbf{a}_i''(t) = \sum_{j\neq i}^{N}\, \{&[B_{ij}(\mathbf{r}_{ij}\cdot\mathbf{r}_{ij}'' + \mathbf{r}_{ij}'^{2}) + C_{ij}(\mathbf{r}_{ij}\cdot\mathbf{r}_{ij}')^2]\mathbf{r}_{ij}\\ &+ 2B_{ij}(\mathbf{r}_{ij}\cdot\mathbf{r}_{ij}')\mathbf{r}_{ij}' + A_{ij}\mathbf{r}_{ij}''\}\end{aligned} \tag{10}$$

and

$$\mathbf{r}_{ij}''' = \sum_{j\neq i}^{N}\, \{A_{ij}\mathbf{r}_{ij}' + B_{ij}(\mathbf{r}_{ij}\cdot\mathbf{r}_{ij}')\mathbf{r}_{ij}\}, \tag{11}$$

where

$$A_{ij} = -\frac{1}{r_{ij}} \frac{du(r_{ij})}{dr_{ij}}, \qquad B_{ij} = \frac{1}{r_{ij}} \frac{dA_{ij}}{dr_{ij}}, \qquad C_{ij} = \frac{1}{r_{ij}} \frac{dB_{ij}}{dr_{ij}}, \tag{12}$$

and all the terms in (10)–(12) have to be taken at time t.

It is, however, not possible to apply Eqs. (5) and (10)–(12) directly in a standard MD program, where contributions to the acceleration of the ith particle at time t are calculated from contribution $\sum_j u'(r_{ij}(t))$. At the time when these contributions are calculated, the total acceleration $\mathbf{a}_i(t)$, which is needed for evaluating $\mathbf{a}_i''(t)$ in Eq. (10), is, of course, not known. Apparently one needs to perform an extra time-consuming double summation over the N particles in order to obtain $\mathbf{a}_i''(t)$, and one could ask if it would not be simpler and more accurate to use the Verlet algorithm (1) and the half time increment instead of including an extra term in the expansion. This is, however, not the case. Halving the time increment increasing the (energy) accuracy by a factor $\lesssim 10$ and doubles the computer time, whereas the inclusion of an extra term in the algorithm increases the accuracy by a factor of $10-70$ and increases the computer time by only ~20% (depending on N, $u(r)$, and the computer). This is because we only need to calculate $\mathbf{a}_i''(t)$ for small particle distances r_{ij} due to the nature of the potential function and the geometry of the system which make differential equations (4) stiff. The many contributions to $\mathbf{a}_i''(t)$ from larger distances

TABLE I

Sixty Time Steps with $h = 0.005$

r_a/r_m	$\langle E/N\varepsilon\rangle$	$\langle\lvert\Delta E/N\varepsilon\rvert\rangle \times 10^5$	c_V
0	-1.414346 ± 0.000224	5.154	1.89471
1.1	-1.414322 ± 0.000004	0.203	1.91373
1.5	-1.414339 ± 0.000003	0.159	1.91382
2.0	-1.414365 ± 0.000003	0.157	1.91380
2.5	-1.414367 ± 0.000003	0.155	1.91380

TABLE II

Six Hundred Steps with $h = 0.005$

Algorithm	kT/ε	pr_m^2/ε	$\langle\lvert\Delta E/N\varepsilon\rvert\rangle \times 10^5$	$E'/N\varepsilon$	c_V	N_d
Verlet (1)	1.013	6.634	3.655	≈ 0	1.54	54
Eq. (5)	1.014	6.623	0.125	≈ 0	1.65	38
Predictor (7)	1.013	6.614	1.198	-8.4×10^{-4}	1.63	40
Predictor-Corrector (7)–(9)	1.015	6.620	0.648	5.8×10^{-4}	1.72	44

TABLE III

Two Hundred Steps with $h = 0.015$

Algorithm	kT/ε	pr_m^2/ε	$\langle\lvert\Delta E/N\varepsilon\rvert\rangle \times 10^4$	$E'/N\varepsilon$	c_V
Verlet (1)	1.007	6.699	13.6	≈ 0	1.54
Eq. (5)	1.016	6.618	1.51	4.3×10^{-4}	1.74
Predictor (7)	0.914	5.810	22.2	−0.11	(−3.60)
Predictor–Corrector (7)–(9)	1.161	7.682	36.1	+0.19	(−0.91)

are small and cancel each other. This property has been tested for three different systems: a two-dimensional (2-D) dense LJ fluid, the three-dimensional LJ system analysed in [7] and a 2-D LJ solid near its melting point [8]. For all three systems, we find that we only need to include contributions to $\mathbf{a}_i''(t)$ from the short-range interactions. The results for the 2-D LJ solid, which is the most difficult system to simulate of the three mentioned, are given in Tables I–III; the calculation details, units, etc. are given in the Appendix.

In Table I, we present the result of MD calculations of 60 time steps with $h = 0.005$ and for different values of r_a, the maximum particle distance for which contributions to the fourth-order term $\mathbf{a}_i''(t)$ are taken into account. The short time interval ensures that the trajectories are essentially the same for the five calculations. The energy E varies from step to step; the table gives the mean energy per particle $\langle E/N\varepsilon\rangle$ and the root mean square (r.m.s.) deviation. The mean amplitude of energy $\langle\lvert\Delta E/N\varepsilon\rvert\rangle$ is calculated as the mean of energy variations $\lvert E_i - E_{i-1}\rvert$ between successive time steps. The heat capacity c_V is obtained from the temperature fluctuations [9]. Sixty time steps are not nearly enough to determine c_V, but the variations in the instant value of c_V show that the temperature fluctuations are also sensitive to the fourth-order term. For $r_a = 0$, algorithm (5) reduces to the usual third-order Verlet algorithm (1). Inclusion of the contributions to $\mathbf{a}_i''(t)$ from the short-range interactions increases the accuracy of the algorithm drastically, whereas the contributions from long-range interactions cancel out. The same behaviour is observed for the two other systems and for other time increments.

The result of a comparison of algorithm (5) with other algorithms is presented in Table II. The table shows the result of MD calculations for the reduced time $\Delta t = 3$ performed as 600 time steps with $h = 0.005$. In this relatively short time interval, the systems will still be in the same subvolume of the phase space so that the differences in the observed quantities are caused by the algorithms. The mean amplitude of energy $\langle\lvert\Delta E/N\varepsilon\rvert\rangle$ gives the short-time behaviour of the energy. The conservation of energy over the total time interval $\Delta t = 3$ is measured by performing a linear least-squares fit to $E(t)$. The slope, $E'/N\varepsilon$, gives the drift in energy, and for predictor (7) and Eqs. (7)–(9), there is a drift in the total energy. The temperature and pressure are not sensitive to the choice of algorithm, but the variations in the instant values of the heat capacity c_V indicate that the usual MD technique might not be accurate enough near a phase transition. The number N_d of crystal defects supports this conclusion. The defects are defined as particles with a number of nearest neighbours different from six [10] (for the 2-D triangular lattice) and the particle configurations after each 60 steps were analysed for defects. The number of defects, using different algorithms, began to deviate after the first 240 steps, and N_d in Table II is the total number of defects in the last 6 subsets. The conclusion from the data in Table II and corresponding data for fluid systems in 2-D and 3-D is that algorithm (5) is much more accurate than either the Verlet algorithm (1) or algorithms (7)–(9) based on the difference method. The new algorithm keeps the energy more constant, and near a phase transition it leads to fewer extreme configurations defined as crystal defects.

In Table III, we have shown the corresponding result of MD calculations again for $\Delta t = 3$, but now performed as 200 steps with $h = 0.015$ and using the same starting positions as for the data of Table II. Predictor (7) and corrector (7)–(9) break down for this time increment. The energy is not conserved and the fluctuations are completely wrong, leading to unphysical heat capacities. The Verlet algorithm begins to predict wrong thermodynamic data (T, p). For 120 time steps with $h = 0.025$ the Verlet algorithm also breaks down, whereas algorithm (1) gives $kT/\varepsilon = 1.011$ and $pr_m^2/\varepsilon = 6.537$, which only deviates from the values in Table II by $\approx 0.1\%$.

The simple algorithm (1) is symmetric with respect to time and it is presumably this quality which ensures the stability of the algorithm even when the energy fluctuates. Thus, for $h = 0.015$, it was not possible to detect any energy drift using the Verlet algorithm. The higher-order algorithm (5), and its corresponding formula (6) for the velocity is only formally symmetric, because the velocities $\mathbf{r}_i'(t)$ in (10) are calculated by the difference method (A2). This shortcoming causes a (very small) drift in the energy for large h at high densities, which can be overcome by predicting the velocities by (A2) and correcting the velocities by (6). There is a cost in computer time, however, since one needs to calculate $\mathbf{a}_i''(t)$ twice. For $h = 0.005$ and for the two (less dense) fluid systems, we did not obtain any drift in the energy using (5).

3. CONCLUSION

The algorithm of Eq. (5) gives a possibility for calculating the trajectories much more accurately. The success of including the fourth-order term $\mathbf{a}_i''(t)$ in the algorithm is due to the strong repulsive behaviour of the pair-potential at short interaction distances, which make the differential equation stiff. Normally this behaviour complicates the calculations, since one needs to choose a correspondingly small time step when using an extrapolation algorithm. One can overcome this problem, however, by calculating the term $\frac{1}{12}h^4\mathbf{a}_i''(t)$ directly. The stiffness of the differential equations reflects itself in the time behaviour of the forces. The short-range forces vary rapidly with time whereas the long-range forces vary slowly. This property was used by Street, Tildesley, and Saville [5], who extrapolated the long-range forces by expanding $\mathbf{a}_i(t)$. In fact, one could argue that the present article only serves to fully extend their discovery by including the fourth-order term for the short interaction distances directly in the algorithm. The two methods can be used simultaneously and supplement each other.

As pointed out in the Introduction, one might not need to calculated the trajectories to a higher accuracy than, e.g., that given by the Verlet algorithm; it depends on the system and the property considered. The more accurate calculation scheme is suitable at phase transitions and the related interfacial systems [11] and for long-term correlated properties [12, 13].

APPENDIX

The Lennard–Jones potential is

$$u_{\mathrm{LJ}}(r) = \varepsilon[(r/r_m)^{-12} - 2(r/r_m)^{-6}], \tag{A1}$$

and the truncated LJ potential, $u_{\mathrm{LJ}}(r) - u_{\mathrm{LJ}}(r_c)$, used in the MD computations was set equal to zero at $r > 3r_m$ ($3.37\,\sigma$). The calculations were performed for $N = 256$ particles in a box with periodical boundaries. The data given in Tables I–III are for the solid state at the density $\rho r_m^2/\varepsilon = 1.1188$ near the melting point. [8] The length is in units of r_m and the time is in units of $(mr_m^2/\varepsilon)^{1/2}$, where m is the mass of the particles. The energy of the system $E/N\varepsilon$, the temperature kT/ε, the ressure pr_m^2/ε, and the heat capacity c_V are calculated as time averages [8]. The calculations were performed in (Univac) double precision (18 digits) since single precision, in which the real numbers are only given by nine digits, led to round-off errors.

The MD program used in the calculations employs a list of nearest neighbours [3]. The calculations of the terms $\mathbf{a}_i''(t)$ were performed by storing $x_{ij}(t)$, $y_{ij}(t)$, and $r_{ij}^2(t)$ for $r_{ij} < 1.1\,r_m$ during the traditional calculation of the accelerations. Before $\mathbf{a}_i''(t)$ and $\mathbf{r}_i'''(t)$ can be calculated, we need to know not only $\mathbf{a}_i(t)$, but also the velocities $\mathbf{r}_i'(t)$. These (predicted) velocities in (10) and (11) were calculated using the third-order formula [7]

$$h\mathbf{r}_i'(t) = \mathbf{r}_i(t) - \mathbf{r}_i(t-h) + \tfrac{1}{6}h^2[2\mathbf{a}_i(t) + \mathbf{a}_i(t-h)] + O(h^4). \tag{A2}$$

The strategy in which the bond lengths are stored might not be the best. The best computer strategy depends on the actual computer, the energy function, and N. The storage of bond lengths (for $j > i$) requires a store of the order $\frac{1}{2}n_D ND$, where n_D is the number of nearest neighbours and D is the dimension. In MD programs with large numbers of particles, the linked-list technique is more efficient [14]. Calculation of $\mathbf{a}_i''(t)$ can then be performed by considering only particles in the narest cell.

ACKNOWLEDGMENT

A grant for computer time by the Danish Natural Science Research Council is gratefully acknowledged.

REFERENCES

1. B. J. ALDER AND T. E. WAINWRIGHT, *J. Chem. Phys.* **31** (1959), 459.
2. A. RAHMAN, *Phys. Rev.* **136** (1964), A405.
3. L. VERLET, *Phys. Rev.* **159** (1967), 98.
4. Since $\mathbf{r}_i(t-h)=2\mathbf{r}_i(t)-\mathbf{r}_i(t+h)+h^2\mathbf{a}_i(t)$. An irreversibility, due to round-off error, is normally negligible and can be diminished by reserving more bits and/or using integer arithmetic.
5. W. B. STREET, D. J. TILDESLEY, AND G. SAVILLE, *Mol. Phys.* **35** (1978), 639.
6. G. W. GEAR, "Numerical Initial Value Problems in Ordinary Differential Equations," Chap. 11, Prentice-Hall, Englewood Cliffs, N. J., 1971; W. L. MIRANKER, "Numerical Methods for Stiff Equations," Vol. 5, Reidel, Dordrecht, 1981.
7. D. BEEMAN, *J. Compute Phys.* **20** (1976), 130.
8. S. TOXVAERD, *Phys. Rev. A* **24** (1981), 2735.
9. J. L. LEBOWITZ, J. K. PERCUS, AND L. VERLET, *Phys. Rev.* **153** (1967), 250.
10. J. P. MCTAGUE, D. FRENKEL, AND M. P. ALLEN, *in* "Ordering in Two Dimensions" (S. K. Sinha, Ed.), p. 147, North-Holland, New York, 1980.
11. S. TOXVAERD, *J. Chem. Phys.* **74** (1981), 1998.
12. D. LEVESQUE AND W. T. ASHURST, *Phys. Rev. Lett.* **33** (1974), 277.
13. S. TOXVAERD, *Phys. Rev. Lett.* **43** (1979), 529.
14. R. W. HOCKNEY, S. P. GOEL, AND J. W. EASTWOOD, *J. Compute Phys.* **14** (1979), 148.