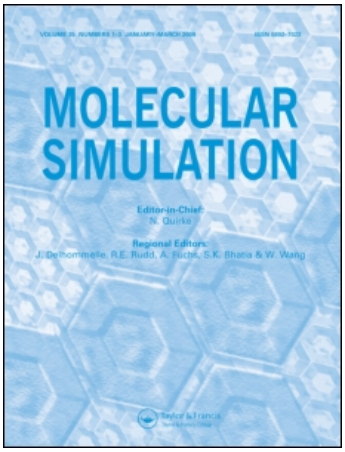

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Integer Interparticle Distances in Molecular Dynamics Simulation

Witold Dzwinel[a]; Monika Bargieł[a]; Jacek Kitowski[a]; Jacek Mościński[a]
[a] Institute of Computer Science, AGH, Cracow, Poland

To cite this Article Dzwinel, Witold , Bargieł, Monika , Kitowski, Jacek and Mościński, Jacek(1991) 'Integer Interparticle Distances in Molecular Dynamics Simulation', Molecular Simulation, 5: 6, 383 — 390
To link to this Article: DOI: 10.1080/08927029108022423
URL: http://dx.doi.org/10.1080/08927029108022423

Molecular Simulation, 1991, Vol. 5, pp. 383–390
Reprints available directly from the publisher
Photocopying permitted by license only

INTEGER INTERPARTICLE DISTANCES IN MOLECULAR DYNAMICS SIMULATION

WITOLD DZWINEL, MONIKA BARGIEŁ, JACEK KITOWSKI and JACEK MOŚCIŃSKI

Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Cracow, Poland

(*Received January 1990, accepted April 1990*)

The paper describes a sequential MD algorithm in which distances between particles are evaluated using fixed point arithmetics. Errors introduced by the method are estimated. Some simulation timings as well as fluctuations of the total energy are compared to the results obtained using floating point arithmetics.

KEY WORDS: Molecular dynamics, Lennard-Jones potentials, computational algorithms, fixed point arithmetics.

1. INTRODUCTION

Interparticle distance evaluation is one of the most time consuming parts of every Molecular Dynamics (MD) or Monte Carlo (MC) program. To avoid $o(N^2)$ complexity of the particle–particle method in simulation with Lennard-Jones (L-J) potentials, various sequential and vector algorithms are applied [1], using for example:

- lists of near neighbours (e.g. [2, 4, 5, 6]),
- the cell methods (e.g. [2, 3, 7, 8, 9, 10]),
- algorithms based on particle coordinates sorting (e.g. [11, 12]),

Efficient parallel algorithms for MD simulation of liquids are still in development. Some examples are published in e.g. [1, 13].

In order to perform MD simulations with 10^3–10^4 particles in a PC/workstation environment an efficient sequential code should be constructed. The *link-cell* method [8, 9, 10] seems to be attractive with respect to its $o(N)$ complexity. After some modifications [14] it can be effectively used in the PC/workstation environment also. One of the modifications is the *cube technique* concept [14]. In this approach the squared distance between any two particles i and j is not evaluated unless differences in particles coordinates fulfil $|r_{\alpha i} - r_{\alpha j}| < R_{\text{cut}}$ where $r_{\alpha i}$ and $r_{\alpha j}$ are the i-th and j-th particles coordinates for $\alpha \in \{x, y, z\}$, $i,j = 1, \ldots, N$, $i \neq j$ and R_{cut} means the potential cutoff radius. The C-language program [15] based on the *cube technique* with potential tables is written in a convention similar to OLYMPUS [2].

In the above method the overall computational time is reduced by limiting the number of interparticle distances computations. It is known however, that integer arithmetics is usually faster than the floating one although the speed-up depends on computer architecture applied. The integer calculations of distances introduce additional errors into the MD simulation. Nevertheless there are some classes of

molecular simulation in which high precision of calculations would not be strongly required. One example is microscale hydrodynamics. It can be simulated with two complementary methods: using MD algorithms [16] with well defined interparticle forces and precise solution of the Newtonian equations of motion and, on the other hand, applying the cellular automaton [17, 18], being a lattice gas. In the second case there is no mutual interactions between particles, besides particle scattering when two particles meet head-on. In such a class of molecular simulation a MD method in which distances between particles are evaluated using integers could be applied, resulting in significant reduction of the computational time demands.

In Section 2 basic principles and errors of the method with integer distances calculations between particles are discussed. Simulation results and main advantages of the algorithm are presented in Section 3.

2. INTEGER MD APPROACH

2.1 Basic Principles

The serious problem of the "fixed point programming" in physics and chemistry seems to be rescaling of various quantities which differ in orders of magnitude. Thus round-off errors for various parts of the program may be different and the total error of simulation is difficult to estimate.

We apply the fixed point arithmetics in the MD code for estimation of distances between particles only, while performing the rest of calculations in the floating point arithmetics and using the tabularized interparticle potential. Higher performance is obtained as well as strictly localized round-off error can be estimated (for details see Section 2.2).

Assume, that the maximum squared distance computed during simulation, $d^2_{\mathrm{MAX}} = \max_{i,j} d^2_{ij}$, is equal to MAXINT, where MAXINT is the maximal integer number (e.g. for the 'unsigned' type of four byte integer representation MAXINT $= 2^{32} - 1$). For the cube technique concept [14] applied to MD forces evaluations

$$d^2_{\mathrm{MAX}} = 3R^2_{\mathrm{cut}} = \mathrm{MAXINT}, \tag{1}$$

so that

$$R_{\mathrm{cut}} = \lfloor \sqrt{\mathrm{MAXINT}/3} \rfloor, \tag{2}$$

$\lfloor \cdot \rfloor$ means the greatest integer function.
Then the program units are chosen on the basis of Equation (2).

Assuming that particle space coordinates are integers, the widely known leapfrog scheme is as follows

$$p^{n+1/2}_{\alpha i} = p^{n-1/2}_{\alpha i} + F^n_{\alpha i}\Delta t, \tag{3}$$

$$r^{n+1}_{\alpha i} = r^n_{\alpha i} + [p^{n+1/2}_{\alpha i}\Delta t/m], \tag{4}$$

$$p^n_{\alpha i} = \tfrac{1}{2}(p^{n+1/2}_{\alpha i} + p^{n-1/2}_{\alpha i}). \tag{5}$$

where
$p^n_{\alpha i}$ – α coordinate of the i-th particle momentum,
$F^n_{\alpha i}$ – α coordinate of the force acting on i-th particle,
Δt – timestep,

m – mass of a particle,
$[\cdot]$ – nearest integer function.
The proposed algorithm consists of Equations (3), (4), (5) and properly chosen scaling factors based on Equations (1), (2).

2.2 Errors Introduced By The Method

The round-off error is generated while the Newtonian equations of motion are solved. The particles move along the constant energy trajectories in the phase space. After the coordinates round-off every particle is shifted in the phase space, changing the constant energy trajectory. Assuming that this process is approximately random, it should generate the additional total energy fluctuations which are random too.

The coordinates round-off introduces the random error $\frac{1}{2}\xi^n_{\alpha i}$, uniformly distributed over $[-\frac{1}{2}, \frac{1}{2}]$. Thus Equation (4) has a form

$$r^{n+1}_{\alpha i} = r^n_{\alpha i} + p^{n+1/2}_{\alpha i}\Delta t/m + \tfrac{1}{2}\xi^{n+1}_{\alpha i} \tag{6}$$

Introducing

$$p^{n+1/2}_{\alpha i} \leftarrow p^{n+1/2}_{\alpha i} + \xi^{n+1}_{\alpha i} m/(2\Delta t), \tag{7}$$

$$R^n_{\alpha i} = m(A^{n+1}_{\alpha i} - A^n_{\alpha i}), \tag{8}$$

$$A^n_{\alpha i} = \xi^n_{\alpha i}/(2\Delta t^2), \tag{9}$$

the set of Equations (3), (5), (6) is written

$$p^{n+1/2}_{\alpha i} = p^{n-1/2}_{\alpha i} + (F^n_{\alpha i} + R^n_{\alpha i})\Delta t, \tag{10}$$

$$r^{n+1}_{\alpha i} = r^n_{\alpha i} + p^{n+1/2}_{\alpha i}\Delta t/m, \tag{11}$$

$$p^n_{\alpha i} = \tfrac{1}{2}(p^{n+1/2}_{\alpha i} + p^{n-1/2}_{\alpha i}). \tag{12}$$

The last set of equations ((10)–(12)) shows that the MD simulation using integer arithmetics for distances calculation is equivalent to the MD simulation of particles interacting each other with the sum of Lennard-Jones force $\mathbf{F}^n_i$ and numerical random force $\mathbf{R}^n_i$.

All techniques used to economise the forces calculations, i.e. the cutoff radius, force interpolation from precalculated tables, etc., introduce force inaccuracies, which can be modelled by adding a random force to the exact one. Amini, Eastwood and Hockney [19] show, that for the pair of particles acting on each other through the simple harmonic oscillator law in the presence of a random force, δF^n, in one dimensional space:

$$\langle E^n_T \rangle = E^0_T + \frac{n}{2m}\Delta t^2 \langle (\delta F^n)^2 \rangle, \tag{13}$$

where
$\langle E^n_T \rangle$ – mean total energy of a particle after n-th timestep,
E^0_T – total energy of a particle in the absence of random force,
$\langle \cdot \rangle$ – mean value operator defined as

$$\langle X^n \rangle = \int X^n \prod_{j=1}^{n-1} P(X^j)\, dX^j, \tag{14}$$

where
$P(X^j)$ – the probability distribution of random force X^j at time level j.

Equation (13) represents linear "heating" of the particle system with respect to the mean square of the random force. If the random force has a uniform distribution in the range $-F_a \leqslant \delta F \leqslant F_a$ then according to [19]

$$\langle \delta F^2 \rangle = \tfrac{1}{3} F_a^2. \tag{15}$$

On the basis of results presented in [19] we could estimate the error introduced by the fixed point arithmetics in the process described by Equations (6–12), still adopting the model of the simple harmonic oscillator [19]. Contrary to the random forces δF^j for which [19]

$$\forall i \neq j; \quad \langle \delta F^i \cdot \delta F^j \rangle = 0, \tag{16}$$

the random forces R^j given by Equations (8), (9) are as follows (for one dimensional case)

$$\langle R^{j+1} \cdot R^j \rangle = -m\langle (A^j)^2 \rangle, \tag{17}$$

$$\langle R^j \cdot R^j \rangle = -m[\langle (A^j)^2 \rangle + \langle (A^{j+1})^2 \rangle]. \tag{18}$$

Using analytical equations presented in [19] and Equations (17), (18) we obtain that

$$\langle E_T^n \rangle = E_T^0 + \frac{n}{2m} \Delta t^2 (2\omega^2 \Delta t^2) \langle (mA^n)^2 \rangle. \tag{19}$$

where ω is the angular frequency of the oscillator. From Equations (9), (15), (19) we obtain

$$\langle E_T^n \rangle = E_T^0 + n \frac{m}{12} \omega^2. \tag{20}$$

Hence, similar to [19], numerical heating is expected, but the total energy slope can be diminished for processes with low ω value. Additionally it is independent of Δt (contrary to Equation 13) and depends on physical values only.

Assuming that

$$E_T^n = E_k^n + E_p^n = 0, \tag{21}$$

the relative fluctuation coefficient of the total energy, η, is equal to

$$\eta = \frac{\Delta E^n}{E_{p\mathrm{MAX}}^n} = n \frac{1}{6K^2}, \tag{22}$$

where
$\Delta E^n = \langle E_T^n \rangle - E_T^0$,
E_k^n – kinetic energy,
E_p^n – potential energy,
$E_{p\mathrm{MAX}}^n = \frac{1}{2} m\omega^2 K^2$,
K – amplitude of oscillations.
For integer representation we can write that

$$K^2 = \mathrm{MAXINT}. \tag{23}$$

Equations (22) and (23) confirm that for more accurate computations MAXINT should be as large as possible.

3. RESULTS AND CONCLUSIONS

Some MD simulations have been performed with the method presented above for Ar–Kr mixture in 116 K. In the algorithm forces between mutually interacting particles are calculated using potential tables. Number of grid points in the tables (for $|\mathbf{r}_i - \mathbf{r}_j| \leqslant R_{\text{cut}}$) has been chosen to be equal to $v = 8000$, due to sufficiently low R.M.S. of the total energy in that case (see Figure 1 for comparison of the total energy R.M.S. for different v).

Simulation results obtained with the fixed point arithmetics are compared with those for the floating point arithmetics [14] (see Figure 2). Fluctuations of the total energy for the fixed point arithmetics are distinctly greater than those obtained using

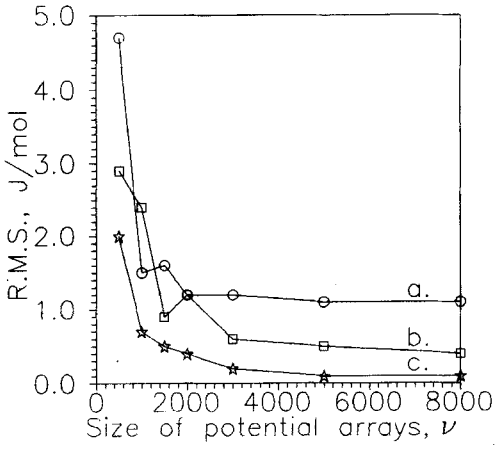

Figure 1 R.M.S. of the total energy for the floating point distance calculations. a. 256 particles, b. 500 particles, c. 2916 particles (for 750 production timesteps).

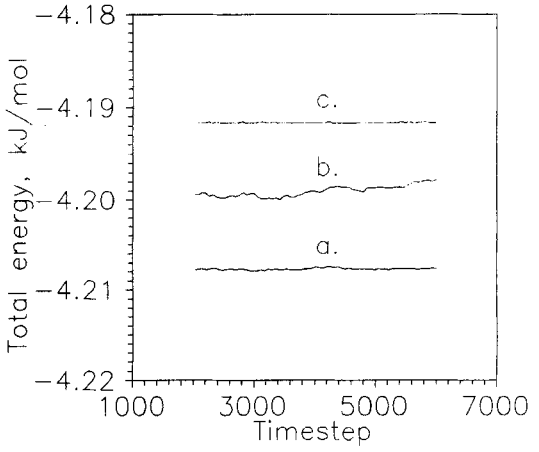

Figure 2 Fluctuations of the total energy for the MD simulation. a. floating point distance calculations ($v = 8000$), b. integer distance calculations ($v = 8000$), c. direct forces calculations with L-J equation. Temperature scaled till 2000 timestep.

floating point calculations, however, they are equal to $\sim 0.01\%$ of the total energy only. This value may be acceptable for the most of MD simulations.

In our one dimensional model of inaccuracies, the value of $\langle |R^n| \rangle$ is usually greater than $\langle |\delta F^n| \rangle$ and is equal to several percent of $\langle |F^n| \rangle$ ($|\cdot|$ means the vector length). Results for the MD simulations show, that the slope of the total energy is relatively low (see Figure 2b) and for moderate number of timesteps could be neglected. The slope can be explained roughly assuming that in liquid each pair of molecules may be considered as an oscillator pair with low ω^2 value (*cf.* Equation (20)). Other characteristics as *rdf* (see Figure 3) and *msd* (Figure 4) are very similar also to those calculated with floating point arithmetics. There is no significant difference in the reported results between the two methods.

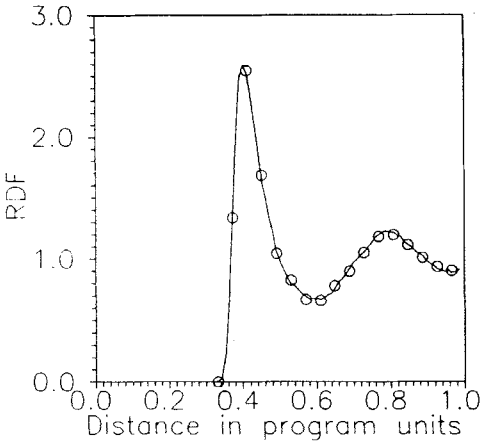

Figure 3 Radial distribution functions for Argon obtained using the *cube technique* (line) and the *cube technique* with integer distance calculation (circles).

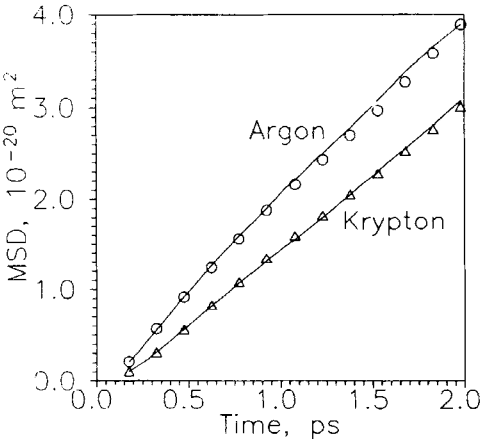

Figure 4 MSDs for Argon and Krypton obtained using the *cube technique* (lines) and the *cube technique* with integer distance calculations (symbols).

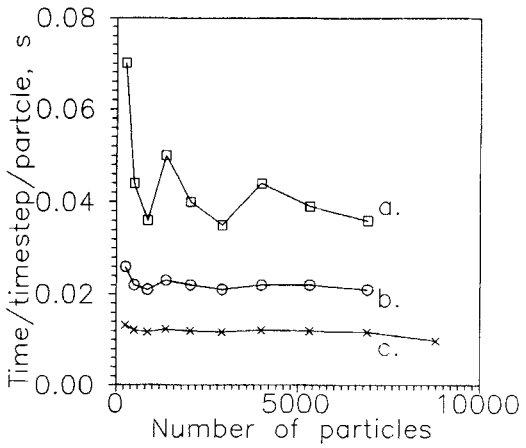

Figure 5 Execution time for MD timestep per particle for various *Forces* procedures, $R_{cut} = 2.5\sigma$. a. Smith algorithm [8] coded in C-language, b. *cube technique* [14], c. *cube technique* with integer distance calculation.

It can be expected, however, that other functions which are usually applied in processing of MD results will differ significantly. This refers to correlation functions and functions depending on differences in particles coordinates, velocities and forces [20]. Some experiments with applications of 32-bit floating point arithmetics show [20] that the quantities mentioned above could differ in tens of percent in comparison with 64-bit calculations. With integer calculations the difference could be even greater due to smaller number of significant digits in integers (see Equations (1), (2)) than in 32-bit reals used for determination of particle coordinates.

Figure 5 shows computational time savings of integer distances calculations in comparison with some other methods. Experiments with $R_{cut} = 3.5\sigma$ and $R_{cut} = 1.5\sigma$ have given similar effect in computational time (σ is the L-J parameter for Krypton).

The simulation results for Ar–Kr mixture in 116 K [15, 21] have been obtained using the cube technique [14]. All the calculations have been performed on the IBM PC/AT clone with DSI-020 plug-in board (MC68020/68881, 12.5 MHz, 1 MB RAM).

Using the fixed point arithmetics for interparticle distances computations the following properties are observed:

- considerable increase of calculational efficiency,
- simulation refers to the situation when particles interact each other with Lennard-Jones short range force $\mathbf{F}_i^n$ and additional random force $\mathbf{R}_i^n$,
- additional, insignificant in comparison with floating point calculations, total energy increase for moderate number of timesteps and acceptable fluctuations.

The method can be applied for MD liquid simulations in PC/workstation environment, when the high precision of calculations is not strongly required. It is not suggested for solid state simulations where molecular oscillations are supposed. It could be used for investigations of propagation of the truncation error. The error may be controlled changing MAXINT value. It would be interesting to apply this method for systems which are continuously heated.

Acknowledgements

We wish to thank Dr. Z.A. Rycerz from the University of Western Ontario, London, Canada, for many valuable discussions during preparation of the manuscript. We are grateful to Professor Jerzy Kapelewski for arranging partial financial support under project number CPBP 01.08 D4.3.

References

[1] Fincham, D., "Parallel Computers and Molecular Simulation", *Mol. Simul.*, **1**, 1 (1987).
[2] Hockney, R.W. and Eastwood, J.W., "Computer Simulation Using Particles", McGraw Hill (1981).
[3] Rycerz, Z.A. and Jacobs, P.W.M., "Vectorized Program of Order *N* for Molecular Dynamics Simulation of Condensed Matter, I. MDPYRV 1: Vector Pyramid, Short-Range Interactions, submitted to *Comput. Phys. Commun.*
[4] Rycerz, Z.A. and Jacobs, P.W.M., "Vectorized Program of Order *N* for Molecular Dynamics Simulation of Condensed Matter, II. MDSLAB 1: Slab, Short-Range Interactions", submitted to *Comput. Phys. Commun.*
[5] Sullivan, F., Mountain, R.D. and O'Connell, J., "Molecular Dynamics on Vector Computers", *J. Computat. Phys.*, **61**, 138 (1985).
[6] Schoen, M., "Structure of a Simple Molecular Dynamics Fortran Program Optimized for Cray Vector Processing Computers", *Comput. Phys. Commun.*, **52**, 175 (1989).
[7] Boris, J., "A Vectorized Near Neighbours Algorithm of Order N Using a Monotonic Logical Grid", *J. Computat. Phys.*, **66**, 1 (1986).
[8] Smith, W., "Fortran Code for the LINK-CELL Method", *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, informal Newsletter, Daresbury Laboratory, No. 20, 52 (1986).
[9] Heyes, D.M., "Correction to Cray Vectorised Link Cell Code", *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, informal Newsletter, Daresbury Laboratory, No. 26, 68 (1987).
[10] Dzwinel, W., Bargiel, M., Kitowski, J. and Mościński, J., "Linked Lists and Methods of Lights in Molecular Dynamics Simulation - Search for the best Method of Forces Evaluation in Sequential MD Codes", *Mol. Simul.*, **4**, 229 (1989).
[11] Mościński, J., Kitowski, J., Rycerz, Z.A. and Jacobs, P.W.M., "A Vectorized algorithm on the ETA 10-P for Molecular Dynamics Simulation of Large Number of Particles confined in a Long Cylinder", *Comput. Phys. Commun.*, **54**, 47 (1989).
[12] Mościński, J., Bargiel, M., Kitowski, J., Skotniczny, Z., Rycerz, Z.A., and Jacobs, P.W.M., "Vectorized Molecular Dynamics Algorithms for very large Number of Particles", Proc. of ICS '89 International Conference on Supercomputing, Crete, Greece, June 5–9 1989.
[13] Raine, A.R.C., Fincham, D., and Smith, W., "Systolic Loop Methods for Molecular Dynamics Simulation using Multiple Transputers", *Comput. Phys. Commun.*, **55**, 13 (1989).
[14] Mościński, J., Dzwinel, W. and Kitowski, J., "C-language code running on PC?? How Forces() procedure looks like?", *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, informal Newsletter, Daresbury Laboratory, No. 27, 26 (1988).
[15] Bargiel, M., Dzwinel, W., Kitowski, J. and Mościński, J., "C-language Molecular Dynamics Program for the Simulation of Monoatomic Molecular Mixtures", presented for publication in *Comput. Phys. Commun.*
[16] Rapaport, D.C., "Microscale hydrodynamics: Discrete-particle simulation of evolving flow patterns", *Phys. Rev. A*, **36**, 7, 3288 (1987).
[17] Bowler, K.C. and Kenway, R.D., "Physics on parallel computers. Part 1: the new technology", *Contemp. Phys.*, **28**, 6, 573 (1987).
[18] Bowler, K.C. and Kenway, R.D., "Physics on parallel computers. Part 2: applications", *Contemp. Phys.*, **29**, 1, 33 (1988).
[19] Amini, M., Eastwood, J.W. Hockney, R.W., "Time Integration in Particle Models", *Comput. Phys. Commun.*, **44**, 83 (1987).
[20] Rycerz, Z.A., private communication.
[21] Smith, W., "The Program ADMIXT, a Molecular Dynamics Program for the Simulation of Monoatomic Liquid Mixtures", CCP5 Program Library, June 1983.