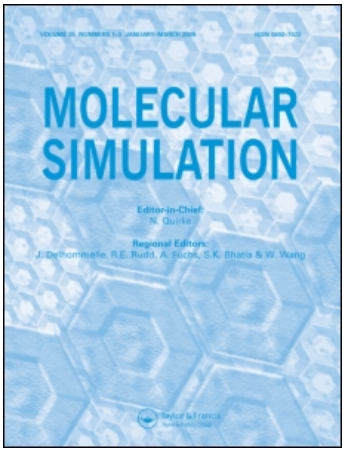


Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Molecular Dynamics on an Excel Spreadsheet

I. J. Fraser[a]; L. V. Woodcock[a]
[a] Department of Chemical Engineering, University of Bradford, Bradford, UK

To cite this Article Fraser, I. J. and Woodcock, L. V.(1995) 'Molecular Dynamics on an Excel Spreadsheet', Molecular Simulation, 15: 3, 135 — 140
To link to this Article: DOI: 10.1080/08927029508024050
URL: http://dx.doi.org/10.1080/08927029508024050

Molecular Simulation, 1995, Vol. 15, pp. 135–140
Reprints available directly from the publisher
Photocopying permitted by license only

MOLECULAR DYNAMICS ON AN EXCEL SPREADSHEET

I. J. FRASER and L. V. WOODCOCK*

Department of Chemical Engineering, University of Bradford, Bradford, BD7 1DP UK

(Received April 1995, accepted April 1995)

KEY WORDS: EXCEL spreadsheet, Molecular dynamics, Lennard-Jones, Soft spheres.

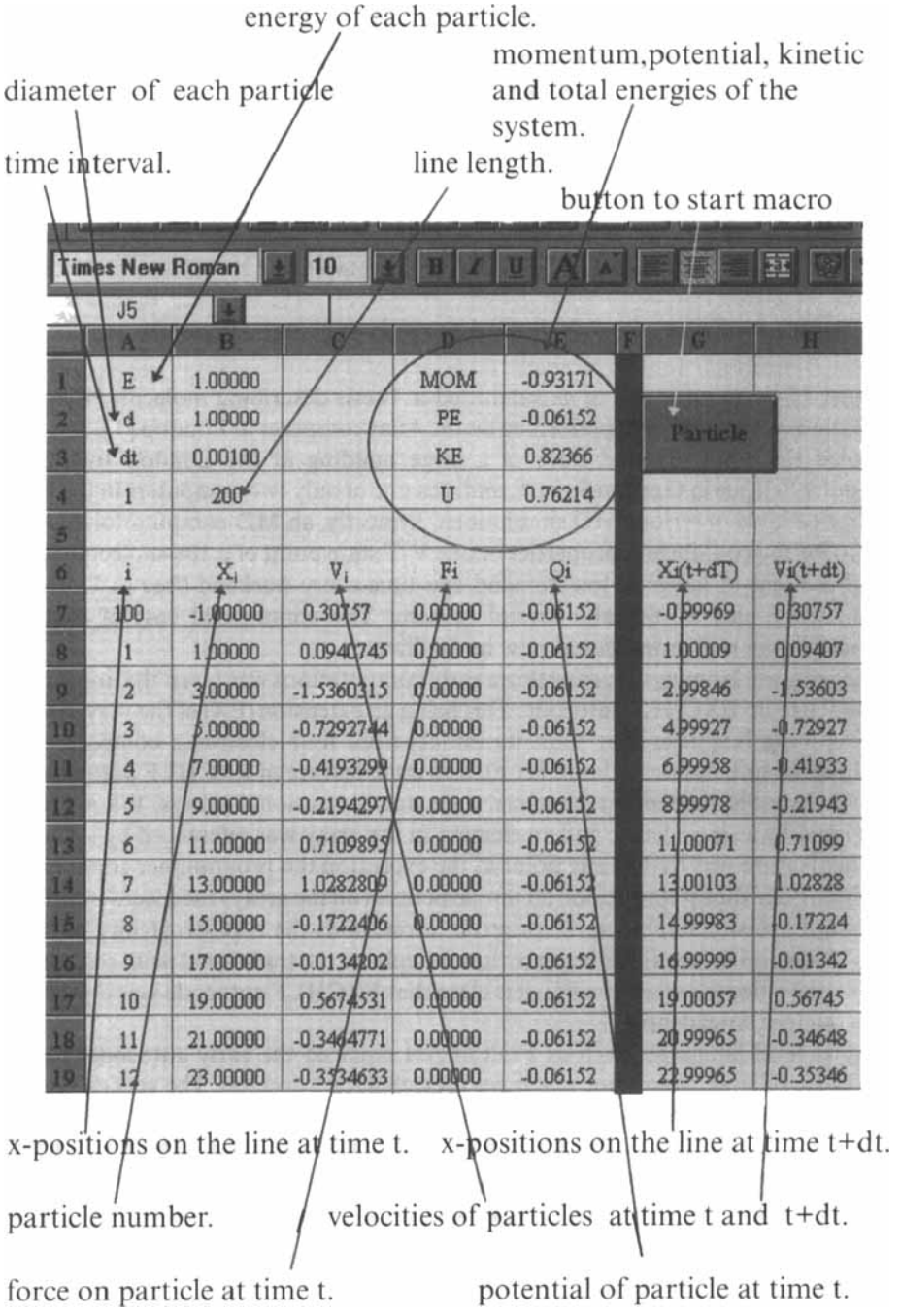
It is just 25 years since one of us submitted a Thesis describing molecular dynamics simulations which had been performed on the Atlas computer in London [1]. The Atlas occupied the whole ground floor of a large building at the London Institute of Computer Science in Gordon Square, and was one of only two computers in the UK at the time capable of serious MD simulations. Typically, an MD computation reported in 1970 for the equilibrium properties of one V-T state point of a 108-molecule system would use tens of hours of low-demand cpu-time every weekend (free to University 'large users'), and would take several months! The commercial cost of one such computation in real terms would now be £millions!

The program language used for those early computations was given the unpronounceable acronym EXCHLF autocode. This being an extension (EX) of the very first user programming language that took its initials from four electronic companies that collaborated to build the Atlas (CHLF). The main feature of EXCHLF autocode was that all the cells for storing numbers were one-dimensional arrays; the array was designated by a single letter, and an element in the array was referenced by a number. Another feature was that at any point in the execution the programmer could ask for a 'postmortem' dump whereupon all the elements of all the arrays used were transferred in a two-dimensional table to the printout. If one adds the EXCEL facility of "filling down", to set up the arrays to begin with, and replaces the traditional print out with the continuous video monitor of a PC, it is clear that EXCHLF autocode was a precursor of the modern spreadsheet.

Higher level languages such as FORTRAN replaced the early autocodes but the general use of large mainframe machines prevailed for 2 decades. The general trend in scientific simulations in recent years, however, has been towards decentralised facilities, group workstations, and to smaller and ever more powerful desk-top PC computers. Whilst many scientific simulation packages have followed and, indeed helped to set this trend, MD simulation has, if anything, gone the other way, with emphasis being placed on more sophisticated languages, larger systems and parallel architecture computers [2]. Fortran has remained the dominant programming language for 25 years, and

* e-mail L.V. Woodcock@bradford.ac.uk

Excel Simulation Screenshot



	A	B	C	D	E	F	G	H
1	E	1.00000		MOM	-0.93171			
2	d	1.00000		PE	-0.06152			
3	dt	0.00100		KE	0.82366			
4	1	200		U	0.76214			
5								
6	i	X_i	V_i	Fi	Qi		Xi(t+dT)	Vi(t+dt)
7	100	-1.00000	0.30757	0.00000	-0.06152		-0.99969	0.30757
8	1	1.00000	0.0940745	0.00000	-0.06152		1.00009	0.09407
9	2	3.00000	-1.5360315	0.00000	-0.06152		2.99846	-1.53603
10	3	5.00000	-0.7292744	0.00000	-0.06152		4.99927	-0.72927
11	4	7.00000	-0.4193299	0.00000	-0.06152		6.99958	-0.41933
12	5	9.00000	-0.2194297	0.00000	-0.06152		8.99978	-0.21943
13	6	11.00000	0.7109895	0.00000	-0.06152		11.00071	0.71099
14	7	13.00000	1.0282809	0.00000	-0.06152		13.00103	1.02828
15	8	15.00000	-0.1722406	0.00000	-0.06152		14.99983	-0.17224
16	9	17.00000	-0.0134202	0.00000	-0.06152		16.99999	-0.01342
17	10	19.00000	0.5674531	0.00000	-0.06152		19.00057	0.56745
18	11	21.00000	-0.3464771	0.00000	-0.06152		20.99965	-0.34648
19	12	23.00000	-0.3534633	0.00000	-0.06152		22.99965	-0.35346

Figure 1

although current 486- and Pentium- generation PCs will load Fortran compilers, it has remained somewhat unwieldy as a PC application. Thus, although simulations on PCs are now perfectly practicable, "popular MD" has not really taken off.

Take off may, however, be imminent. We can now report that it is relatively easy to solve the classical particle-simulation problem on a modern spreadsheet, and moreover, EXCEL is especially user-friendly for these simulations to the uninitiated. The time required to complete a single V-T calculation for 100 LJ atoms is typically less than 1 hour on a modern desktop computer, i.e. roughly one time increment per second of real time in the integration of the equations of motion using a simple one-step algorithm.

As part of an undergraduate teaching/research project [3], we have developed MD computations for soft-sphere and Lennard-Jones pair-potential models on an EXCEL spreadsheet, for both 1- and 2- dimensional systems. The spreadsheet computations run on EXCEL 5 on both 486-66mhz IBM compatibles, and, slightly faster, on an Apple Macintosh 8100/80av Power PC.

The first problem attempted was a 1D simulation of 100 softspheres with periodic boundary conditions. The "screenshot" of the spreadsheet at the beginning of the computation is shown in Figure 1 together with the annotation. Note that at $t = 0$ the

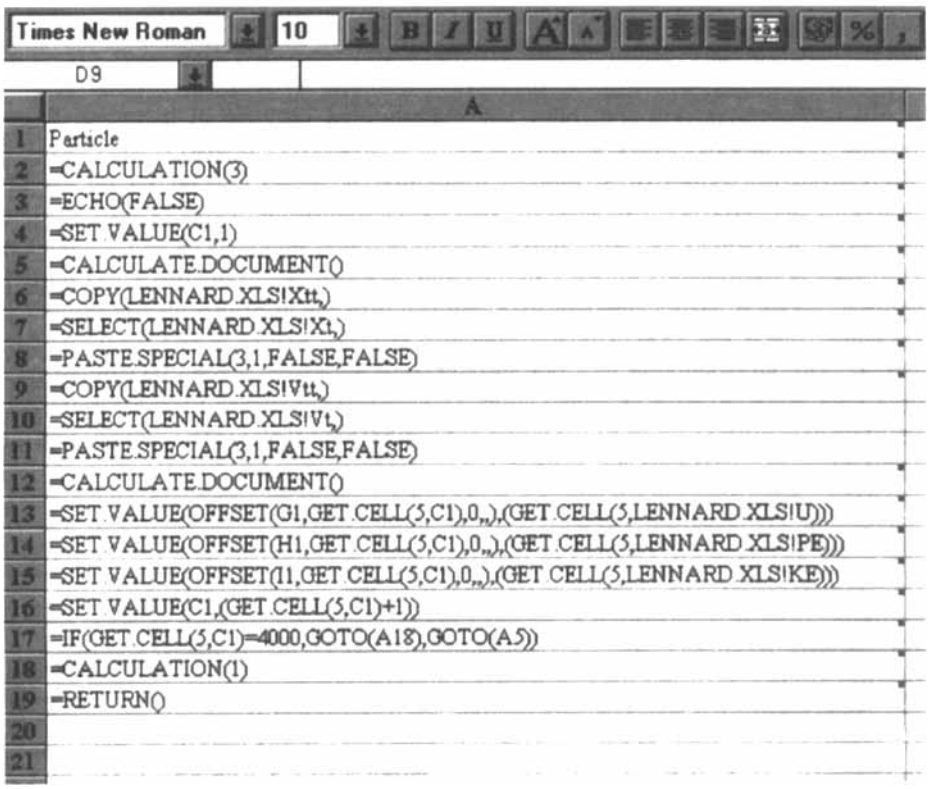


Figure 2 A screenshot of the macro that runs the Excel particle simulation. The operations 1-19 are described in the Appendix.

particles are in an ordered array, and the forces are initially zero. Using the customised 'button' facility, the simulation is started just by clicking on the start button. The EXCEL macro [4] sheet is shown in Figure 2. This is a list of instructions which the EXCEL application will follow in sequence, rather like a conventional high-level sequential program. The EXCEL application also conveniently collects and plots, on request, the progress of the simulation. It is very easy to visualise the progress of a typical simulation as demonstrated in Figure 3 either pictorially or graphically. This shows the EXCEL graph plotter picture of 4000 Δt runs of 100 soft spheres in 1D at constant total energy, and for a thermostatted isothermal simulation of 100 Lennard Jones molecules in 2D.

The progression to higher dimensions requires a slightly different approach because the interacting particle "neighbours" are changing with time. The 2-dimensional simulation runs about 8 times slower than the 1D simulation for the same number of particles. The program works as follows. A table is set up in EXCEL with the headings as shown below.

Table 1 Spreadsheet column headings for a two-dimensional particle simulation

Particle	i	X	Y	Vx	Vy	Fx	Fy	X(t+	Y(t+	Vx(t+	Vy(t+

From the values of the positions of the particles (X and Y) seven further tables are set up for the interactions between any two particles on a second spreadsheet. EXCEL version 5 is especially convenient for these MD simulations because of its novel three-dimensional facility of stacked spreadsheets.

Then, from the values of X_i and Y_i seven further tables can be set up for the interactions between two particles, notably pair separations $X_i - X_j$, $Y_i - Y_j$, r_{ij}, pair potential ϕ_{ij}, pair forces Fx_{ij}, Fy_{ij}, and F_{ij} the total force. When all the components of the forces have been calculated, the computation proceeds as demonstrated in Figure 1. The forces on each particle are summed and put into the appropriate column of the spreadsheet. Then the MACRO proceeds and the new values of the velocities and positions may be calculated and the computation proceeds as for one-dimensional simulations. A comparison of speeds for the calculations shown in Figure 3 is given in the table below as the time in hours minutes for 4000 MD time increments on desktop computers using an EXCEL5 spreadheet

Table 2 Timings for molecular dynamics on Pc's.

	100 *soft-spheres 1D*	100 *L-J 2D*
IBM 486 66 Mhz	35 mins.	4 hours 35 mins.
Apple Power PC 8100	25 mins.	3 hours 20 mins.

In contrast to the cost of MD on the Atlas of 25 years ago, the total purchase cost of a 486 PC plus the EXCEL 5 licence is presently less than £1000. This development brings round-the-clock molecular simulations to the desk-top of millions of would-be molecular simulators in education and industry. A Monte Carlo computation is

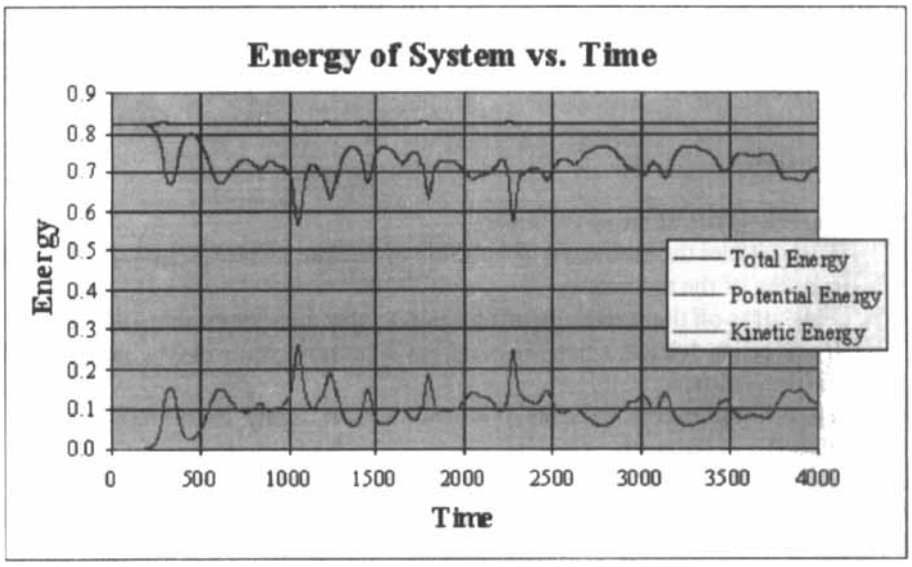


Figure 3a Energy of system vs. Time for a 100 particle system with Soft-Sphere potentials, and a time increment of 0.001.

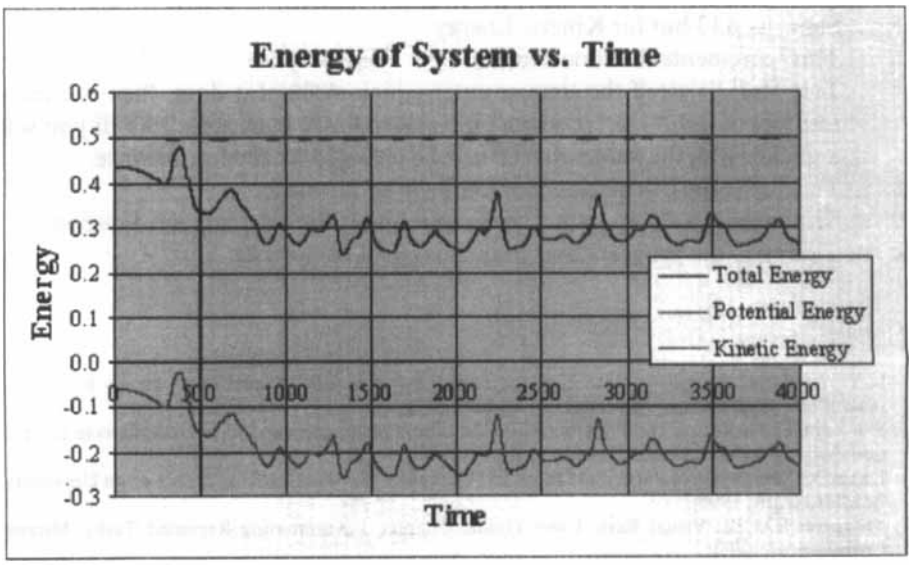


Figure 3b Energy of System vs. Time for an isothermal 100 particle system with Lennard-Jones potentials, and a time increment of 0.001.

presently underway. More details on the calculations outlined here are given in reference [3], and further information of the methods and results are available from the authors on request.

Cell	Operation
A1	Particle is the name of the macro.
A2	This disables the automatic calculation of the spreadsheet, which speeds up the running of the macro.
A3	This turns off the screen updating again to speed up the running of the macro. If this cell is deleted, when the program is running you will be able to 'see' the particles moving.
A4	This initialises the variable that counts how many time increments have passed.
A5	This calculates all the values of all the cells in the whole worksheet.
A6	This copies the positions of the particles at time $t+dt$ to the clipboard (a workspace).
A7	This selects the area where the particles' positions are stored at time t
A8	This copies the particle positions from the clipboard to time t.
A9	Same as A6 but for velocities.
A10	Same as A7 but for velocities.
A11	Same as A8 but for velocities.
A12	This calculates the document again.
A13	This copies the instantaneous value of the total energy to a space so that graphs can be drawn at a later date.
A14	Same as A13 but for Potential Energy.
A15	Same as A13 but for Kinetic Energy.
A16	This increments the time counter.
A17	This tests to see if the time counter equals 4000, if it does, then the macro resumes at cell A18, if it doesn't the macro loops back to cell A5. If you want a qucker run, the value of 4000 can be changed to 100 for instance.
A18	This turns the calculation method back to normal.
A19	This sounds a Beep, so that you can tell that the program has finished.
A20	This ends the program and returns control to the user.

References

[1] L. V. Woodcock "*Studies of the thermodynamic and structural praperties of liquids by computer simulation*", PhD Thesis, University of London (1970).

[2] S. J. Zara, D. Nicholson and J. Barber, "The use of supercomputers and microcomputers in computational chemistry", *Molecular Simulation* **1**, 299 (1988)

[3] I. Fraser, "Simulation of Particulate Flow and Computer Graphics", MEng Dissertation University of Bradford (April 1995).

[4] Microsoft EXCEL Visual Basic Users Guide, Chapter 1 Automating Repeated Tasks; Microsoft Corporation (1995).