Daresbury Laboratory

INFORMATION QUARTERLY FOR COMPUTER SIMULATION OF CONDENSED PHASES

An informal Newsletter associated with Collaborative Computational Project No.5
on Molecular Dynamics, Monte Carlo & Lattice Simulations of Condensed Phases.

Number 38 August 1993

Contents

Editor: Dr. M. Leslie

SERC Daresbury Laboratory
Daresbury, Warrington WA4 4AD UK

General News

STEERING COMMITTEE MEETING OF CCP5

A meeting of the steering committee of CCP5 will be held on 16th. September 1993 at the University of Keele. This is timed to coincide with the CCP5 annual meeting. It will be necessary to elect 3 new members to the executive committee at this meeting. N. Allan (Chemistry, Bristol), M. Rodger (Chemistry, Reading) and S. Price (Chemistry, University College London) will have completed three years. The other members of the executive committee are Dr. M. Allen (Dept. of Physics, Bristol), Dr. J. Goodfellow (Dept. of Crystallography, Birkbeck College London) and Dr. W. Mackrodt (ICI). The executive committee consists of the chairman, 6 other members and the two permanent Daresbury staff. The members serve on the committee for three years. The committee meets three times a year to deal with day-to-day business of CCP5. If you would like to nominate someone for the executive committee please return the slip below.

My nominations for executive committee: ______________________
are

: ______________________

: ______________________

Signed : ______________________

FUTURE MEETINGS

A summary table is given below, further details may be found inside. CCP5 has been asked to publisise the molecular liquids meeting as it will be of interest to our readers but is not involved with its organisation.

TOPIC	DATES	LOCATION
LARGE SCALE MOLECULAR SIMULATION	SEPTEMBER 15-17 1993	UNIVERSITY OF KEELE
MOLECULAR LIQUIDS	SEPTEMBER 24-29 1993	UNIVERSITY OF KENT

CCP5 PROGRAM LIBRARY

Details are contained in this issue of accessing the CCP5 program library through E-mail automatically. Also details of accessing the Allen/Tildesley example programs at Cornell. A new improved version of MDCSPC4 is available incorporating modifications suggested by Dr. U. Essmann, City University of New York. An updated version of SFMK has been supplied by the author A.P. Lyubartsev. New additions to the library from our readers are always welcome.

CRAY NEWS

CCP5 participants are reminded that CCP5 has an annual allocation of Cray time at Rutherford Laboratory. This is available for the development of simulation programs which are of general use to the CCP5 community. Readers who wish to use some of this allocation should write to the CCP5 Secretary, Dr. M. Leslie, TCS Division, SERC Daresbury Laboratory, Daresbury, Warrington WA4 4AD.

INTEL NEWS

CCP5 also has an annual allocation of time on the Intel IPSC/860 at Daresbury. If any CCP5 member wishes to make use of some of this time please contact M. Leslie at Daresbury.

CCP5 FUNDS FOR COLLABORATIONS

CCP5 can make available funds of up to £300 per annum for groups of two or more UK researchers wishing to undertake a collaborative project within the scientific area covered by CCP5. The funds are intended to cover travel and subsistence costs. Researchers who wish to apply for funds are requested to submit a brief proposal (about 1/2 a page) describing the intended work to Dr. M. Leslie, SERC Daresbury Laboratory, Daresbury, Warrington, Cheshire. Alternatively reply by Email to M.LESLIE@UK.AC.DARESBURY

CCP5 VISITORS PROGRAM

CCP5 organises a visitors program which funds the visit to the UK of overseas collaborators. We would normally expect a visitor to visit three sites in the UK and give a lecture at each site. These lectures would be open to all members of CCP5 as well as members of the host university. The visit would normally last between one or two weeks. CCP5 would pay for the cost of travel to the UK and within the UK between universities. CCP5 would expect some contribution towards accommodation expenses at the host university to be met by the university. We will also consider longer collaborations or visits just one place if this can be justified by the nature of the work to be done. If you have an overseas collaborator who you would like to invite under this program, please make a request to Dr. M. Leslie, SERC Daresbury Laboratory, Daresbury, Warrington, Cheshire. UK Alternatively reply by Email to M.LESLIE@UK.AC.DARESBURY

JANET:	M.LESLIE@DARESBURY
INTERNET:	M.LESLIE@daresbury.ac.uk
EARN/BITNET:	M.LESLIE%daresbury.ac.uk@ukacrl

REQUEST FOR CONTRIBUTIONS

The deadline for contributions for the next 2 newsletters will be **15 September 1993 and 15 December 1993** Readers are reminded that contributions are always welcome. Contributions may be sent by Email in LaTeX; this makes the task of collating the newsletter simpler for the editor.

Contributors to the current issue

Our thanks go to:

D. Fincham
Department of Physics
Keele University

R. Wcisło
W. Dzwinel
J. Kitowski
J. Długopolski
J. Mościński
W. Alda
D. Nikolow

Institute of Computer Science
AGH
Cracow
Poland

CCP5 ANNUAL MEETING

LARGE SCALE MOLECULAR SIMULATION

Keele University
15th - 17th September, 1993

The aim of this meeting is to report and discuss the simulation of large complex systems made possible by recent advances in high performance computing. The title is intentionally broad. Examples of problems and systems within the scope of the meeting are: polymers and biomolecules; structural organisation and self-assembly; micelles and surfactants; surface processes involved in STM; and many others.

The invited speakers are:

D.L. Beveridge,	Wesleyan University, Connecticut
S. Homans,	Dundee University
U. Landman,	Georgia Institue of Technology
B. Smit,	Shell Laboratories, Amsterdam
D.J. Tildesley,	Southampton University

Contributed papers are warmly welcomed, and there will be poster as well as oral sessions.

The conference will be held in the pleasant park-like surroundings of Keele University. It will begin with lunch on Wednesday September 15th and end with lunch on Friday September 17th. Accommodation will be available on campus. The cost of 40 pounds for non-residents includes registration, lunches on Wednesday, Thursday and Friday, and coffee and tea breaks. The fee of 120 pounds for residents includes in addition dinner, bed and breakfast for the nights of 15th and 16th September. Students can come at HALF PRICE. To book for the meeting please send the registration form below as soon as possible.

Dr. David Fincham
Department of Physics
Keele University
Staffordshire, ST5 5BG, U.K
Email: D.Fincham@KEELE.AC.UK
FAX: +44 - 782 - 711093

REGISTRATION FORM, “LARGE SCALE MOLECULAR SIMULATION”

Name:

Address:

Email:

Please tick one:

Resident	£120	□
Student resident	£60	□
Non-resident	£40	□
Student non-resident	£20	□

Students please enclose a supervisor’s letter.

Special dietary requirement:

U.K participants should pay in advance: please enclose a cheque payable to ”CCP5 Meeting”.

Contributed paper: please give title and enclose abstract.

Title:

RESEARCH CONFERENCE ON

MOLECULAR LIQUIDS:

Computer Simulations and Experiments

organized in association with the

European Molecular Liquids Group (EMLG)

University of Kent at Canterbury, United Kingdom, 24–29 September 1993

Chairperson	A. Geiger (Dortmund)
Vice–Chairperson:	R.M. Lynden–Bell (Cambridge)
Local Organisation:	J.C. Dore (Canterbury)

Scope of the Meeting

The interpretation of experimental data on molecular liquids very often suffers from the necessity to use oversimplified analytic models, which are not able to reproduce the details of the structure and dynamic in such systems. Recent developments in computational techniques and the continuous improvement of interaction potentials now allow the realistic simulation of molecular liquids. This opens the possibility to use simulations for the evaluation of experimental data. This largely increases the power of many experimental techniques.
The goals of the planned meeting are

- to discuss the present state of simulation techniques and their power to reproduce 'real' molecular liquids,
- to discuss the possibilities to extract data from simulations, which are directly comparable to various experimental results,
- to plan experiments, which are suitable for combination with simulations,
- to give recent examples of such efforts.

A list of scientists who have so far accepted to give a presentation or to act as session chairpersons is as follows:

S. Bratos	*Université P. et M. Curie, Paris, France*
M. Ferrario	*Universitá di Messina, Italy*
D. Frenkel	*FOM Institute for Atomic and Molecular Physics, Amsterdam, The Netherlands*
E. Guàrdia	*Universitat Polytècnica de Catalunya, Barcelona Spain*
B. Guillot	*Université P. et M. Curie, Paris, France*
W.G. Hoover	*University of California at Davis/Livermoore, USA*
G. Kneller	*Centre d'Etudes Nucléaires de Saclay, Gif-sur-Yvette, France*
P. Madden	*Oxford University, U.K.*
A. Maliniak	*Arrhenius Laboratory, Stockholm, Sweden*
G.W. Neilson	*University of Bristol, U.K.*
G. Palinkas	*Hungarian Academy of Sciences, Budapest, Hungary*
J.A. Padró	*Universitat de Barcelona, Spain*
H. Posch	*Universität Wien, Austria*
J.G. Powles	*University of Kent at Canterbury, U.K.*
J. Schnitker	*University of Michigan, Ann Arbor, USA*
W.A. Steele	*Pennsylvania State University, USA*
A.J. Stone	*University of Cambridge, U.K.*
S. Toxvaerd	*Ørsted Institute Copenhagen, Denmark*
R. Vallauri	*Università degli Studi di Trento, Povo, Italy*
P.-O. Westlund	*University of Umeå*
M.D. Zeidler	*RWTH Aachen, Germany*

A poster session will also be organized. A few contributed papers can be admitted for short oral presentation.

Computational Chemical Physics Specialist

The Parallel Applications Centre (PAC) of the University of Southampton applies parallel high-performance computing (HPC) technology to problems in industry and commerce. With a staff of nearly 20 HPC applications specialists, state-of-the-art HPC systems and a location on the University of Southampton's science park at Chilworth, the PAC provides a professional, high quality service to client organisations.

The PAC works closely with industrial and commercial HPC users and with leading third-party software developers, acting as both a porting centre for existing software and as an applications development house, producing new software to commercial standards. PAC projects cover a wide range of HPC applications from engineering and scientific supercomputing to high-performance databases and real-time embedded applications.

The PAC has an immediate requirement for a specialist in computational chemical physics to work on major projects with clients in the UK and Europe.

The successful applicant will possess an advanced practical knowledge of at least one area of computational chemical physics and a good awareness of several widely used computational chemical physics techniques. Advanced Fortran and Unix skills are essential. Experience in the use of parallel computing technology for computational chemical physics is highly desirable. Experience of working in or with industry would also be an advantage.

Competitive salaries, in the range £15,000 to £25,000 depending on experience, are offered.

Please send two copies of your CV to:

Dr Colin Upstill Parallel Applications Centre 2 Venture Road Chilworth Research Centre Southampton SO1 7NP

Tel. 0703-760834

The CCP5 Program Library

W. Smith

CCP5 Program Library Conditions of Distribution

The CCP5 Program Library provides programs and documentation free of charge to academic centres upon application to Dr. W. Smith, TCS Division, S.E.R.C. Daresbury Laboratory, Daresbury, Warrington WA4 4AD, U.K.. Please supply a magnetic tape to receive the copies. Industrial and commercial applicants should enclose a £100 handling charge. No magnetic tape need be sent in this case. Listings of programs are available if required. Please note that use of inappropriate packing for magnetic tapes (e.g. padded bags) may result in them being returned without the required software. Please ensure that these forms of packaging are not used. A list of programs available is presented in the following pages.

All applicants will be required to sign an agreement not to exploit the programs for commercial purposes e.g. for resale or distribution as part or whole of a commercial product.

Readers should also note that we are authorised to supply the example programs originally published in the book "Computer Simulation of Liquids", by M.P. Allen and D.J. Tildesley (Clarendon Press, Oxford 1987). These are supplied in the same manner as the resident CCP5 programs. We are grateful to Mike Allen and Dominic Tildesley for their permission.

We should also like to remind our readers that we would welcome further contributions to the Program Library. The Library exists to provide support for the research efforts of everyone active in computer simulation and to this end we are always pleased to extend the range of software available. If any of our readers have any programs they would like to make available, please would they contact Dr. Smith.

Please Note: For copyright reasons we are not able to supply the programs CASCADE, SYMLAT, THBFIT,THBPHON and THBREL free of charge to Universities outside the United Kingdom.

Programs from the Book: "Computer Simulation of Liquids" by M.P. Allen and D. Tildesley, Clarendon Press, Oxford 1987.

These programs originally appeared on microfiche in the book "Computer Simulation of Liquids" by M. P. Allen and D. J. Tildesley, published by Oxford University Press, 1987. They are made freely available to members of CCP5, in the hope that they will be useful. The intention is to clarify points made in the text, rather than to provide a piece of code suitable for direct use in a research application. We ascribe no commercial value to the programs themselves. Although a few complete programs are provided, our aim has been to offer building blocks rather than black boxes. As far as we are aware, the programs work correctly, but we can accept no responsibility for the consequences of any errors, and would be grateful to hear from you if you find any. You should always check out a routine for your particular application. The programs contain some explanatory comments, and are written, in the main, in FORTRAN-77. One or two routines are written in BASIC, for use on microcomputers. In the absence of any universally agreed standard for BASIC, we have chosen a very rudimentary dialect. These programs have been run on an Acorn model B computer. Hopefully the translation of these programs into more sophisticated languages such as PASCAL or C should not be difficult.

M.P.Allen

CCP5 Program Library E-Mail Service

From January 1 1991 it will be possible for CCP5 members to get copies of CCP5 programs through E-mail *automatically*. To do so they should send an E-mail message to **info-server@uk.ac.daresbury**. The contents of the e-mail message should be as follows (Note: the use of upper and lower case is significant - this is a unix system!):

```
request sources
topic index CCP/ccp5
topic CCP/ccp5/program-name
```

Where program-name is the name of the desired source code. A mail server will automatically process this message and return a copy of the source code to your e-mail address. Please note the following however:

The program source will be returned to you in **uue** format, which is a form of encoding most suitable for mail messages. It can easily be decoded on any unix system using the uudecode command. (Check your local unix *man* file for details). Also, to speed the transfer, the source will be split into files of 1200 records each, so expect two or three such files for the average CCP5 program. Once again, uudecode will help you to sort things out.

Readers who do not have unix facilities should include the following lines at the start of the above message:

```
line-limit: nnnnn
coding: off
```

Where nnnnn is the number of records in the source (in most cases 6000 should be enough). The program will be sent in plain FORTRAN as a single file. It may take a while to arrive, but be patient! Also beware in case your system mailer cannot handle messages of this size.

The programs from "Computer Simulation of Liquids" are also available. To access them use program-name F.01, for example.

Readers must realise that the terms of use and distrubution of the CCP5 programs that have applied hitherto will be maintained. Users should not redistribute or sell the programs, nor is any liability accepted for their use, either by SERC or the program authors. It is a requirement on the user that the programs be fully tested for their intended purpose. Any bugs found should be reported to the librarian, for the benefit of other users.

Lastly readers should realise that this means of transfer does not include any program documentation. So if you are unable to make sense of the programs, write for the documentation!

We are grateful to Mr. P. Griffiths of Daresbury's ITS Division for implementing this facility.

Availability of the Allen/Tildesley example programs at Cornell

Appendix F of the Book "Computer Simulation of Liquids" by M. P. Allen and D. J. Tildesley describes a method whereby the example programs may be obtained from the statistical mechanics group FTP facility at Cornell. This facility is no longer operational as advertised, due to software and hardware changes. However, the programs are still available. To obtain them, please follow the procedure outlined here. The description below is taken from the HELP file that is distributed by the file server; to obtain the Allen/Tildesley example programs, simply use "ALLEN_TILDESLEY" as the package name (without the quotes, note underscore character _ not hyphen -).

STATMECH is a file distribution service for the Statistical Mechanics community that uses electronic mail facilities to deliver files. To communicate with STATMECH, send an EMAIL message to: statmech@cheme.tn.cornell.edu Commands are sent in the body of hte message you send to STATMECH (not in the subject line). Several commands may be sent at one time; just put one command per line.

STATMECH commands:

SENDME package	Sends all parts of the specified package.
SENDME package.n	Sends part n of the specified package.
LIST [pattern]	Gives a brief description of all packages matching "pattern". If pattern is omitted, a description of all packages is sent.
HELP	Sends this help file.

For each request you make, a transaction log is returned to you indicating the status of the request. The status report will indicate whether the request was successfully completed, and when the file was or will be sent. Large files are sent only during off-peak hours.

Problems, questions and comments about STATMECH service on this system should be directed to "statmech-mgr@cheme.tn.cornell.edu".
Steve Thompson, School of Chemical Engineering, Cornell University, Ithaca NY 14853 USA.

THE CCP5 PROGRAM LIBRARY.

ADMIXT	[MD,LJA/MIX,LF,TH+MSD+RDF]	W. Smith
CARLOS	[MC,VS+Aquo,TH]	B. Jonsson & S. Romano
CARLAN	[DA,CARLOS structure analysis]	B. Jonsson & S. Romano
CASCADE	[LS,DIL,EM,TH+STR]	M. Leslie & W. Smith
CURDEN	[DA,Current Density Correlations]	W. Smith
DENCOR	[DA,Density Correlations]	W. Smith
HLJ1	[MD,LJA,LF,TH+MSD+RDF]	D.M. Heyes
HLJ2	[MD,LJA,LF,TH+MSD+RDF+VACF]	D.M. Heyes
HLJ3	[MD,LJA,LF/LC,TH+MSD+RDF]	D.M. Heyes
HLJ4	[MD,LJA,LF/CP+CT,TH+MSD+RDF]	D.M. Heyes
HLJ5	[MD,LJA/SF,LF,TH+MSD+RDF]	D.M. Heyes
HLJ6	[MD,LJA,TA,TH+MSD+RDF]	D.M. Heyes
HMDIAT	[MD,LJD,G5+Q4,TH+MSD+QC]	S.M. Thompson
HSTOCH	[MD/SD,VS+BA,LF+CA,TH]	W.F. van Gunsteren & D.M. Heyes
MCN	[MC,LJA,TH]	N. Corbin
MCLSU	[MC,LJA,TH]	C.P. Williams & S. Gupta
MCMOLDYN	[MD/MC,LJS+FC+AQ,LF+QF/G5+QS,TH+RDF]	A. Laaksonen
MCRPM	[MC,RPE,TH+RDF]	D.M. Heyes
MDATOM	[MD,LJA,G5,TH+RDF+MSD+QC]	S.M. Thompson
MDATOM	[MD,LJA,LF,TH+MSD+RDF]	D. Fincham
MDCSPC2P	[PRMD,BHM,LF,TH+STF+RDF+VACF+MSD]	W.Smith
MDCSPC4B	[PRMD,BHM+FC,G5+G4,TH+STF+RDF]	W. Smith
MDDIAT	[MD,LJD,LF+CA,TH+MSD]	D. Fincham
MDDIATQ	[MD,LJD+PQ,LF+CA,TH+MSD]	D. Fincham
MDIONS	[MD,BHM,LF,TH+MSD+RDF+STF]	D. Fincham & N. Anastasiou
MDLIN	[MD,LJL,G5+Q4,TH+MSD+QC]	S.M. Thompson
MDLINQ	[MD,LJL+PQ,G5+Q4,TH+MSD+QC]	S.M. Thompson
MD3DLJ_C	[MD,LJA/MIX,LF/LC,TH+MSD+RDF]	M. Bargiel, W. Dzwinel, J. Kitowski and J. Moscinski
MDMANY	[MD,LJS+FC,LF+QF,TH]	D. Fincham & W. Smith
MDMIXT	[MD,LJS/MIX,LF+QF,TH]	W. Smith
MDMPOL	[MD,LJS+FC/MIX,LF+QF,TH]	W. Smith & D. Fincham
MDNACL	[MD,BHM,LF,TH+MSD+RDF]	W. Smith
MDPOLY	[MD,LJS,G5+Q4,TH+MSD+QC]	S.M. Thompson
MDMULP	[MD,LJS+PD+PQ/MIX,LF+QF,TH]	W. Smith
MDSGWP	[MD,LJA/SGWP,LF,TH+VACF+RDF+QC]	W. Smith & K. Singer
MDTETRA	[MD,LJT,G5+Q4,TH+MSD+QC]	S.M. Thompson
MDZOID	[MD,GAU,LF+QF,TH+MSD+RDF+VACF]	W. Smith
NAMELIST	[UT, Namelist emulation]	K. Refson
NEMD	[MD,LJA,SLLOD,Macintosh]	D. J. Evans
NSCP3D	[UT, Hard sphere packing]	M. Bargiel & J. Moscinski
PIMCLJ	[PIMC,LJA,MC,TH+RDF+QC]	K. Singer & W. Smith

SCN	[MC,LJA,RFD,TH] N. Corbin
SMFK	[MC-SCF, Cylindrical Polyelec.] A.P. Lyubartsev
SLS_PRO	[MD,Proteins,LF,TH+RDF] A. Raine
SOTON_PAR	[MD,LJA,LC,TH] M.R.S. Pinches
SURF	[MD,BHM/TF/2D,LF,TH+RDF] D.M. Heyes
SYMLAT	[LS,PIL,EM+SYM,TH+STR] Harwell
TEQUILA	[GP] A. Wilton and F. Mueller-Plathe
THBFIT	[LS,PIL,EM,Potential fitting] Harwell
THBPHON	[LS,PIL/3B,EM,Phonon dispersion] Harwell
THBREL	[LS,PIL,EM,TH+STR] Harwell

Key:

Program types:		
	MD	Molecular dynamics
	MC	Monte Carlo
	PRMD	Parrinello-Rahman MD
	LS	Lattice simulations
	SD	Stochastic dynamics
	DA	Data analysis
	UT	Utility package
	PIMC	Path Integral Monte Carlo
	GP	Graphics program

System models:		
	LJA	Lennard-Jones atoms
	LJD	Lennard-Jones diatomic molecules
	LJL	Lennard-Jones linear molecules
	LJT	Lennard-Jones tetrahedral molecules
	LJS	Lennard-Jones site molecules
	RPE	Restricted primitive electrolyte
	BHM	Born-Huggins-Meyer ionics
	SGWP	Spherical gaussian wavepackets
	TF	Tosi-Fumi ionics
	VS	Variable site-site model
	BA	Bond angle model
	PD	Point dipole model
	PQ	Point quadrupole model
	MIX	Mixtures of molecules
	GAU	Gaussian molecule model
	FC	Fractional charge model
	PIL	Perfect ionic lattice model
	DIL	Defective ionic lattice model
	3B	3-body force model
	2D	Two dimensional simulation
	SF	Shifted force potential
	FC	Fractional charge model
	AQ	Aqueous solutions

Algorithm:	G5	Gear 5th order predictor-corrector
	Q4	Quaternion plus 4th order Gear P-C.
	LF	Leapfrog (Verlet)
	QF	Fincham Quaternion algorithm
	QS	Sonnenschein Quaternion algorithm
	LC	Link-cells MD algorithm
	CP	Constant pressure
	CT	Constant temperature
	TA	Toxvaerd MD algorithm
	CA	Constraint algorithm
	EM	Energy minimisation
	SYM	Symmetry adapted algorithm
	RFD	Rossky-Friedman-Doll algorithm
	SLLOD	Thermostatted SLLOD equations
Properties:	TH	Thermodynamic properties.
	MSD	Mean-square-displacement
	RDF	Radial distribution function
	STF	Structure factor
	VACF	Velocity autocorrelation function
	QC	Quantum corrections
	STR	Lattice stresses

Programs from the Book "Computer Simulation of Liquids"

F.1	Periodic boundary conditions in various geometries
F.2	5-value Gear predictor-corrector algorithm
F.3	Low-storage MD programs using leapfrog Verlet algorithm
F.4	Velocity version of Verlet algorithm
F.5	Quaternion parameter predictor-corrector algorithm
F.6	Leapfrog algorithms for rotational motion
F.7	Constraint dynamics for a nonlinear triatomic molecule
F.8	Shake algorithm for constraint dynamics of a chain molecule
F.9	Rattle algorithm for constraint dynamics of a chain molecule
F.10	Hard sphere molecular dynamics program
F.11	Constant-NVT Monte Carlo for Lennard-Jones atoms
F.12	Constant-NPT Monte Carlo algorithm
F.13	The heart of a constant μVT Monte Carlo program
F.14	Algorithm to handle indices in constant μVT Monte Carlo
F.15	Routines to randomly rotate molecules
F.16	Hard dumb-bell Monte Carlo program
F.17	A simple Lennard-Jones force routine
F.18	Algorithm for avoiding the square root operation
F.19	The Verlet neighbour list
F.20	Routines to construct and use cell linked-list method
F.21	Multiple timestep molecular dynamics
F.22	Routines to perform the Ewald sum
F.23	Routine to set up alpha fcc lattice of linear molecules
F.24	Initial velocity distribution
F.25	Routine to calculate translational order parameter
F.26	Routines to fold/unfold trajectories in periodic boundaries
F.27	Program to compute time correlation functions
F.28	Constant-NVT molecular dynamics - extended system method
F.29	Constant-NVT molecular dynamics - constraint method
F.30	Constant-NPH molecular dynamics - extended system method
F.31	Constant-NPT molecular dynamics - constraint method
F.32	Cell linked-lists in sheared boundaries
F.33	Brownian dynamics for a Lennard-Jones fluid
F.34	An efficient clustering routine
F.35	The Voronoi construction in 2d and 3d
F.36	Monte Carlo simulation of hard lines in 2d
F.37	Routines to calculate Fourier transforms

OPTIMISATION OF THE EWALD SUM

David Fincham, SERC Daresbury laboratory and Keele University

Introduction

It is part of the folk law of simulation that the execution time for the Ewald sum scales with N, the number of ions, as $N^{3/2}$. This result has appeared in print in various forms, e.g [1]. However, in view of a recent paper [2] which claims it is order N, I thought it worthwhile to go over the argument here, and bring out some points of practical importance.

We will consider a cubic simulation box of side L containing N particles. As we move to larger systems we of course keep the number density $n = N/L^3$ fixed, so L varies as $N^{1/3}$.

Specification of parameters

The real space part of the system energy involves a sum over ion pairs of the form

$$\sum_{i<j} q_i q_j \,\mathrm{erfc}(\alpha r_{ij})/r_{ij}$$

which depends on the complementary error function $\mathrm{erfc}(\alpha r)$ where α is the Ewald separation parameter. To ensure convergence of the real space part of the sum we need to choose α so that the erfc function is small at the real-space cutoff, R. Suppose we specify "small" to be $\exp(-p)$. Thus p is a parameter controlling the desired Ewald sum accuracy. At large values of the argument the erfc function behaves as $\exp(-\alpha^2 r^2)$. Thus we require

$$\alpha^2 R^2 = p$$

or

$$\alpha = p^{1/2}/R \qquad (1)$$

The reciprocal space contribution to the energy involves a sum of the form

$$\sum_{\mathbf{k}} k^{-2}\exp(-k^2/4\alpha^2)\ |\sum_i q_i \exp(i\,\mathbf{k}.\mathbf{r})|^2$$

The rate of convergence of this sum is controlled by the factor $\exp(-k^2/4\alpha^2)$. If we require the terms to have the value $\exp(-p)$ at the reciprocal space cutoff, K, then we find

$$p = K^2/4\alpha^2$$

or

$$K = 2\alpha p^{1/2} = 2p/R \qquad (2)$$

using (1).

Thus these two equations specify the choice of separation parameter α and reciprocal space cutoff K once the accuracy parameter p and real space cutoff R have been chosen.

Evaluation of execution time

Now let us evaluate the execution time per time step of the real space sum. The number of ions within the cutoff sphere is

$$\frac{4\pi}{3}R^3 n$$

where n is the number density. Each of the N ions interacts with the other ions in the surrounding cutoff sphere, but each pair interaction needs to be considered only once. Thus the execution time is

$$T_R = \frac{1}{2}N\frac{4\pi}{3}R^3 n t_R \qquad (3)$$

where t_R is the execution time to evaluate one interaction.

We also need to work out the time to evaluate the reciprocal space sum. The volume of reciprocal space within the cutoff K is

$$\frac{4\pi}{3}K^3 = \frac{4\pi}{3}\frac{8p^3}{R^3}$$

using (2). The reciprocal space points are given by $\mathbf{k} = \frac{2\pi}{L}(l,m,n)$ where l,m,n are integers, so the volume of reciprocal space per point is $(2\pi/L)^3$. The number of points in the cutoff sphere is thus

$$\frac{4\pi}{3}\frac{8p^3}{R^3}\frac{L^3}{8\pi^3}$$

Writing $L^3 = N/n$ to bring out the N dependance for fixed number density, this becomes

$$\frac{4\pi}{3}\left(\frac{p}{\pi}\right)^3\frac{N}{nR^3}$$

Note that for fixed cutoffs the number of $\mathbf{k}$-points increases as N, because the density of points in reciprocal space increases with system size. In counting the number of points, an additional factor of a half may be included, because of the inversion symmetry of reciprocal space. A sum over the N ions must be performed for each $\mathbf{k}$-point, so the execution time is

$$T_F = \frac{1}{2}\frac{4\pi}{3}\left(\frac{p}{\pi}\right)^3\frac{N^2}{nR^3}t_F \qquad (4)$$

where t_F is the execution time to evaluate one term in the sum.

The total execution time is then

$$T = \frac{1}{2}\frac{4\pi}{3}\left[NnR^3 t_R + \left(\frac{p}{\pi}\right)^3 \frac{N^2}{nR^3} t_F\right] \tag{5}$$

The above derivation is rough and ready. We have considered only the energy, whereas in molecular dynamics one is more interested in the forces. We have not considered the overhead of locating the neighbours in the real space sum; this is discussed further below. We have ignored the factor k^{-2} in the reciprocal space terms, which increases the rate of convergence somewhat. A more thorough analysis can be found in [3], which estimates errors in energy and forces by making reasonable assumptions about the charge distributions beyond the cutoffs. Nevertheless the above argument is adequate to determine the N dependence of the sum when optimised.

Optimisation

Equation (5) shows that, for fixed p and R, T_R varies as N, but T_F varies as N^2, because of the increasing density of points in reciprocal space. Conversely, if we increase R as the system size increases in such a way that R/L is constant, T_R varies as N^2 but T_F varies as N. This suggests that by appropriate choice of parameters we may be able achieve better than N^2 behaviour in the total time.

For a given accuracy, the only free parameter is R, since this determines α and hence K by equations (1) and (2). So to find the value of R which minimises the total execution time we set $dT/dR = 0$. This gives

$$R_{OPT} = \left(\frac{p}{\pi}\right)^{1/2}\left(\frac{t_F}{t_R}\right)^{1/6}\frac{N^{1/6}}{n^{1/3}} \tag{6}$$

Thus the optimal choice of R increases slowly (1/6th power) with N. Substituting in Equation (5) we find for the optimal time

$$T_{OPT} = 2T_R = 2T_F = \frac{4\pi}{3}N^{3/2}\left(\frac{p}{\pi}\right)^{3/2}(t_R t_F)^{1/2} \tag{7}$$

When the total time is optimised it is equally divided between real and reciprocal space parts of the calculation. Equation (7) shows the anticipated $N^{3/2}$ behaviour. We also see that the time depends on the 3/2 power of the precision parameter p, and on the geometric mean of t_R and t_F.

Discussion

To further the discussion, it is useful to introduce the dimensionless parameters.

$$\bar{R} = R/L \tag{8a}$$

$$\bar{\alpha} = \alpha L \tag{8b}$$

$$\bar{K} = KL/2\pi \tag{8c}$$

Then, for precision parameter p, we have from (1) and (2)

$$\bar{\alpha} = p^{1/2}/\bar{R} \tag{9a}$$

$$\bar{K} = p/(\pi\bar{R}) = (p^{1/2}/\pi)\bar{\alpha} \tag{9b}$$

and

$$\bar{R}_{OPT} = \left[\frac{p}{\pi}\right]^{1/2}\left[\frac{t_F}{t_R}\right]^{1/6} N^{-1/6} \tag{9c}$$

The factor of 2π in the definition of $\bar{K}$, (8c), is included since $\bar{K}$ then corresponds to the "integer cutoff" i.e the maximum value of $(l^2+m^2+n^2)^{1/2}$.

The precision required in the Ewald sum depends on the purpose for which it is being used. A lattice energy minimisation program may require greater accuracy than a molecular dynamics program. For the sake of argument let us follow [3] and assume the value $p = \pi^2$. This gives an accuracy of $\exp[-p] = 5.2\times10^{-5}$, which should be adequate for most purposes. With this choice we have, from (9a) and (9b), $\bar{\alpha} = \pi/\bar{R}$ and $\bar{K} = \bar{\alpha}$.

On the basis of operation counts alone we would expect $t_F \approx 2t_R$, though this is obviously very hardware and software dependent. However, we also need to take into account the efficiency with which we can locate neighbours in the real-space part of the calculation. In the case of a solid system, and with the use of a neighbour list, this efficiency will be very high. For a liquid system we obviously need to use an order N neighbour-search algorithm, such as the link-cell method, but this is still not particularly efficient in locating neighbours, unless it is also used in conjunction with a neighbour list. Again, in order to have a concrete example, let us assume $t_F = t_R$. Combining with the suggested value for p we then have

$$\bar{R}_{OPT} = \pi^{1/2}N^{-1/6} \qquad \bar{\alpha} = \bar{K} = \pi^{1/2}N^{1/6}$$

There is a problem in optimising with small systems. In most molecular dynamics programs (but not lattice energy programs) the maximum real-space cutoff which can be used is half the box length. This is because the nearest-image convention is used in locating neighbours in nearby cells. The optimal value of $\bar{R}$ is greater than 0.5 for small systems. With our example parameters it only becomes less than 0.5 when N reaches about 2000. Simulations on smaller systems must therefore use the non-optimal value of 0.5 for $\bar{R}$. With the example parameters this corresponds to $\bar{\alpha} = \bar{K} = 2\pi = 6.28$. These values are close to the values which have been adopted in most simulations to date.

There is another effect to be considered in the simulation of small systems. Conventional link-cell programs use cells whose side is equal to or greater than the cutoff. They only become more efficient than a simple all-pairs neighbour search when there are at least four link cells along each side of the computational box. This implies an $\bar{R}$ of 1/4 or less. With our parameters this condition is reached only when $N \approx 127000$! The situation can be improved by using a variant of the linkcell method which uses smaller cells, as discussed below. However, the fact remains that with smaller systems one may be forced to use a less efficient neighbour search method, or else a smaller and hence non-optimal value of $\bar{R}$.

We may also need to consider memory requirements. Most Ewald sum programs, in their reciprocal space routine, precompute and store factors of the form $\exp(i2\pi l x_i/L)$, which can involve substantial amounts of memory. The number of these factors is proportional to $N\bar{K}$, and hence varies as $N^{7/6}$ when performance is optimised. A compact version of the Ewald sum which computes these factors as they are needed can be several times slower. Furthermore, efficient neighbour location really needs a neighbour list, which again uses a large amount of memory. In this case the number of neighbours of each particle depends on R^3, and the memory requirement scales as $N^{3/2}$.

Size of link cells

In the link-cell method the computational box is divided into sub-cells, and the search for neighbours of particles in a particular cell is limited to those nearby cells whose closest distance of approach to the central cell is within the cutoff distance. In the conventional version, which goes back twenty years to a paper by Hockney, Goel and Eastwood [4], the cells have side equal to R, and the search is limited to the central cell and the 26 touching cells.

This conventional version is not particularly efficient at finding neighbours. The volume searched, $27R^3$ is 6.45 times the actual volume, $\frac{4}{3}\pi R^3$, containing neighbours. The efficiency can be improved by taking more, but smaller, cells so that the volume of those included in the list of nearby cells can more nearly approximate to the cutoff sphere. It is not widely known that this idea also goes back to 1973. Quentrec and Brot [5] had a method in which they took cells so small that they could only include zero or one particles. However, the use of very small cells is not ideal because of the overheads of dealing with very short loops, or empty cells. In experiments to minimise the computer time by varying the link cell side I found that the best size of link cell to use is one which contains about 4 particles. Only after doing these tests did I find another 1973 paper, this time by Schofield [6], which came to exactly the same conclusion. Truly a vintage year for neighbour searching!

Do we need the reciprocal space sum?

Equation (9) shows that, optimally, the "integer cut-off" in reciprocal space increases as $N^{1/6}$, and the number of points within the cutoff therefore increases as $N^{1/2}$. In practice, as we saw in the previous section, for all but the very largest of systems we may wish to reduce the real space cutoff below the optimal value to enable the use of an efficient neighbour search algorithm. This will require a compensating increase in α and hence even more terms in the reciprocal-space sum. The suggestion in [2] that we can omit the reciprocal-space sum in the simulation of large systems thus seems unlikely. However, most simulators have noticed that the reciprocal-space sum is usually small, particularly in the case of large systems, and there are reasons for thinking it may be less important than the above analysis suggests.

The arguments above have been based simply on the magnitudes of the individual terms in the sums, and are therefore valid for arbitrary charge distributions. But the distributions found in molecular simulations have very special characteristics: in particular they observe approximate uniform density and local charge neutrality. It is because of cancellations arising from local charge neutrality that the sum tends to be rather small. In special cases the sum may be very small indeed: this is particularly noticeable with the ideal rocksalt structure, in which the reciprocal space energy is effectively zero for systems larger than 1000 ions. In general it may well be possible to take fewer terms than the general analysis would suggest. If reducing cutoffs to save computer time, the best technique is still to confirm the accuracy of the Ewald sum by checking that energies are independent of α. When doing this, it is essential to use a configuration that is typical of the simulation to be performed: do not rely on reproducing the rocksalt Madelung constant!

It is amusing to note that there is valid argument for dropping the *real-space* term from the Ewald sum. By taking a large enough value of α, and hence a very short cutoff, it is possible to ensure that there are *no* neighbours within the cutoff, because of the finite size of the ions. This is of no practical use in molecular dynamics, because we need a real-space sum anyway for the non-Coulombic interactions.

Example timings

My program for these timing tests creates a configuration by distributing equal numbers of positive and negative charges randomly throughout the box. Though this technique will observe approximate local charge neutrality, the density of course will be far from uniform. This can therefore be regarded as a "worse case" test. The program calculates energies only, not forces. The real space sum uses a link cell method, but based on link cells which have sides which can be chosen to be various submultiples of the cutoff. I chose to test on a system of 4096 ions. Running on an HP 750 workstation I found the value of t_F to be 1.9μs. For t_R, by optimising the link cell side, I achieved a value of 5.2μs. Taking the above values, and $p=\pi^2$, we find

$\bar{R} = 0.377$, and hence $\bar{\alpha} = \bar{K} = 8.33$. The execution time for the program was then 19.6 s, made up of 9.9 s in the real space sum, and 9.7 s in the reciprocal space sum. The link cell side was just over $0.25R$, giving an average of 4.1 particles per cell. The reciprocal space sum provided about 15% of the energy, and certainly cannot be neglected entirely. However, reducing $\bar{K}$ to 6 reduced the reciprocal space time to 3.8 s, and gave an error in the total energy of about 0.05%. This demonstrates that in real problems it may be possible to make considerable savings in computer time, without significant loss in accuracy, by reducing the reciprocal space cut off.

Achieving the real-space times above took some effort. The reason lies in the treatment of the nearest-image transformation. My original code used the popular coding trick

```
rx = rx - anint (rx)
```

etc., but this gave an execution time of 61.6 s. Changing this to

```
rx = rx - nint (rx)
```

and so allowing the compiler rather than the intrinsic function to convert from integer to real, reduced the time to 33.7 s, and almost the same time resulted if the "nint trick" was abandoned and a double IF statement used instead. The much faster time of 9.9 s was obtained by using the link cell information to determine when the nearest-image transformation needs to be applied. The "nint trick" works better on other computers.

By removing the nearest-image transformation and spherical cutoff I was able to determine that the actual evaluation time for a single real-space interaction is about 4.0μs. This shows that the neighbour search and nearest-image transformation still impose a 30% overhead.

REFERENCES

[1] J.W. Perram, H.G. Petersen and S.W. de Leeuw, "An algorithm for the simulation of condensed matter which grows as the 3/2 power of the number of particles", *Molec. Phys.* 65 (1988) 875-893.

[2] Z.A. Rycerz, "Calculation of the Coulomb interactions in condensed matter simulations", *Molec. Simulation* 9 (1992) 327-349.

[3] J. Kolafa and J.W. Perram, "Cutoff errors in the Ewald summation formulae for point charge systems", *Molec. Simulation* 9 (1992) 351-368.

[4] R.W. Hockney, S.P. Goel and J.W. Eastwood, "A 10000 particle molecular dynamics model with long range forces", *Chem. Phys. Lett.* 21 (1973) 589.

[5] B. Quentrec and C. Brot, "New method for searching for neighbours in molecular

dynamics computations", *J. Computat. Phys.* 13 (1973) 430.

[6] P. Schofield,, "Computer simulation studies of the liquid state", *Comput. Phys. Commun.* 5 (1973) 17.

MOLECULAR DYNAMICS FOR REAL TIME MACRO-WORLD PHENOMENA ANIMATION

Rafał Wcisło[1], Witold Dzwinel[1], Jacek Kitowski[1,2]
and
Jacek Mościński[1,2]
✉ [1]Institute of Computer Science AGH, 30-059 Kraków, POLAND
[2]Academic Computer Centre CYFRONET, 30-950 Kraków, POLAND
✆ *dzwinel@uci.agh.edu.pl*

ABSTRACT

Usefulness of the molecular dynamics method for real world and real time animation is discussed. As an example, the falling sheet simulation is examined to show both the most demonstrative advantages of the method proposed and problems encountered. Though the quality of the simulation results seems to be sufficient, the real time animation involves additional effort.

1. INTRODUCTION

Molecular dynamics (MD) technique is a widely used numerical tool for microscopic phenomena simulations. Physical scale limitation is the effect of considering a single MD item as a model of more or less elementary physical object (a particle, atom, molecule) with its principal features determined by the time and space scale and the way of particles interactions. Such an approach leads to irresolvable problem of the real world approximation by ever and again increasing N - number of simulated particles. For 3-D simulation, l-proportional increase of time and space scale (for the simplest case when the short range interactions are considered only) involves l^4 increase of CPU time and l^3 desirable computer memory. However, the results concerning micro-hydrodynamic flows simulations [1,2] and Rayleigh-Benárd convection [3,4] show that for N of order 10^4 particles, the macro-scale effects can be observed in micro-world. Although strikingly realistic qualitative patterns were obtained, the quantitative results often significantly differ from those obtained in macro-scale. Nevertheless, we can hope that some of them can be improved considerably for higher N but still tens of orders of magnitude less than the Boltzmann limit.

The facts mentioned above can bring to the conclusion that by simple rescaling of the Newtonian equations of motion to the dimensionless units or to the macroscopic time and space scale with simultaneous redefinition of the semi-particle and the semi-interaction potential, the MD method could be used as the numerical tool for macro-world simulation. It is especially interesting in the case of lack or incompleteness of other approaches. MD method could be competitive with either the lattice gas simulation, giving additional possibilities lacking for this technique, or with the classical models of continuous mechanics described by partial differential equations (PDE) irresolvable for many physical problems. The MD simulation results for the penetration mechanism presented in [5] can serve as an example that selected macroscopic problems exist which can be solved using N-body model governed by the Newton's laws of motion using relatively small computer resources. However, for the real world animation in the real time, the computational time is still to long.

In this paper we present the simulation results of the scenes of: the single sheet falling in the gravitational field, picking up from the table and tearing up. The timings are presented and the ways of speeding up the computations (both just applied and proposed) are reported. This not trivial example shows that MD method not only can be used as a numerical tool for the macroworld phenomena simulation but it could be used in the nearest future to the real world phenomena in the real time animation. The underlying principles of such the way of animation, contrary to others widely used computer animation, come directly from physics and are not limited by the recorded set of scenes and artificial rules. Thus such an approach better approximates the real world.

2. PROBLEM PRESENTATION

To simulate the single sheet behaviour let us assume that it can be modelled by the mesh of N=N^2 points with the masses m=M/N (where M is the mass of the sheet). Let us assume additionally that the single point interacts with its closest neighbours only. To ensure the proper material stiffness we choose the following i,j particle-particle potential (see [6]):

$$V_{i,j} = \frac{1}{4} \cdot k (r_{i,j}^2 - a_{i,j}^2)^2 \tag{1}$$

where:

k - stiffness factor,

$r_{i,j}$ - particle-particle distance,

$a_{i,j}$ - the characteristic mesh distance to the closest neighbours at the beginning of simulation.

Two sorts of meshes are considered (see Fig. 1):

- rectangular - each of points has four closest neighbours with $a_{i,j}=1$ and four with $a_{i,j}=\sqrt{2}$ on diagonals,
- hexagonal - each of points has six equally spaced neighbours with $a_{i,j}=1$.

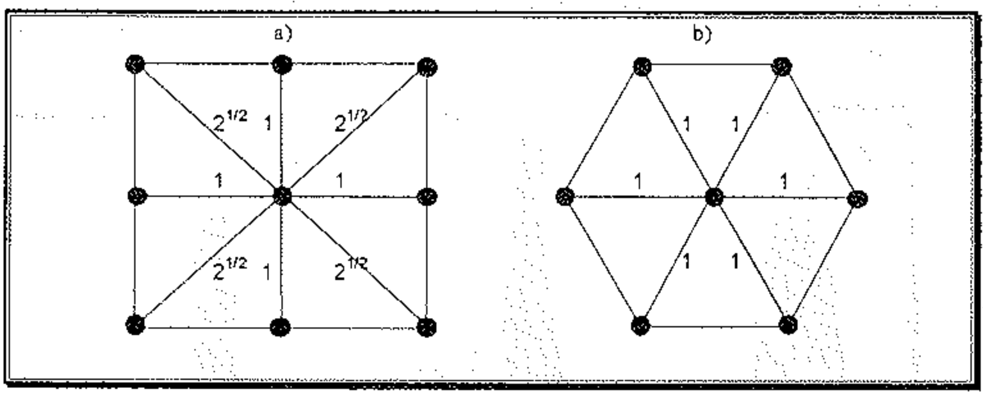


Fig.1

The proposed potential fits better for simulation purposes than, for example, harmonic one. It ensures initial stability of the system (especially for the rectangular mesh) and is more efficient since the square root has not to be computed for the forces evaluation. To reduce the numerical oscillation, to use the greater timesteps and to render better the real physical conditions the friction force

$$\mathbf{F}_i = -\lambda \cdot \mathbf{v}_i \tag{2}$$

is introduced. The leap-frog scheme for the sheet in the gravitational field (which can be turn on or off optionally), namely

$$\mathbf{v}_i^{n+1/2} = \frac{(1-\varphi)}{(1+\varphi)} \cdot \mathbf{v}_i^{n-1/2} + \frac{\alpha \Delta t}{(1+\varphi)} \cdot \left\{ \sum_{j=1}^{K} ({r_{i,j}^n}^2 - a_{i,j}^2)\mathbf{r}_{i,j}^n + \frac{g}{\alpha}\mathbf{i}_z \right\} \tag{3}$$

$$\mathbf{r}_i^{n+1} = \mathbf{r}_i^n + \mathbf{v}_i^{n+1/2} \cdot \Delta t \tag{4}$$

where:

$$\alpha = \frac{k}{m}, \quad \varphi = \frac{\lambda}{2m} \cdot \Delta t \leq 1, \tag{5}$$

and K-the number of the closest neighbours, is applied. The timestep Δt can be optimized using the relations (3-5).

Assume that the size of the sheet L×L is constant. On one hand, l-times of N increase (what is equivalent to decreasing of the nodes spacing Δx=L/N) can improve the granularity of the system. On the other, both N increase and Δt decrease (the timestep is proportional to the nodes spacing divided by the maximal velocity in the system) causes the execution time increase as l^3. Too high granulation produces also the excessive folding of the sheet. The non-zero wrinkle resistance could be simulated modifying the sort and the range of nodes interactions or assuming more than one particles layer. All of these, however, lead to additional decrease of computational efficiency, deteriorating animation effect. The selection of appropriate number of nodes giving not too low Δx, seems to be the best issue.

In course of simulation the initially plain sheet shape changes considerably. Taking into account only the fixed closest neighbours, the self-crossing effect is impossible to avoid. It can be partially reduced by the proper (not to low) Δx selection. However, the procedure eliminating un-physical self- crossing of the sheet must be used in the course of simulation. In Fig.2b the effect of self-crossing is shown using as an example a ribbon falling in the gravitational field. Turning on the "reflecting" procedure lets to avoid crossing (see Fig.2c) but it lowers, of course, computational efficiency. Therefore, it must be thoroughly optimized like the procedure of the sheet dynamics computations.

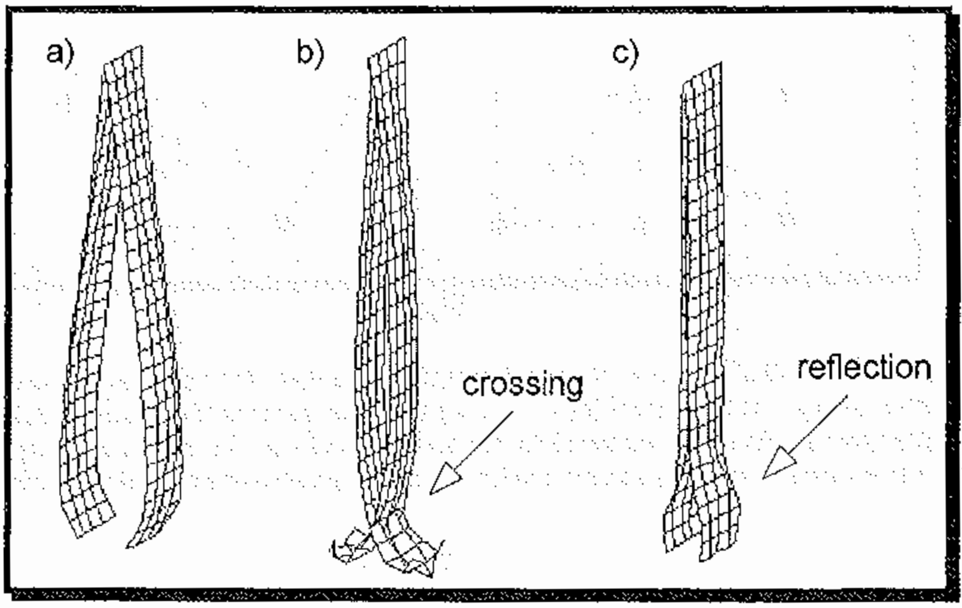


Fig.2. Falling ribbon in the absence (b) and presence (c) of the procedure eliminating self-crossing effect.

The procedure for self-crossing effect elimination bases on the linked list conception [7]. The projection onto the plain, parallel to the sheet initial position, is divided onto the square cells (see Fig.3). The *CellSize*=$\Delta x + \varepsilon$, i.e., is assumed slightly greater than the basic mesh spacing Δx (ε can be greater for less rigid, elastic material).

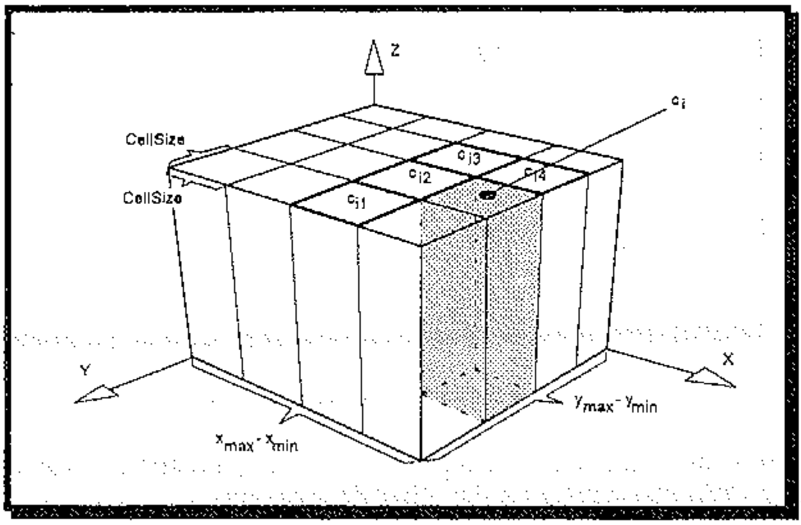


Fig.3 Computational box and cells assumed for procedure of self-crossing elimination. $\Delta\eta=\eta_{max}-\eta_{min}$, where $\eta\in(x,y)$, are evaluated every timestep in accordance with actual position of the sheet.

If in a neighbouring cell of c_i, i.e., in one of ($c_{i1},c_{i2},c_{i3},c_{i4}$), appears the "alien" node, i.e., the node which does not belong to the closest neighbours of the nodes being the members of c_i, it deflects from the nearest node (the member of c_i) in accordance with the rule presented in Fig.4, provided that the "alien"-member nodes are closer than assumed distance $\mathbf{R}_{ref}$.

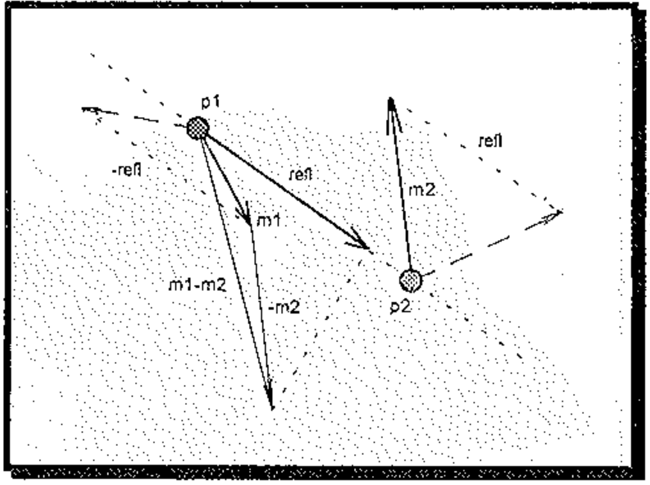


Fig.4 The reflection rule for two nodes **p1** and **p2** with the respective initial momenta **m1** and **m2**. The dashed lines represent the momenta after deflection.

To enable the wide range of realistic events simulation the more extreme disruption of the sheet shape can be realised. If the distance between two of the neighbouring nodes is greater then

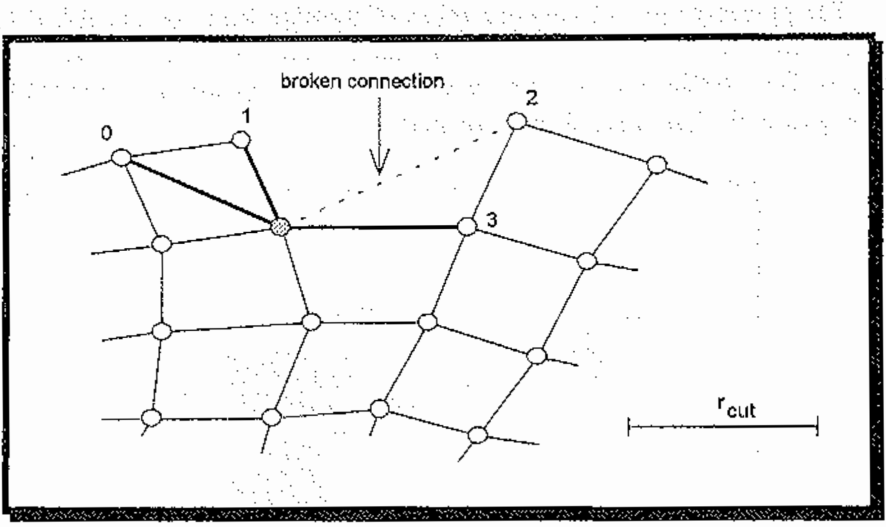


Fig.5 Diagram of tearing up the sheet .

r_{cut} (see Fig.5) the connection is broken and in course of simulation such nodes are treated as the alien ones, i.e., they can interact via the reflection rules only.

The results and problems with the computer realisation of the model presented are shown and discussed below.

3. RESULTS AND EXPERIMENTS

Animation were carried out for meshes with N=20 and 25 (i.e. 20×20 and 25×25 nodes). The selected snapshots of a few animations are shown in Figs.6-8. It seems that they represent adequately the real physical motion.

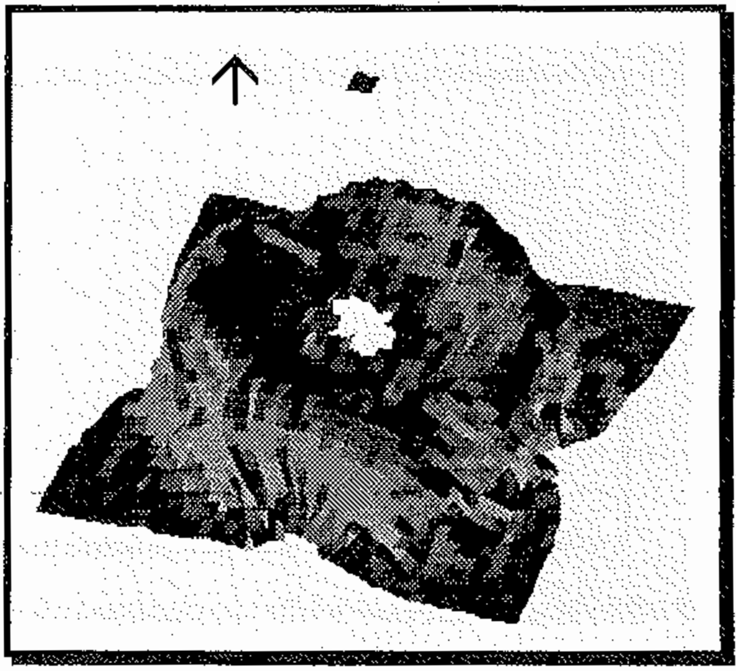


Fig.6. One of the final phases of simulation of the squared (0.5×0.5 m) sheet picking up rapidly (v=5m/s, the arrow shows the velocity direction) from the table in the presence of gravity. Leaky sheet falls down while the torn piece of material is drawn in velocity vector direction.

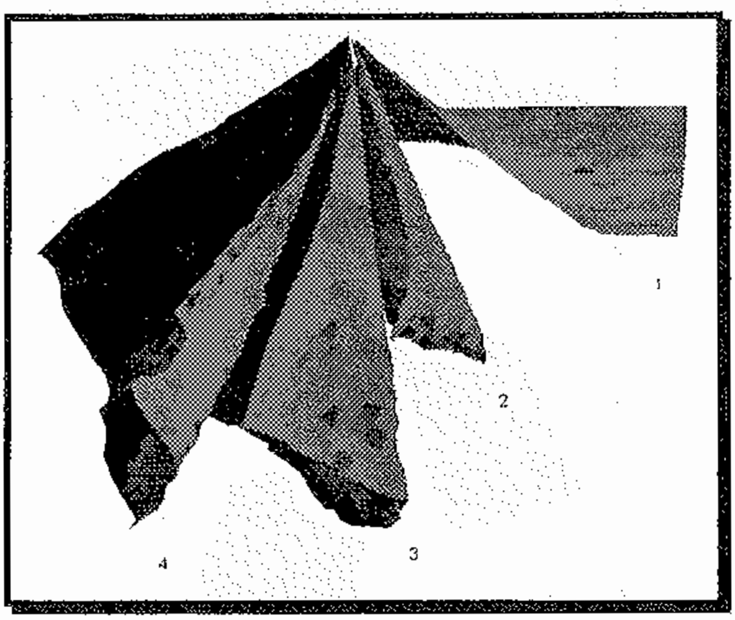


Fig.7. Falling sheet animation snapshots.

For more realistic pictures a source of light and shading are applied. For the model granulation assumed, the fluent motion can be observed using SUN SPARCstation2 GX. However, taking into account that the real time each of the physical experiment is about 2 seconds, the wall clock time is still to high. Let us compare this figure with these comprised in the second column of the Table 1 presenting the timings of computer experiment for the sequential C code.

Table 1

The timings for computer animation of the experiment shown in Fig.7, representing the real time behaviour of about 2 seconds.

Number of nodes	Wall clock time for sequential code [s]	Wall clock time for "two segments" distributed code [s]	*speedup*
10 x 10	8	6.5	1.23
15 x 15	11	8	1.37
20 x 20	15	11	1.36
25 x 25	22	14	1.57
30 x 30	29	18	1.61
35 x 35	39	23	1.69
40 x 40	51	30	1.7

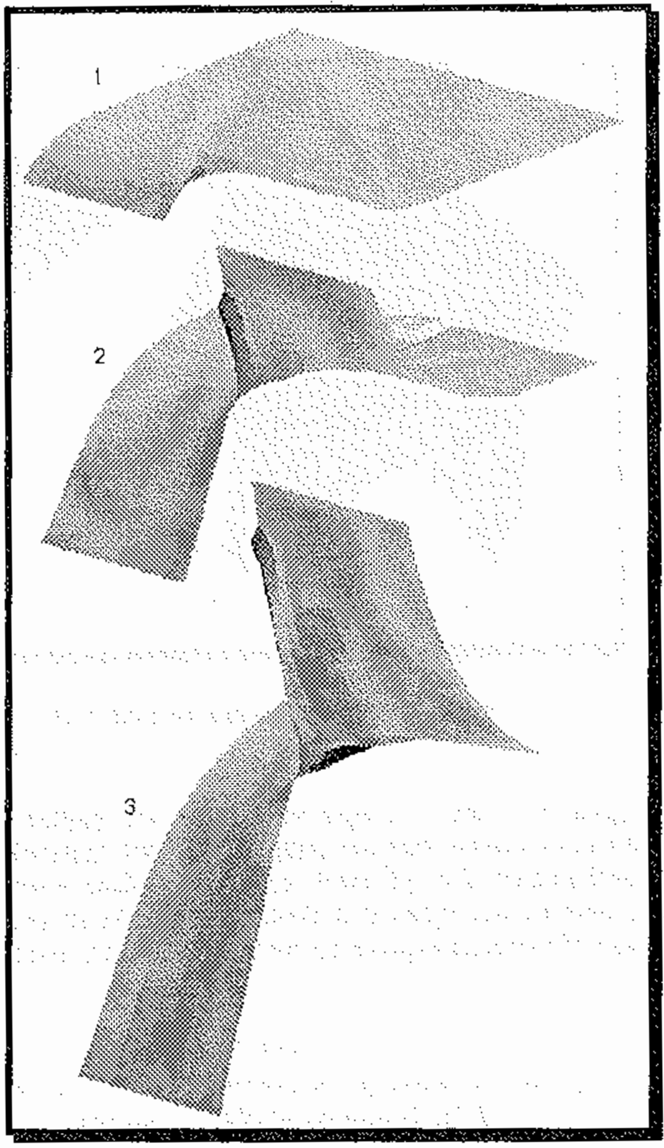


Fig. 8 The snapshots of tearing up the sheet (with gravitation turned off).

Fig.9 shows the proportions of the computational time for each part of the program for the granulation distinguished in Table.1. It was assumed additionally that each picture is generated every L timesteps giving the impression of fluent motion (L=20 for the distinguished case). To obtain the desired speedup (about 8 times for assumed case) the authors tried to use the distributed computations approach for the problem under consideration.

For this purpose two sorts of parallelism were applied:

- algorithmic distribution - the processes of pictures generation and calculations should be disjoined and performed concurrently,
- geometrical parallelisation of MD algorithm.

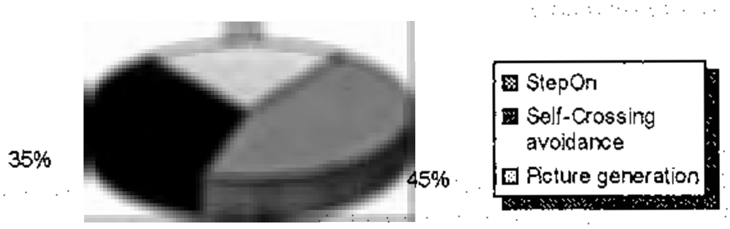


Fig.9 The proportions of the computational time for each part of the program, for the granulation distinguished in Table.1.

Only the first sort of parallelism gave positive result. As shown in Table.1, using one workstation for pictures generation from the data transmitted by the another one where calculations were made, gave maximum speed up. However, it was estimated and approved by computations [8] that for geometrical partition of computational object like in [9] the speed up less than 1 can be expected (for the Ethernet protocol). For this negative result the communication time is responsible mainly. Information exchange is almost 100 times greater than those for the first case.

Because the system consists of a relatively small number of nodes (particles) the problem parallelisation on the stage of single timestep is ill conditioned and the load balancing factor for distributed computing can not be considered. For the animation purposes the main result is the pictures generation. On the other hand, however, to obtain a single picture, L timesteps have to be performed. Therefore, for relatively high L (>10) the geometrical parallelisation of computational box represent the lowest level of parallelism (fine grain) which is unsuitable for distributed computing approach. From distributed computing point of view, such a problem is rather sequential (in time) what was approved by the results obtained. The problem could be resolved using another (tightly coupled) parallel architecture or vector processing. The relatively short timestep which have to be used to ensure the simulation stability is the main drawback of MD application for animation. So that, in our opinion, the crucial point is to resolve just this problem, i.e., to find the efficient algorithm for nodes (particles) motion, or from the distributed computing point of view; substitution the sequential in L timesteps algorithm of nodes motion for a parallel one.

4. SUMMARY

The main conclusions can be summarised as follows.

1. The MD paradigm can be used for some macroscopic phenomena simulation giving realistic results. For many purposes it can replace the continuous PDE (partial differential equation) phenomena description especially in such cases, in which deterministic solutions are highly disturbed by chaotic motion [1-4], discontinuities appear [5], mathematical description does not exist or the MD model is more flexible than the other one.
2. MD method constitutes an interesting alternative as a technique for real time animations purposes.
3. The relatively short timestep which has to be assumed to preserve simulation stability, interfere with high computational efficiency. The necessity to perform L>10 timesteps to generate a single animation snapshot makes, from distributed computing point of view, the algorithm sequential and difficult to parallelise for the required computation to communication ratio level (low for Ethernet protocol).

The authors would be very grateful for any comments and suggestions concerning the topic and welcome any proposition and examples of MD application as the numerical technique for macroscopic world simulation.

REFERENCES

[1] Rapaport, D., C., "Microscale hydrodynamics: Discrete-particle simulation of evolving flow patterns", *Phys. Rev.*, **A36**, 7, 3288 (1987).

[2] Rapaport, D., C., "Unpredictable convection in a small box: Molecular--Dynamics experiments", *Phys. Rev*, **A46**, 4, 1971 (1992).

[3] Rapaport, D., C., "Molecular--Dynamics Study of Rayleigh-Benárd Convection", *Phys. Rev. Let.*, **60**, 24, 2480 (1988).

[4] Puhl, A., Monsour, M., M., Mareschal, M., "Quantitative comparison of molecular dynamics with hydrodynamics in Rayleigh-Benárd convection", *Phys. Rev.*, **A40**, 4, 1999 (1989).

[5] Alda, W., Dzwinel, W., Kitowski, J., Mościński, J. and Yuen, D., A., "Penetration Mechanics via Molecular Dynamics", *Reports of Army High Performance Computing Research Center*, **93-037**, University of Minnesota, (1993).

[6] Phillipson, P., E., Armstrong, A., J., "Map Dynamics and Newton's Second Law", *Physics Letters A*, **146**, 403, (1990).

[7] Hockney, R., W., Eastwood, J., W., "Computer Simulation Using Particles", Mc Graw-Hill, New York (1981).

[8] Wcisło, R., "An Example of Molecular Simulation Application for Macroscopic Objects Animation", M.Sc. Thesis, Institute of Computer Science, Dept. of Electrical Eng., The Stanisław Staszic University of Mining and Metallurgy, Kraków (1993) (in Polish).

[9] Bruge, F., Fornili, S., L., "A Distributed Dynamic Load Balancer and its Implementation on Multi-Transputer System for Molecular Dynamics Simulation", *Comput.Phys. Commun.*, **60**, 39, (1990)

[10] Smith, W., "Fortran code for the link-cell method", *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, **30**, 35 (1986).

Using PICASSO for Visualisation of MD Results

J. Dlugopolski[1], W. Alda[1], J. Kitowski[1,2]

[1]Institute of Computer Science, AGH, al. Mickiewicza 30, 30-059 Cracow/Poland

[2]Academic Computer Centre CYFRONET, ul. Nawojki 11, 30-950 Cracow/Poland

e-mail: kito@uci.agh.edu.pl

Abstract

In this report we present the usage of PICASSO visualisation system to molecular dynamics fluid flow simulations. Conversion program from local MD file format to PICASSO file format is briefly described and sample images are presented.

1 Introduction

The role of visualisation of results obtained in simulations increases all the time. Problem arises when the computer experimentator has to choose the visualisation tool. He may create his own program, well suited to the problem, but it needs relatively a lot of work. Often more reasonable is to take existing tool and cut it to one's needs. There is a number of different visualisation tools available (e.g. [1], [2], [3]). In this paper we report our attempts in using PICASSO [4], advanced visualisation program obtained from the author, Dr. Marc Westermann (ETH Zurich).

PICASSO (Painting Interactively Colored and Shaded Simulation Objects) is object oriented interactive graphic program designed mainly for visualisation of results from simulation of semiconductor materials. It allows visualisation of two and three dimensional scalar fields using iso-lines, shaded iso-lines and colour palettes. Vector fields can be displayed with arrows of different length and/or colour. Scalar and vector fields can be displayed simultaneously. It is possible to zoom images, scale and rotate, change view point and direction of observation, add description text, prepare hardcopy PostScript files, make animations (using additional MOVIE program) and many more.

PICASSO works in UNIX X-Windows environment and uses its own Graphic User Interface – DOGMA [5].

Input files for PICASSO should be prepared in DATEX [6] format, in text files. There are two basic DATEX files: first describing geometry of the simulated system (e.g. shape of grid, number of nodes) and second containing the simulation data itself.

PICASSO expects to have its data on a grid and shows them as continuous values. In molecular dynamics, as the number of particles rises, displaying individual particles becomes impossible, and using such tools as PICASSO is very reasonable.

2 Conversion of binary MD format to DATEX format. b2d program

To enable using PICASSO with MD data, we have added a program for data interpretation and conversion to DATEX format. b2d program has been designed for particular local format of MD output file, used by our simulation programs. It is also possible to generate sequences of images to enable animation by MOVIE program. Work to make b2d more general is in progress.

There are three basic interpretation methods (modes) available in b2d.

1. The first (most simple) method is representing every particle by distinct item in DATEX format and by distinct dot in the picture. Additionaly, arrows showing momenta or forces may be displayed. No geometrical information is prepared in this mode, so a lot of PICASSO capabilities is not available. An example representing particles momenta in this mode of visualisation is shown in Fig. 1. The example concerns a test simulation of two-kind Lennard-Jones particles system. Most of the particles are similar to Argon particles, while in the centre the particles with lower σ are introduced. Since the initial distances between the particles are the same, this implies instability in the centre of the system which develops in time roughly similar to the implosion effect.

2. The second method is based on building two dimensional rectangular grid. Every node of the grid represents particles placed in the cell around it, keeping two values: number of particles in the cell (number density) and average momentum or force over the cell. b2d allows the choice of arbitrary grid dimension and possibility of converting a sequence of images at a time. An example is shown in Fig. 2. In this example we present results of MD applications to qualitative studies of the penetration mechanism [7], in which both – the hitting object and the wall are simulated with the Lennard-Jones particles. The figure presents just the transient of the penetration, showing the momenta distribution in arbitrary units.

3. Next method is an enhancement of the previous one. The cell, as described in previous item is treated as the basis of pyramid. Height of the pyramid is proportional to the modulus of vector value (momentum or force) connected with the cell or to the numerical density. Colour palette added may improve the image. In Fig. 3 this mode is applied for presentation of force distribution from the simulation similar to one discussed in the previous item.

b2d program can be run in two modes: command line and as X-Window utility (sample window is shown in Fig.4). General form of command line calling sequence is as follows:

b2d [-h] [-v] [-kx] [-px1[,x2]] [-dx] [-ax] [-t] [-c] prefix

Options have the following meaning:

prefix This is the prefix of input file names, one for files from one experiment. May be treated as a name of experiment.

-h Displays information about all options.

-v Displays information about input files Sample calling sequence: **b2d -v prefix**

-kx Chooses conversion mode.

-px1[,x2] Chooses timestep (or sequence of timesteps) for conversion. E.g. **-p3,12** means conversion of images from timestep 3 up to 12.

-dx Chooses grid dimension. **-d100** means 100x100 grid.

-ax Path to input files, e.g. **-a/usr/home/files/**.

-t Switches to command line mode when used (X-Window mode otherwise).

-c Compress output files (using standard UNIX *compress* program).

Here is a complete example of calling sequence:

```
b2d -t -k2 -d100 -p17 -a/symul/pliki TEST
```

This will run b2d in command line mode, with input binary files **TESTpos.bin** (coordinates) and **TESTmom.bin** (momenta) placed in /symul/pliki. Data from seventeenth timestep will be read.

3 Conclusions

We have found PICASSO very powerful, flexible and easy to use program. With conversion program it became a very useful tool for visualisation of two dimensional MD simulations. We found it particulary convenient for examining density fluctuations and stress changes in our fluid flow simulations [7]. At the moment b2d program is fitted for one input format only, so its usage is limited to rather narrow group of users. However changes adopting it to other formats can be made fairly easy. We continue development of b2d by supplying it with interactively defined user input format.

References

1. Scientific Visualization Tools and Libraries at CRS4, Centre for Advanced Studies, Research and Development in Sardinia, May 1992.

2. Software Tools Group, National Center for Supercomputing Aplications, University of Illinois at Urbana-Champaign.

3. PAW - Physics Analysis Workstation, The Complete Reference, version 1.14 (July 1992), CERN Program Library Long Writeup Q121, CERN Geneva.

4. M. Westermann, "Picasso 2.1 User's Guide", ETH Eidgenossische Technische Hochschule, Zurich 1992.

5. M. Westermann, "DOGMA - Direct Output for Graphic Modeling Application", ETH-Zurich, Juni 1991.

6. ——, "DATEX 2.1 - Data Exchange Format" (Semiconductor Simulation Group), ETH Eidgenossische Technische Hochschule, Zurich 1992.

7. W. Alda, W. Dzwinel, J. Kitowski, J. Mościński and D. A. Yuen, "Penetration Mechanics via Molecular Dynamics", University of Minnesota Supercomputer Research Report UMSI 93/58, April 1993.

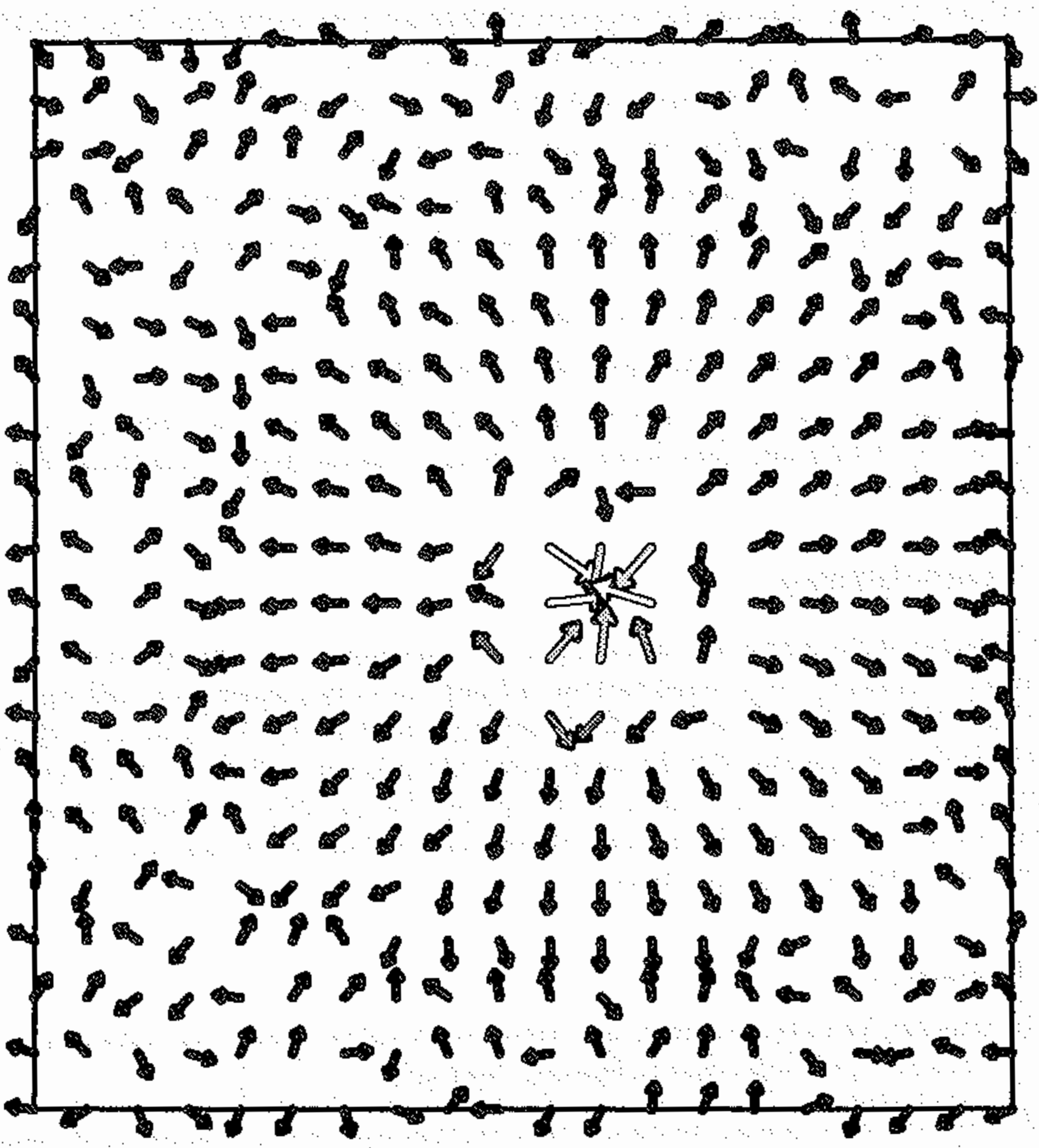


Figure 1: Example of visualisation using the first mode (with particles momenta). Shown instability caused by attraction in the centre.

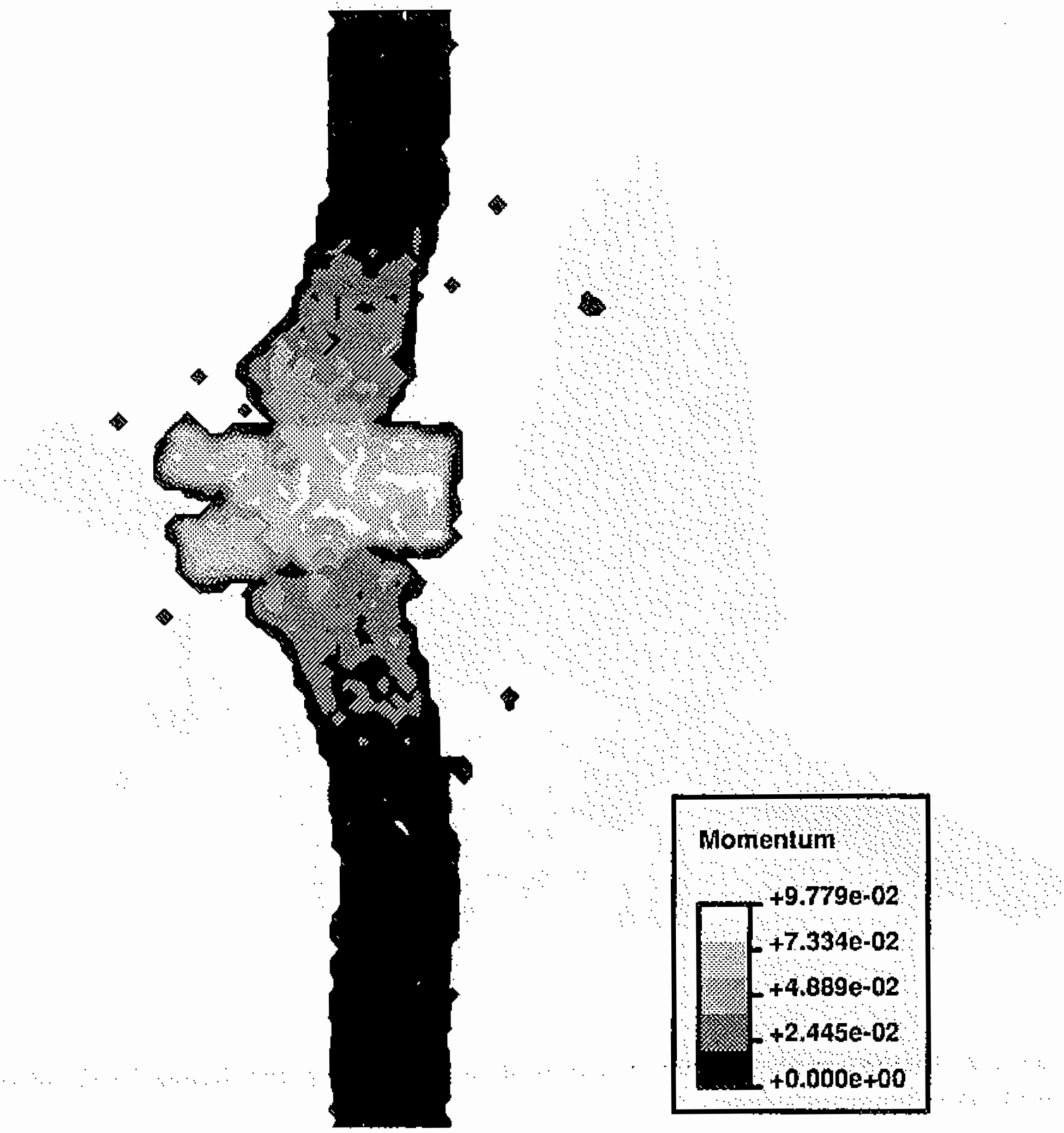


Figure 2: Example of visualisation using the second mode. The object penetrates through the wall. Momenta distribution in arbitrary units.

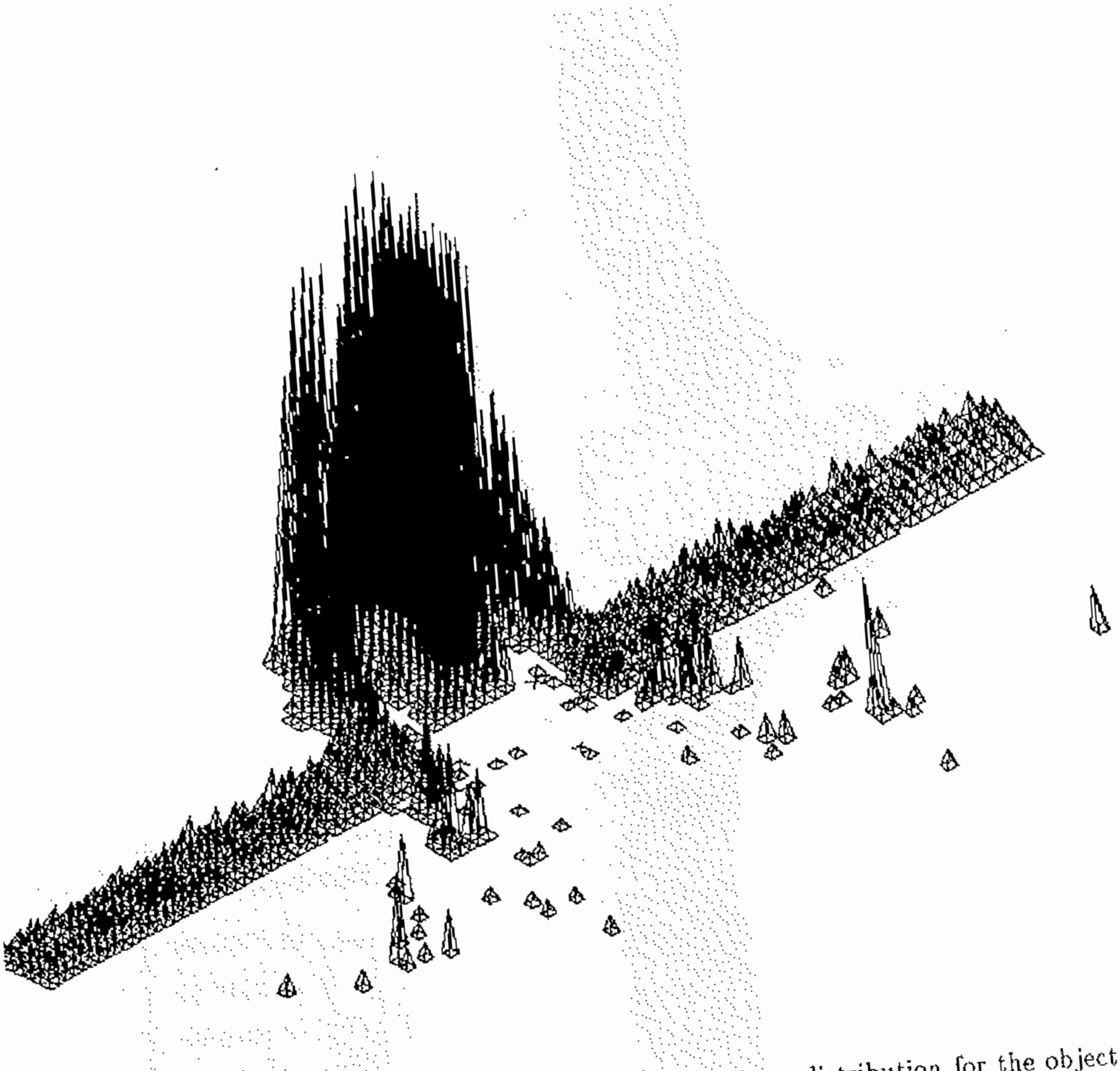
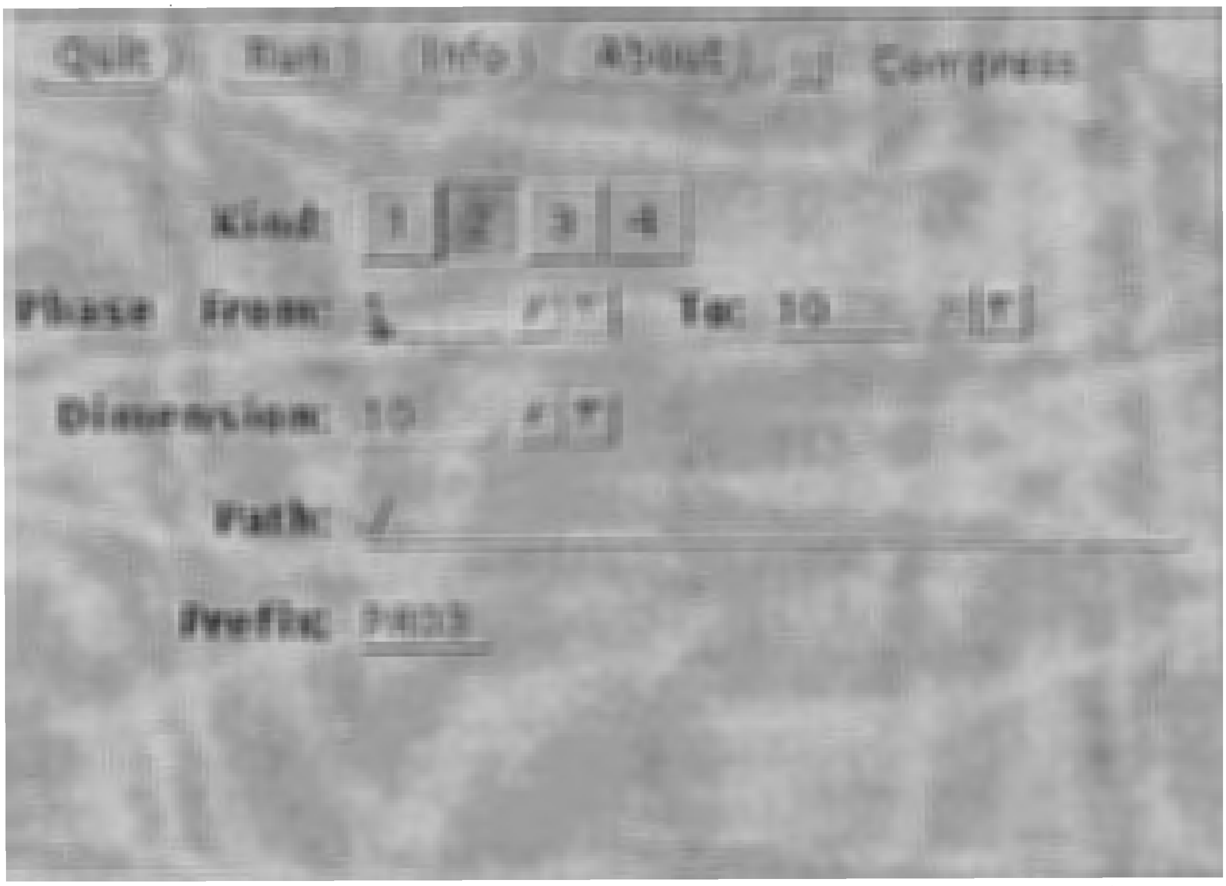


Figure 3: Example of visualisation using the third mode. Force distribution for the object penetration through the wall.



Kind: 1 2 3 4

Phase From: 1 To: 10

Dimension: 10

Path: /

Prefix: [illegible]

Figure 4: B2D in X-Windows environment

XEDS - X Windows Tools for input data edition and result visualization in molecular simulation

Darin Nikolow*, Witold Alda*, Jacek Kitowski*†
* Institute of Computer Science, AGH, al.Mickiewicza 30, 30-059 Cracow, Poland
† Academic Computer Centre CYFRONET, ul.Nawojki 11, 30-950 Cracow
e-mail: kito@uci.agh.edu.pl

Program Summary

Title of the program: XEDS
Computers: Sun SPARC Station 2, IBM RS/6000-320,
HP Apollo 9000/720, Convex C3210
Installation: Institute of Computer Science, Academy of Mining and
Metallurgy and Academic Computer Center CYFRONET,
Cracow, Poland
Operating system: SUN OS 4.1, AIX 3.1, HP-UX 8.05, Convex OS 10.0
Programming language used: C [1]
Number of lines in programs: about 3400
Keywords: Visualization, X11, molecular simulation

Nature of the problem

XEDS consists of two software tools respectively for graphical presentation of results from molecular simulation, `xout`, and graphical input data edition for the simulation - `xlay`. The first tool, `xout`, can display the positions of the particles, their momenta or velocities at certain moments of the simulation. Animation is possible, as far as the positions are concerned, with different length of traces, so one can better follow the trajectories. By using the second tool, `xlay`, one can graphically edit or create the input data. Possibilities for line-drawing and enclosed-area-filling are added to make the editing easier.

Method of solution

The positions or momenta of particles are read from a file and then the current picture is drawn. All options are accessible through button-pressing or using other X Windows gadgets. The graphic library used is the standard X11 library. Any X Snapshot-like applications can be used to create a hardcopy of the picture.

The program is written in the C language.

Restrictions of the complexity of the problem:

The program is appropriate for 2-D molecular simulation and is suitable for results presentation of non-equilibrium simulation for relatively large number (up to 150000) of particles. The tools can be started only on color screen X server.

[1] A workstation superset of [1]

Typical running time:

The program runs in the interactive mode and the time of presentation of particle positions depends on a number of particles and steps. On SPARC station2 GX, for 37121 particles and 31 recorded steps obtained in 2-D molecular dynamics simulations the whole animation takes about 25 s, in which 65% is taken for graphical displaying and image computing.

1 Introduction

Computer simulations as a rule produce an enormous amount of information, so methods and utility programs are necessary in order to extract essential features of a process.

XEDS is a specialized computer graphics system for graphical presentation of results from 2-D molecular dynamics. It is similar to EDSIMP [2] for IBM PC, and to sEDS [3] for Suns under SunView, but it runs in X window environment and is more developed than those. XEDS is written in the C language and uses graphics functions from the X11 library [4] and Athena Widget Set [5] for managing the gadgets (buttons, scrollbars etc.).

The user of the system can easy create the desired layout of particles for the molecular simulation (i.e. the initial particle distribution). This function is particularly useful when the computational box is to be inhomogeneously filled with the particles of different kinds. There can be placed a single particle of a chosen-before kind, drawn a line or rectangle of particles or filled an enclosed empty area with particles. Finally the layout is saved in a file which is to be used by the simulation program. When the simulation is over the graphical presentation of the results can be performed using this system. The data can be visualized step by step or animated. Zooming is possible for observing an area closer. When animating the trace length is to be chosen.

The data file format used by `XEDS` was pre-defined by the simulation program which already existed. The input/output files contain binary data of different variables and arrays, usually doubles, used by the programs. Because the purpose of the visualization tools created was to represent graphically the results from MD simulations program no precautions were taken to stick to some standard format like NCSA HDF [6] for instance. However, the effort for using standard formats for simulation programs and XEDS will be undertaken in close future. The tools are extremely useful for primary selecting the results of the MD simulation experiments and to choose the best results that are supposed to be proceeded by more sophisticated programs like `PICASSO` [7] adopted for MD result visualization [8], which uses the DATEX format.

2 Description of XEDS operations

2.1 Edit the input data using `xlay`

There are three stages:

- reading the input data,
- editing the layout using the mouse,
- writing data.

The second stage is a very simple layout editor. The user can place a single particle or draw a line of particles. Particles are initially places on, what we call '2-D fcc mesh'. Node sites of the mesh may be either occupied by a particle or empty. When there is an empty area surrounded by particles it can be filled up using the fill function.

Two files are used to keep the input data:

1. text file containing parameters: `????inp.txt` (created by the user, using a text editor). An example of such file is shown in appendix A.

2. binary file containing positions: `????lay.bin` (created and/or modified by `xlay`, to be used by the simulation program).

The file `????lay.bin` has the following structure:

- **Layout[].X** — array of `Nparts` 8-byte doubles keeping the x-coordinates of the particles,
- **Layout[].Y** — array of `Nparts` 8-byte doubles keeping the y-coordinates of the particles,
- **Layout[].KindSp** — array of `Nparts` 4-byte integers keeping the kinds of the particles,
- **Ncx** — 4-byte integer keeping the number of cells in x-direction,
- **Ncy** — 4-byte integer keeping the number of cells in y-direction,
- **Nofccx** — 4-byte integer keeping the number of fcc cells in x-direction,
- **Nofccy** — 4-byte integer keeping the number of fcc cells in y-direction,
- **vp[]** — array of `Nparts` 4-byte integers keeping the indexes of the vertices in fcc-lattice, which are occupied by particles.

2.2 Graphical presentation of the results using `xout`

This part runs in the following stages:

- setting of the form of graphical presentation,
- displaying the desired presentation.

In the first stage the user chooses a mode of the graphical presentation: positions, momenta or velocities can be displayed. There is also scaling provided for better seeing if the density is too high and the particles cannot be distinguished. Other menu items are: a step to go, trajectory length for the animation, density histograms on x or y axes, number of intervals for these histograms (up to 300).

In the second stage the user observes the results step by step or animated. The modes can be changed at any time. The program can run only on color screens.

The animation changes the pictures smoothly because of the double buffering used. The graphical image is first drawn in the memory and then it is copied into the screen memory with a fast raster-copy operation. This prevents unpleasant blinking of the pictures.

The user can choose kind of particles to be displayed with the toggle gadget called 'Drawing parts'. Any combination of maximum eight kinds is possible.

There are two input files used:

1. binary file containing positions,

2. binary file containing momenta.

These files are created by the simulation program and it should be stressed that no other files (e.g. with raster images) are needed. In the description below `Nparts` refers to number of particles and `Nspec` to number of kinds of species. Both files contain the following information:

- **header** — array of 800 bytes for comments,
- **Nparts** — 4-byte integer keeping the number of particles taking part in the simulation,
- **Nspec** — 4-byte integer, which give us the number of kinds of particles,
- **Nparsp[]** — array of `Nspec` × 4 bytes keeping the number of particles for each kind,
- **Kindsp[]** — array of `Nparts` × 4 bytes keeping the kind for each particle,
- **Ncx** — 4-byte integer; this is the number of fcc cells in x-direction,
- **Ncy** — 4-byte integer; this is the number of fcc cells in y-direction,
- **Dt** — 8-byte double; this is the simulation timestep,
- **Field** — 8-byte double; (not used by **xout**),
- **Fldbox** — 8-byte double; (not used by **xout**),
- **Dummy** — 8-byte double; (not used by **xout**),
- **Ul** — 8-byte double; this is the unit used for describing the position of a particle or the momentum.

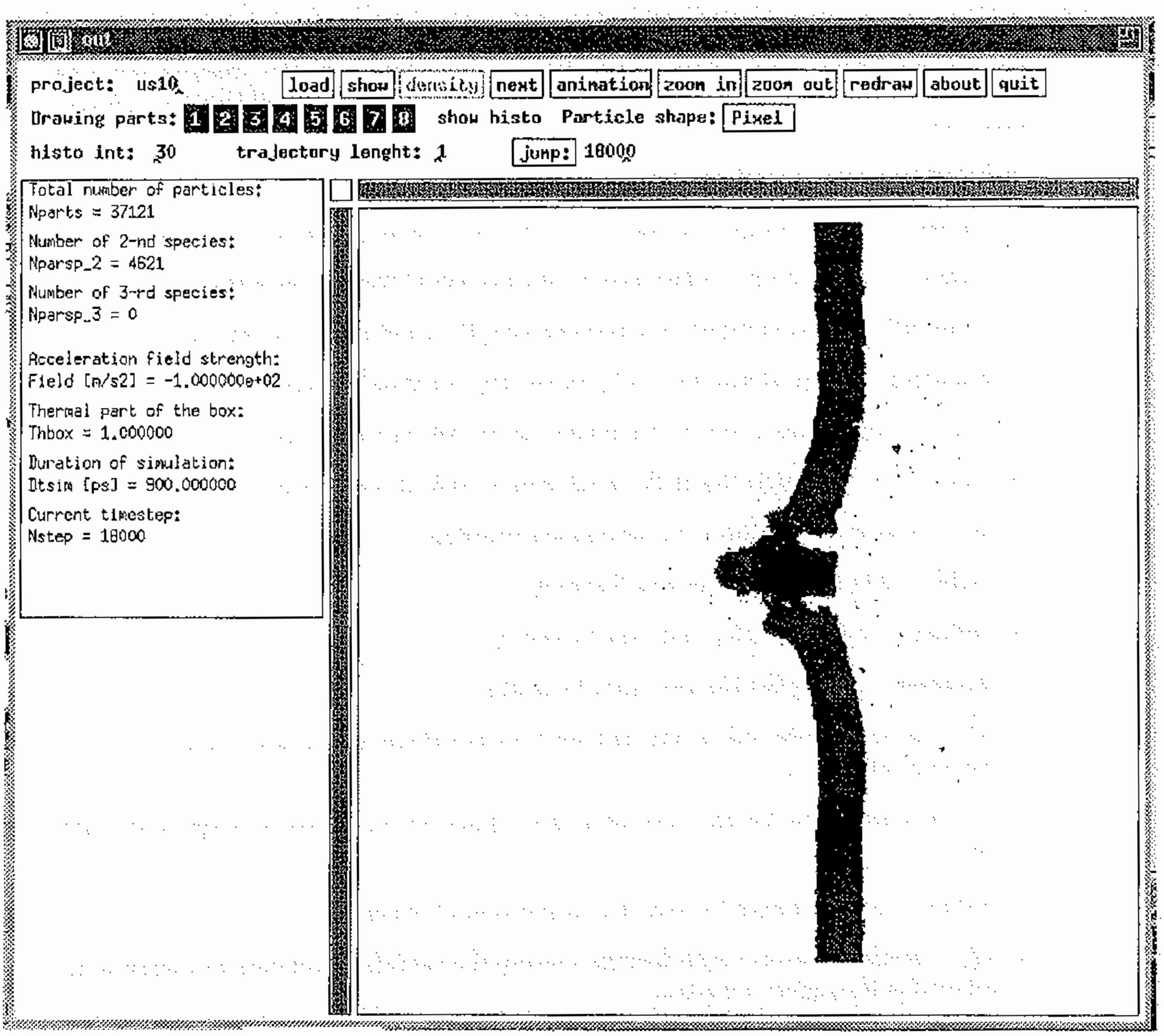
The next items of the file are repeated as many times as the number of steps in the MD simulation

- **Nstep** — 4-byte integer keeping the current step of simulation,
- **X[]** — array of Nparts 8-byte doubles keeping the particle coordinates or momenta in x-direction for a given timestep,
- **Y[]** — array of Nparts 8-byte doubles keeping the particle coordinates or momenta in y-direction for a given timestep.

Two typical pictures of result presentation and one for layout edition are given at the end of this paper. In Fig.1 an example of simulation of penetration mechanism using MD approach is presented [9]. Both the moving object (drawn in black) hitting the wall and the wall itself (drawn in dark-gray) are simulated with Lennard-Jones particles. In Fig.2 the same moment of penetration is presented using unit vectors indicating momenta directions of the particles of the moving object. Fig.3 shows the zoomed initial particle distribution for the penetration experiment. Fragments of the wall and the penetrating object are represented.

Acknowledgments

D.N. is grateful to the Polish Government for the Ph.D. fellowship.



project: us10

Drawing parts: 1 2 3 4 5 6 7 8

Particle shape: Pixel

histo int: 30

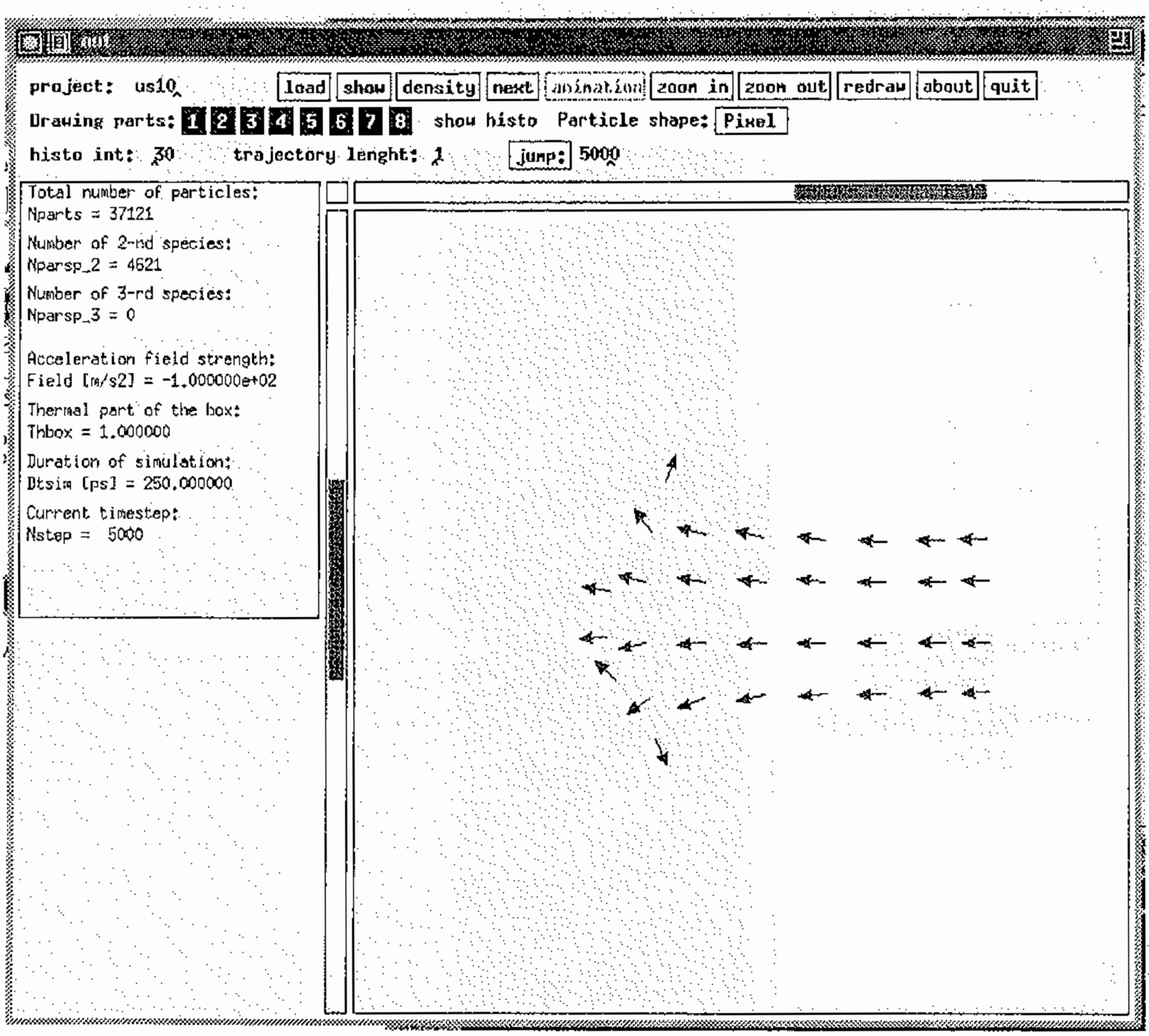
trajectory lenght: 1

jump: 18000

```
Total number of particles:
Nparts = 37121
Number of 2-nd species:
Nparsp_2 = 4621
Number of 3-rd species:
Nparsp_3 = 0

Acceleration field strength:
Field [m/s2] = -1.000000e+02
Thermal part of the box:
Thbox = 1.000000
Duration of simulation:
Dtsim [ps] = 900.000000
Current timestep:
Nstep = 18000
```

Figure 1. Positions of the particles

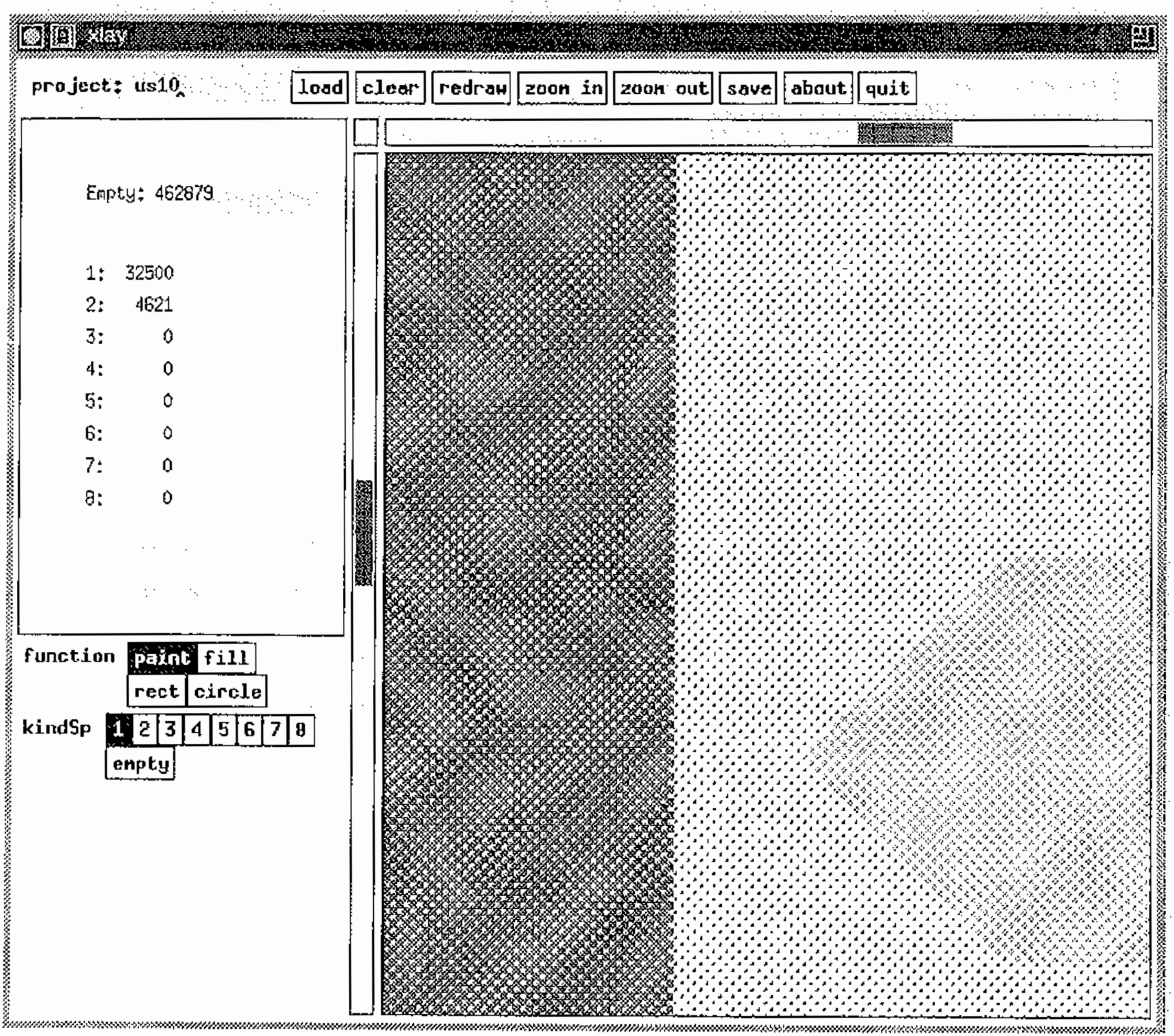


```
project:  us10
Particle shape: Pixel
histo int: 30     trajectory lenght: 1     jump: 5000
```

```
Total number of particles:
Nparts = 37121
Number of 2-nd species:
Nparsp_2 = 4621
Number of 3-rd species:
Nparsp_3 = 0

Acceleration field strength:
Field [m/s2] = -1.000000e+02
Thermal part of the box:
Thbox = 1.000000
Duration of simulation:
Dtsim [ps] = 250.000000
Current timestep:
Nstep =  5000
```

Figure 2. Unit vectors indicating momenta directions of the particles



project: us10

```
Empty: 462879

1:  32500
2:   4621
3:      0
4:      0
5:      0
6:      0
7:      0
8:      0
```

Figure 3. Layout edition

References

[1] Brian W. Kernighan, Dennis M. Ritchie, "The C Programming Language", (polish edition) PWN, Warszawa 1987.

[2] Tarbaj, J., Zimnoch, M., Mościński, J., Kitowski, J., and Bubak, M., "Microscale hydrodynamics on microcomputers", *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, **33** (1991) 62–70 (Informal Newsletter of Daresbury Laboratory, Warrington, England).

[3] Alda, W., Baran, J., Bubak, M., Kitowski, J., Mościnski, J., Nikolov, D., and Słota, R., "Utilities for data edition and results visualization in molecular simulation", presented at EUROMECH coll.287 "Discrete models in fluid dynamics: theory, simulation and experiment", Cagliari, Italy 22-25 September 1992, and *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, **37** (1993) 56-63 (Informal Newsletter of Daresburry Laboratory, Warrington, England).

[4] Xlib - C Language X Interface, MIT X Consortium Standard, X Version 11, Release 5, 1991.

[5] Athena Widget Set - C Language Interface, X Window System, X Version 11, Release 5.

[6] NCSA HDF Calling Interfaces and Utilities, Version 3.1, University of Illinois at Urbana-Champaign, July 1990.

[7] Westermann, M., "PICASSO 2.1 User's Guide", ETH Zürich, 1992.

[8] Długopolski, J., Alda, W., Kitowski, J., "Using PICASSO for visualization of MD results", *CCP5 Information Quarterly for Computer Simulation of Condensed Phases*, this issue, (Informal Newsletter of Daresbury Laboratory, Warrington, England).

[9] W. Alda, W. Dzwinel, J. Kitowski, J. Mościński and D. A. Yuen, "Penetration Mechanics via Molecular Dynamics", University of Minesota Supercomputer Research Report UMSI 93/58, April 1993.

A Input file of parameters

```
       1.1  BASIC CONTROL DATA
= 0  FOR NEW RUN, >=1 FOR EACH RESTART RUN   (LNEWRN    ) =  0
= 1  IF COORDINATE FILE IS DESIRED           (LRESLX    ) =  1
= 1  IF MOMENTUM FILE IS DESIRED             (LRESLV    ) =  0
= 1  IF OUTPUT ON NPRINT = NSTDOU = 7        (LCON      ) =  0
= 1  IF RUN UNDER PASCAL                     (LPAS      ) =  1
     NAME OF PROJECT                         (PRNAME    ) =  test
                 RUN LABEL

   RUN NUMBER                    :  1           DATE   :  14/feb/1991
------------------------------------------------------------------------------
   PROJECT                       :  PROJ
   SIMULATION OF                 :  ARGON
   NUMBER OF MOLECULES           :  3200
   ACCELERATION FIELD STRENGTH   :  1.0E14 [M/S]
------------------------------------------------------------------------------
                  EDITOR FOR SIMULATIONAL PROGRAMS     :  EDS

       2.1  PRINCIPAL PHYSICAL PARAMETERS
DESIRED TEMPERATURE OF THE SYSTEM, K         (TEMPD     ) =  86.5E00
ACCELERATION FIELD STRENGTH, M/S            (FIELD     ) =  1.0E13
PART OF BOX TO ACCELERATION,                 (FLDBOX    ) =  1.0E00
TOTAL NUMBER OF VERTICES                     (NVERTS    ) =  2888
SURFACE DENSITY OF THE SYSTEM, PART/M       (RHOD      ) =  7.17E18
TOTAL NUMBER OF PARTICLES,                   (NPARTS    ) =  25
NO.OF FCC CELLS IN X-DIRECTION,              (NOFCCX    ) =  38
NO.OF FCC CELLS IN Y-DIRECTION,              (NOFCCY    ) =  38
TIME STEP  ,      SEC                        (DT        ) =  1.0E-14
TOTAL NUMBER OF SPECIES,                     (NSPEC     ) =  3
       2.2  DATA FOR INDIVIDUAL SPECIES
MASS OF PARTICLE (1),                        (SMASSD(1) ) =  40.00E11
MASS OF PARTICLE (2),                        (SMASSD(2) ) =  40.00E00
MASS OF PARTICLE (3),                        (SMASSD(3) ) =  40.00E01
TOTAL NUMBER OF 1-ST SPECIE,                 (NPARSP(1) ) =  25
TOTAL NUMBER OF 2-ND SPECIE,                 (NPARSP(2) ) =  0
TOTAL NUMBER OF 3-TH SPECIE,                 (NPARSP(3) ) =  0
PARAMETER EPSILON FOR L-J POTENTIAL,         (EPSLD 1-1 ) =  119.8E00
PARAMETER EPSILON FOR L-J POTENTIAL,         (EPSLD 1-2 ) =  119.8E00
PARAMETER EPSILON FOR L-J POTENTIAL,         (EPSLD 1-3 ) =  119.8E00
PARAMETER EPSILON FOR L-J POTENTIAL,         (EPSLD 2-2 ) =  119.8E00
PARAMETER EPSILON FOR L-J POTENTIAL,         (EPSLD 2-3 ) =  119.8E00
PARAMETER EPSILON FOR L-J POTENTIAL,         (EPSLD 3-3 ) =  119.8E00
PARAMETER SIGMA   FOR L-J POTENTIAL,         (SIGMAD1-1 ) =  0.3405E-9
PARAMETER SIGMA   FOR L-J POTENTIAL,         (SIGMAD1-2 ) =  0.3405E-9
PARAMETER SIGMA   FOR L-J POTENTIAL,         (SIGMAD1-3 ) =  0.3633E-9
PARAMETER SIGMA   FOR L-J POTENTIAL,         (SIGMAD2-2 ) =  0.3405E-9
PARAMETER SIGMA   FOR L-J POTENTIAL,         (SIGMAD2-3 ) =  0.3405E-9
PARAMETER SIGMA   FOR L-J POTENTIAL,         (SIGMAD3-3 ) =  0.3405E-9
```

```
RANGE OF LJ POTENTIAL IN SIGMA UNITS,        (SHORT 1-1 ) =  2.5000E00
RANGE OF LJ POTENTIAL IN SIGMA UNITS,        (SHORT 1-2 ) =  2.5000E00
RANGE OF LJ POTENTIAL IN SIGMA UNITS,        (SHORT 1-3 ) =  1.1224E00
RANGE OF LJ POTENTIAL IN SIGMA UNITS,        (SHORT 2-2 ) =  2.5000E00
RANGE OF LJ POTENTIAL IN SIGMA UNITS,        (SHORT 2-3 ) =  1.1224E00
RANGE OF LJ POTENTIAL IN SIGMA UNITS,        (SHORT 3-3 ) =  2.5000E00
       3.1  NUMERICAL PARAMETERS
NO.OF TIMESTEPS PLANNED FOR THIS RUN,        (MSTEPS    ) =  60000
NO.OF TIMESTEPS IN ROLLING AVERAGES,         (NROLAV    ) =  10
NO.OF GRID POINTS IN POTENTIAL TABLES,       (NSTPOT    ) =  10000
HARD CORE OF LJ POTENTIAL IN SIGMA UNITS,    (HSPH      ) =  0.7E00
PART BOX TO THERMALIZATION,                  (THBOX     ) =  1.0
INDEX FOR TEMPERATURE SCALING OPTION,        (LSCTEM    ) =  1
NO.OF TIMESTEPS BETWEEN EACH TEMP. SCAL.,    (NSTEMP    ) =  10
LAST TIMESTEP TEMPERTURE SCALING APPLIED,    (NSCTMP    ) =  1000
       4.1  OUTPUT PARAMETERS
LENGTH OF PRINTOUT PAGE,                     (NPGLEN    ) =  52
INITIAL TIMESTEP COUNTER FOR DATA PRINT.,    (NPCNT     ) =  0
NO.OF TIMESTEPS BETWEEN DATA PRINTOUT,       (NPST      ) =  100
INITIAL TIMESTEP COUNTER FOR DATA SAVING,    (NSCNT     ) =  0
NO.OF TIMESTEPS BETWEEN DATA SAVING,         (NSST      ) =  1200
COUNTER INDEX FOR RESULTS STORING,           (NRSCNT    ) =  0
NO.OF TIMESTEPS BETWEEN RESULTS STORING,     (NRSST     ) =  300
INITIAL TIMESTEP FOR RESULTS STORING,        (NOROUT    ) =  0
INITIAL TIMESTEP FOR AVG. AND RMS. CALC.,    (NSWMV     ) =  5001
```