

FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

**Thermalization of Large Fluid Systems
consisting of Lennard-Jones Mixtures**

Matthias Hloucha

FZJ-ZAM-IB-9902

Februar 1999

(letzte Änderung: 4.2.1999)

Contents

1	Program MCMD	1
1.1	Introduction	1
1.2	Method	1
2	Usage	4
2.1	Compilation	4
2.2	Running Jobs	4
2.3	Test	5
2.4	Parameters	5
2.5	Recipe for building a large thermalized configuration	5
3	Results	7
3.1	Properties	7
3.2	Benchmarks	7
3.3	Remarks	11

Chapter 1

Program MCMD

Old Chemists never die – they just reach equilibrium

1.1 Introduction

At the Center for Applied Mathematics, Research Center Jülich, Germany, work has been started to investigate massively parallel algorithms for molecular dynamics computer simulations of liquids. As part of this project, a program MCMD was developed in order to construct starting configurations for simulations with a large number of particles. The ideas, usage and some benchmarks of MCMD are presented by the present text.

There are two common methods to construct a starting point for a subsequent liquid simulation. While some authors suggest to start from a regular lattice [1], others prefer random configurations [2]. Both methods have their advantages and drawbacks. A start from a regular lattice might be problematic for dipolar fluids at high densities, where melting is inhibited by long range dipolar interactions. This approach is also expected to fail for mixtures of molecules of significantly different size. Random mixing of molecules, however, often leads to unrealistic short intermolecular distances. The resulting huge repulsive forces would make molecular dynamics simulations either instable or one would have to use a very short time step.

The idea of this work is to use a combination of both methods – random mixing and lattice start – together with Monte Carlo *and* molecular dynamics simulation techniques.

1.2 Method

The principal function of MCMD is illustrated in the Figures 1.1–1.4. First, the required particle number and volume is divided into a subsystem, which is allowed to contain not more than 1,000 particles. The subsystem is also chosen to be a unit cell of a cubic lattice for the whole system. If it is not possible to divide the particle number exactly into a cubic subsystem, then the particle number in the subsystem is slightly increased.

The coordinates of particles in the subsystem are randomly distributed in the given volume (see Figure 1.1). In the picture of an Argon–Krypton mixture at a typical liquid density some unrealistic short interatomic distances are clearly visible as overlapping spheres.

A Monte Carlo simulation in the NVT ensemble is performed next in order to allow a configurational relaxation of the system (see Figure 1.2) [3]. The displacements of the MC moves are independent of the potential difference; therefore the simulations are stable, even for such unrealistic initial configurations. Unbiased molecular dynamics codes would fail, because of the huge accelerations.

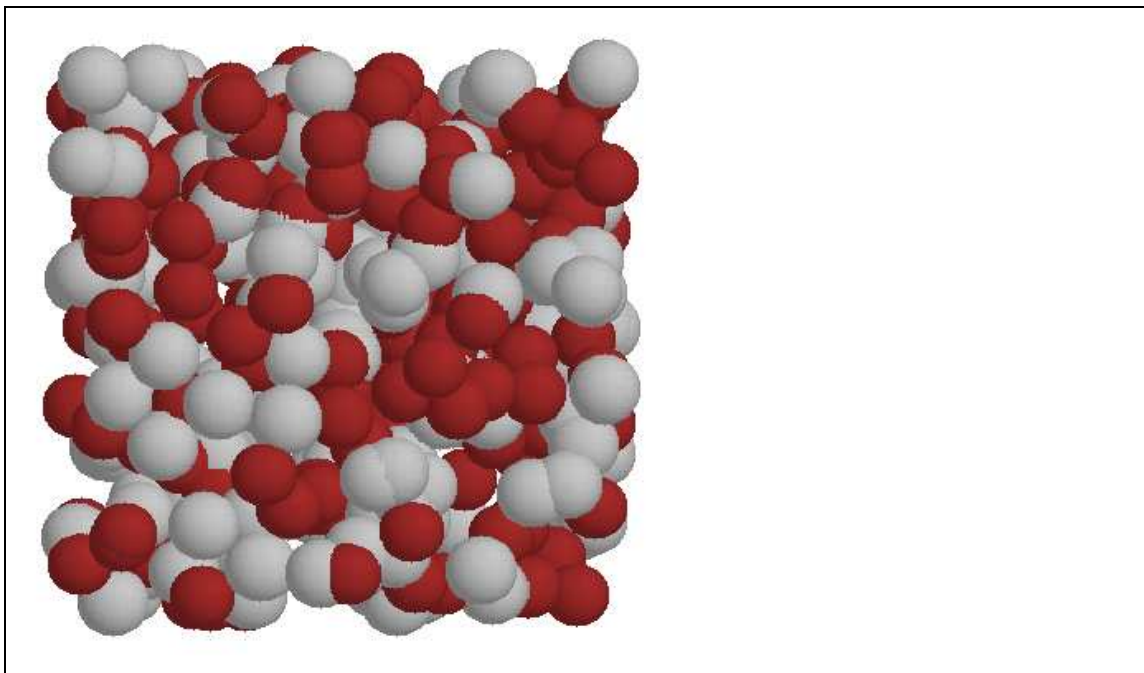


Figure 1.1: Construct a random configuration of a subsystem.

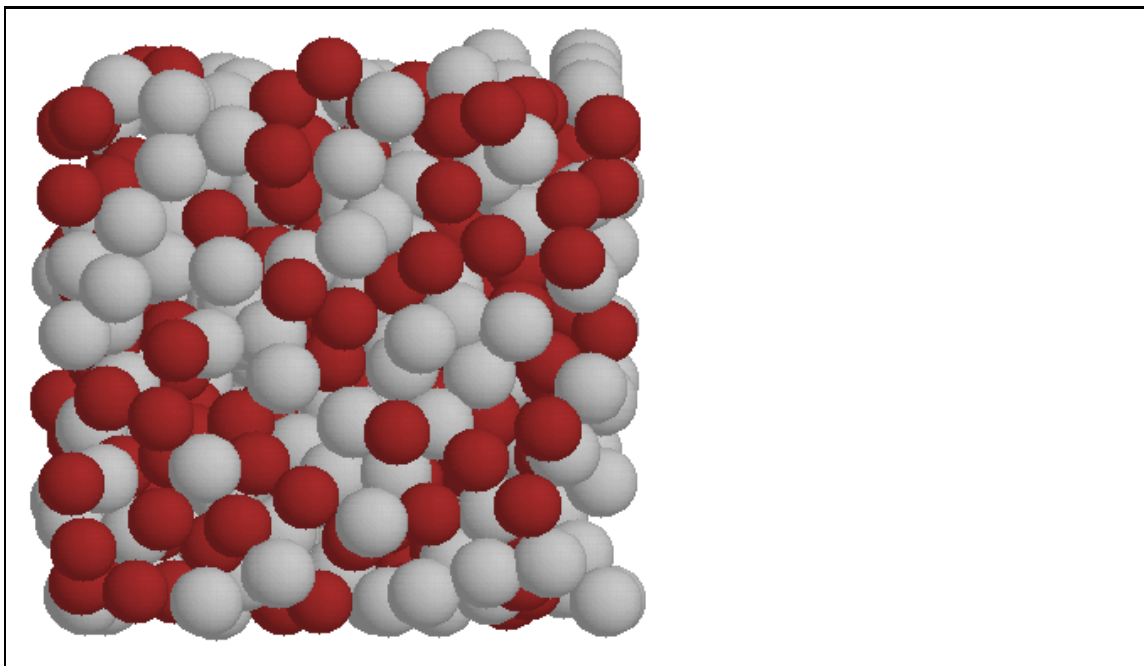


Figure 1.2: Monte Carlo simulation to thermalize the subsystem.

In a third step, a cubic lattice with a large number of particles is constructed from periodic copies of the subsystem (see Figure 1.3). The periodic boundary conditions imposed on the subcell avoid conflicts between neighbor cells. At this stage there is no limit for the desired particle number.

Additional molecular dynamics simulations are used to randomize and thermalize the large system (see Figure 1.4). The MD part of the program MCMD is optimized for large particle numbers. The linked cell method is used to reduce the effort for the calculation of pair distances [4]. The calculation of the force matrix – which usually consumes more than 98 percent of the computing time – is parallelized for shared memory and distributed memory parallel computing machines (SGI,SUN,Cray). Parallelization was not a primary goal of this project, but a speedup of 10 is easily achieved by a simple master–slave approach. The parallelized version of the program is capable of simulations with up to one million particles.

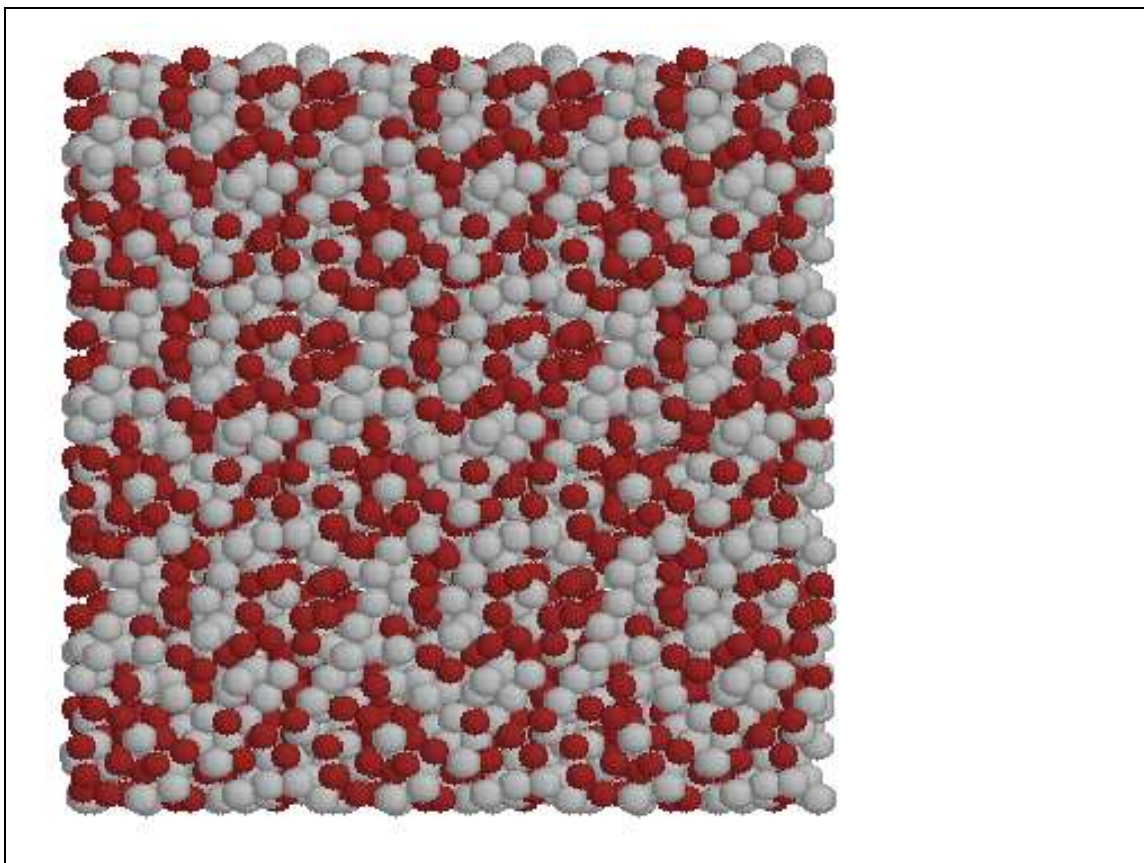


Figure 1.3: Build a lattice from the equilibrated subsystem.

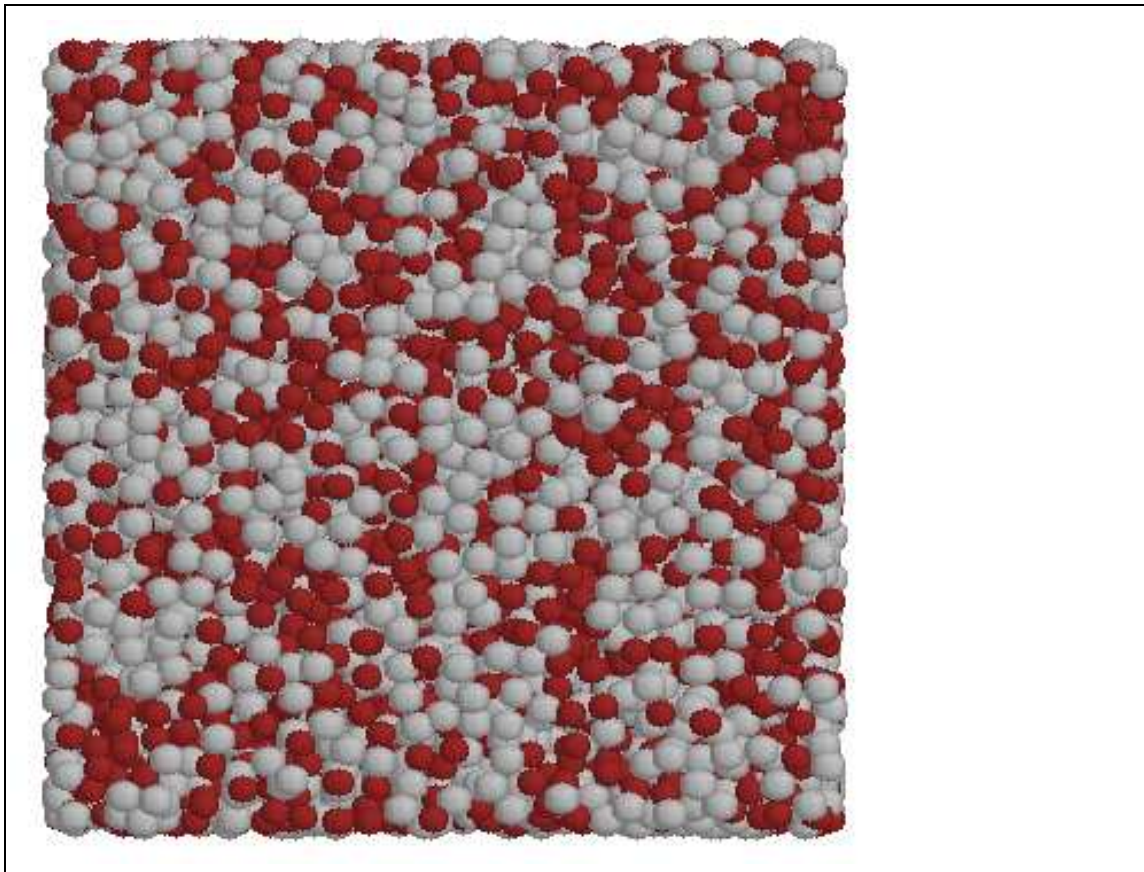


Figure 1.4: Parallelized molecular dynamics code to thermalize the large system.

Chapter 2

Usage

2.1 Compilation

The program MCMD is written in C. A serial version is running on all standard UNIX platforms. It was tested on: IBM AIX, DEC Alpha, SGI Irix, SUN Solaris, Intel based Linux and Cray UNICOS. Optimized and parallelized versions of this program exist for SUN Solaris and SGI Irix shared memory architectures and Cray T3E. Shared memory parallelization is done by `pragma` directives; the version for Cray T3E is based on MPI routines.

The compilation is managed by compiler options in the Makefile. They are summarized in Figure 2.1. Preprocessor defines are used to build the platform dependent optimized version (`SERIAL`, `MPP`, `SUNPAR`, `SGIPAR`). Further speedup on SGI Power Challenge machines with Mips R8000 processors is obtained by an additional flag: `-r8000`.

2.2 Running Jobs

The following lines contain the UNIX commands to start the program on two CPUs:

```
SGI:  setenv NUM_THREADS 2; mcmd
SUN:  setenv PARALLEL 2;    mcmd
T3E:  mpprun -n 2 mcmd
```

Default	
CC	= cc
CFLAGS	= -DSERIAL
LIBS	= -lm
Cray MPI parallel	
CFLAGS	= -DMPP -h inline0 -h scalar2 -h vector2
LIBS	= -lm -lmpi
SUN shared memory parallel	
CFLAGS	= -DSUNPAR -xO5 -fast -xexplicitpar -xloopinfo -xvpara
LIBS	= -lm
SGI shared memory parallel	
CFLAGS	= -DSGIPAR -O3 -mp
LIBS	= -lm -mp

Figure 2.1: Compiler options


```

control of the program MCMD
0          O:MD 1:MC
20         n-step blocking of vir., g(r) and energy calc.
1500      resolution g(r): number of points
15.0      max. g(r) [A]
0.4       MC: acceptance ratio
4         MD: temp bath 0:strict 1:weak 2: Andersen 3:NVE 4:Nose
1         MD: linked cell method; 0=off
4         MD: factor between mean density and max. cell density
2000.0    MD: max. velocity distribution [m/s]
100       MD: n-step waiting before start of acf calculation
1         MD: n-step blocking of acf calculation
100       MD: resolution acf: number of points
1.0       MD: Nose heatbath: (Chi) []
1.0e-45   MD: Nose heatbath: (Q) [Js2]
1.0       MD: Nose heatbath: (lnS) []

```

Figure 2.2: mh.inp

2.3 Test

In order to check the syntax and runtime performance of the program a test case is implemented. This may be executed by:

```

cp TEST.konfig.xyz konfig.xyz
cp TEST.mcmd.inp mcmd.inp
cp TEST.mh.inp mh.inp
mcmd > out
diff out TEST.out
diff konfig.xyz TEST.sav.konfig

```

2.4 Parameters

The program is controlled by two input files: `mcmd.inp` and `mh.inp`. For compatibility reasons the first one has the same structure as the input file for the program DMMD, which is under construction at ZAM. The file `mcmd.inp` contains temperature, volume, number of species in the fluid system and their interaction parameters; it also contains simulation control parameters like number of simulation steps, blocking size, time step and cutoff radius (see Figure 2.3). In `mh.inp` the type of the simulation is specified: *NVT-MC*, *NVE-MD* or *NVT-MD* with four different methods for fixing the temperature. Some parameters for the evaluation of system properties are also given in this file (see Figure 2.2).

2.5 Recipe for building a large thermalized configuration

1. Monte Carlo run to build and relax a subsystem and to construct a large system from the subsystem. Necessary steps:
 - Change line 15 in `mcmd.inp` into `false` (construct a new configuration).
 - Change line 2 in `mh.inp` into 1 (use Monte Carlo).
 - Run the program
 - Check the file with the large configuration in `konfig.xyz`.

```

N=1000 Argon Krypton mixture

parameter V,T
  3.85408e25      molar volume   [A3]
  300.0          temperature   [K]
  10             MD: time step  [fs]
  10e0          cutoff         [A]

parameter  simulation run
  500                simulation steps
  100                blocking
  0                  x
  0                  x
  0                  x
  .true.            restart
  x                 x
  x                 x

parameters for system properties
  x                 x
  x                 x

parameters for xyz film file
  x                 x
  0                 x

Parameter Temperature-Scaling
  0                 x
  0                 x
  0                 x

  400                MD: t_temp_scal [fs]

molecule  parameters
  2          n_spec (no. of lennard-jones species)
  500      500  n_part(i)
  119.8   167.0 eps(i)
  3.405   3.633 sig(i)
  39.948  83.800 mass(i)
Ar  Kr    species_name(i)

```

Figure 2.3: mcmd.inp

2. Molecular dynamics for further thermalization of the large system.

- Change line 15 in mcmd.inp into true (restart from konfig.xyz).
- Change line 2 in mh.inp into 2 (molecular dynamics).
- Change line 7 in mh.inp into 0 (strict scaling is about 15 percent faster than Nosé thermostat).
- Change line 8 in mh.inp into 1 (linked cell method is necessary for good performance).
- Run the program on SGI, SUN or Cray machines with up to 8 CPUs.

Chapter 3

Results

3.1 Properties

Simulations with an Argon–Krypton mixture were performed in order to demonstrate the capabilities of MCMD. The potential parameters are given in Table 3.1. The interaction parameters for Argon and Krypton are standard Lennard-Jones values; the Lorentz–Bertholet combination rules were used to calculate the Ar–Kr potential parameters. Details of the simulation are listed in Table 3.2. Several properties are accessible from simulations with MCMD: internal energy U , pressure P , kinetic temperatures, average displacement per time step, pair distribution functions $g(r)$ for all possible interaction types, velocity autocorrelation functions and velocity distributions for each species [5].

Simulated pair distribution functions are shown in Figure 3.1. The first peak of $g(r)$ is shifted with increasing σ to larger distances. The height of the first peak is noticeably influenced by the depths of the pair potential (compare Table 3.1). The velocity autocorrelation functions for Argon and Krypton are shown in Figure 3.2. The correlation time of Krypton is clearly larger than that of Argon. The lower mass of Argon leads to – as expected from the Maxwellian velocity distribution function – a broader velocity distribution (see Figure 3.3).

3.2 Benchmarks

The parameters for the benchmark runs are given in Tables 3.1 and 3.2. The radial cutoff for simulations with 1,000,000 particles is reduced to 10×10^{-10} m. The trajectories are calculated using the velocity Verlet algorithm [4] with strict temperature scaling is applied in every time step. Computing times per MD time step for systems with one million particles are shown in Table 3.3. These values were obtained from an average interval over 10 time steps.

	σ / 10^{-10} m	ϵ /K
Ar–Ar	3.405	119.8
Kr–Kr	3.633	167.0
Ar–Kr	3.519	141.4

Table 3.1: MD simulation of Ar–Kr mixture. Lennard-Jones pair potential parameters.

temperature /K	density / g cm^{-3}	number of particles	time step / 10^{-15} s	cutoff / 10^{-10} m
300	1.605	15,000 – 800,000	10.0	15.0

Table 3.2: MD simulation parameters for the Ar–Kr system.

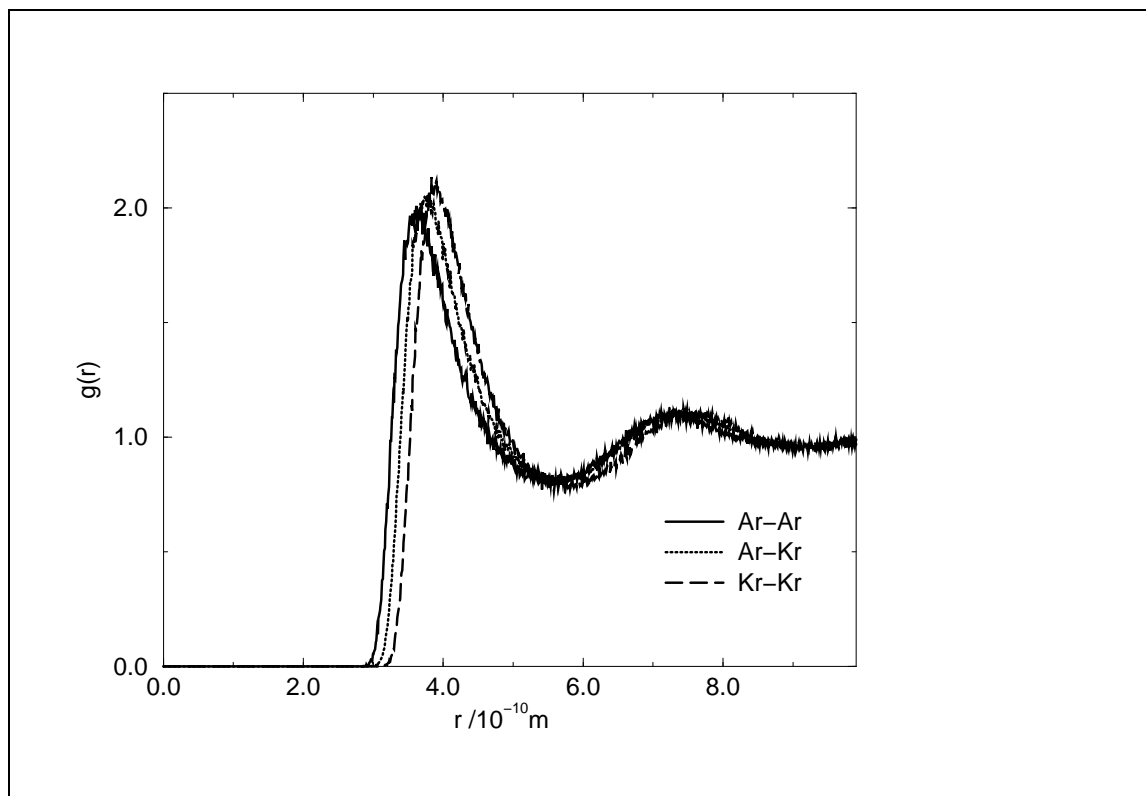


Figure 3.1: MD simulation of Ar-Kr mixture. Radial pair distribution function $g(r)$ at $T = 300$ K and $\rho = 1,6 \text{ g cm}^{-3}$.

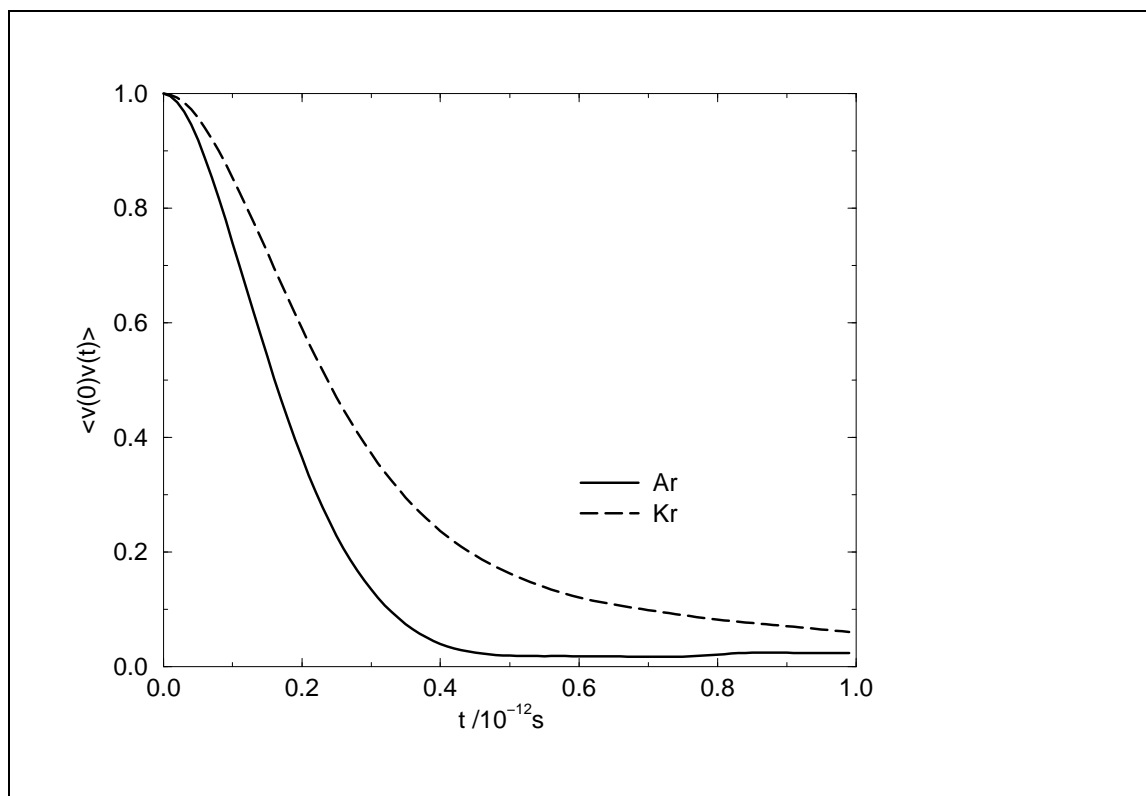


Figure 3.2: Simulated velocity autocorrelation functions in a mixture of Argon and Krypton at $T = 300$ K and $\rho = 1,6 \text{ g cm}^{-3}$.

	SUN Enterprise	SGI Power Challenge	Cray T3E
number of CPUs	8	8	8
time per MD step /s	43,9	45,3	18,5

Table 3.3: MD simulation with 1,000,000 particles.

PEs	MPI code	shared memory code
	time MD step /s	time MD step /s
1	14.3	14.3
2	6.85	7.27
4	3.90	3.99
8	2.56	2.34
16	2.36	1.50
32	3.02	1.08
64	4.65	0.88

Table 3.4: Simulation with 100,000 particles. Computing times for one MD time step on SGI O2000.

Performance measurements on Cray T3E are shown in Figures 3.4 and 3.5. The program exhibits very high MFLOPS rates and good scaling on up to eight CPUs. The MFLOPS rate drops noticeably with increasing particle number (see values for 8 PEs in Figure 3.4). The master-slave parallelization of the force matrix calculation leads to more data traffic for larger systems.

Speedup data for simulations with 100,000 particles on SGI O2000 and Cray T3E are shown in Figure 3.6. The best speedup of almost 10 is achieved with the shared memory code on SGI O2000 using 16 processors. The shared memory code shows better scaling in comparison to the MPI code on SGI O2000. The scaling of the MPI code is slightly better for Cray T3E than for SGI O2000. Computing times for 100,000 particles are shown Table 3.4.

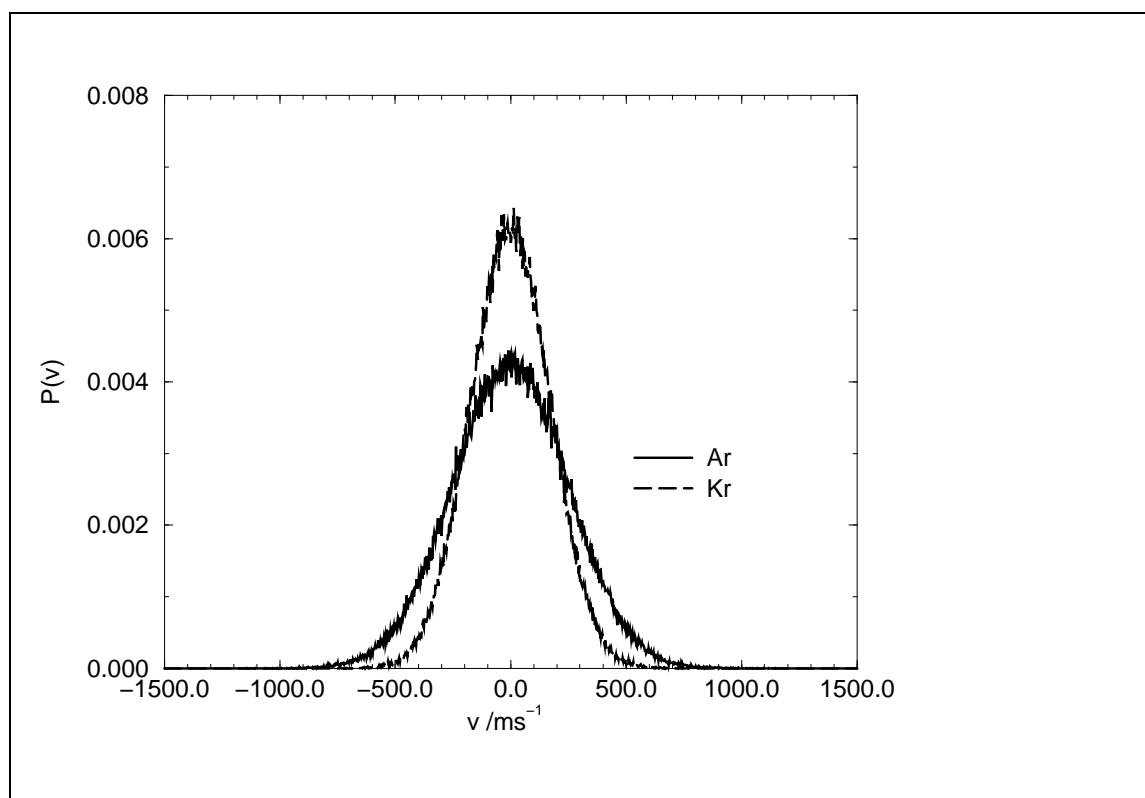


Figure 3.3: Velocity distribution in a liquid mixture of Argon and Krypton at $T = 300$ K and $\rho = 1,6$ g cm⁻³.

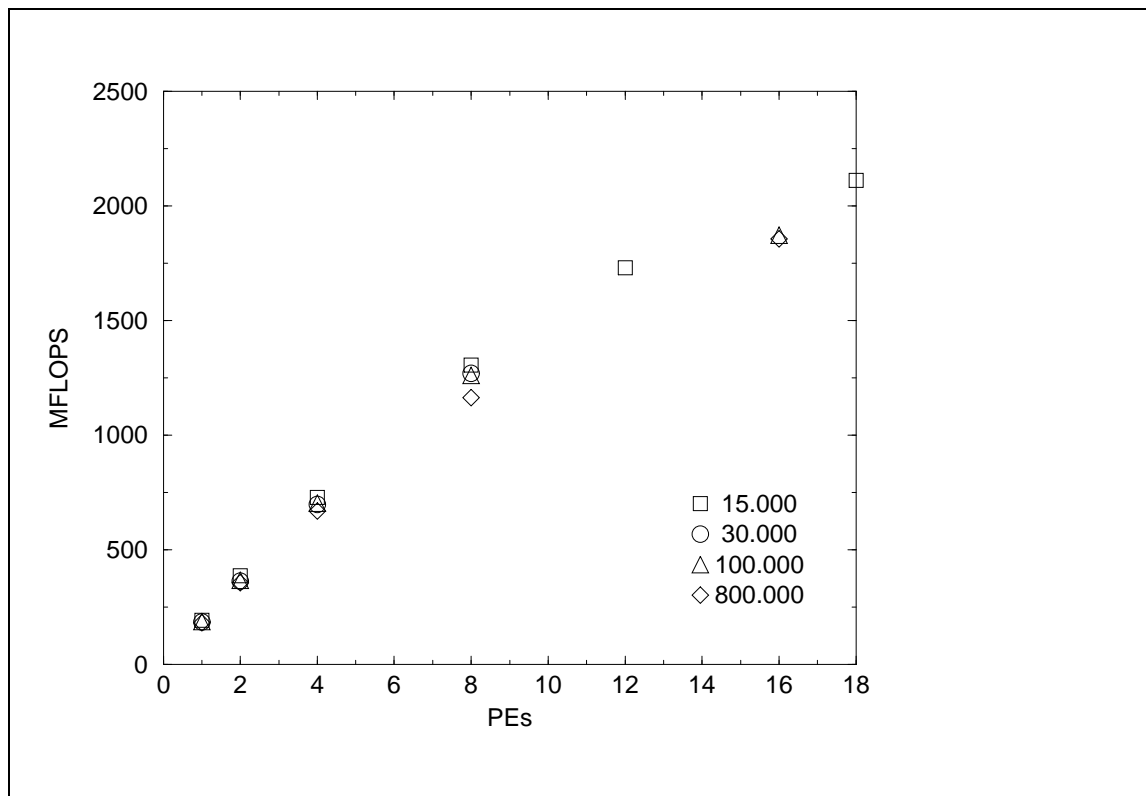


Figure 3.4: MD simulation of Ar-Kr mixture. MFLOPS rates on Cray T3E.

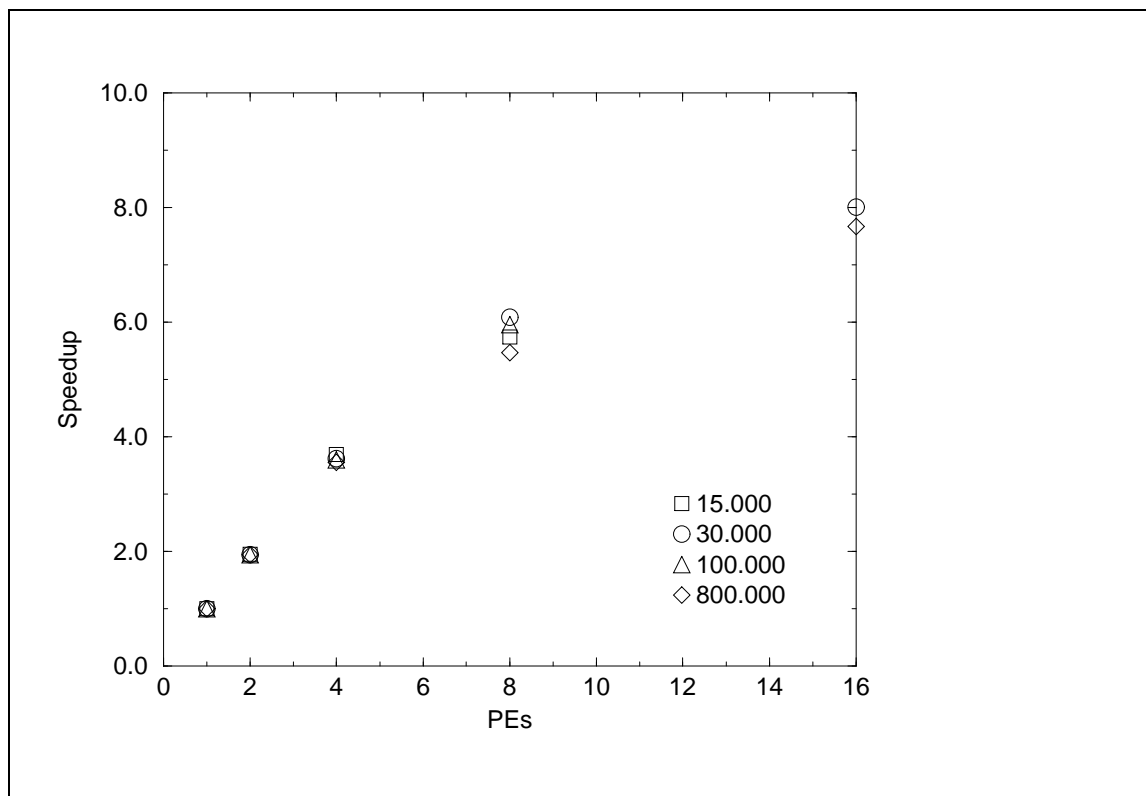


Figure 3.5: MD simulation of Ar-Kr mixture. Speedup on Cray T3E for different particle numbers.

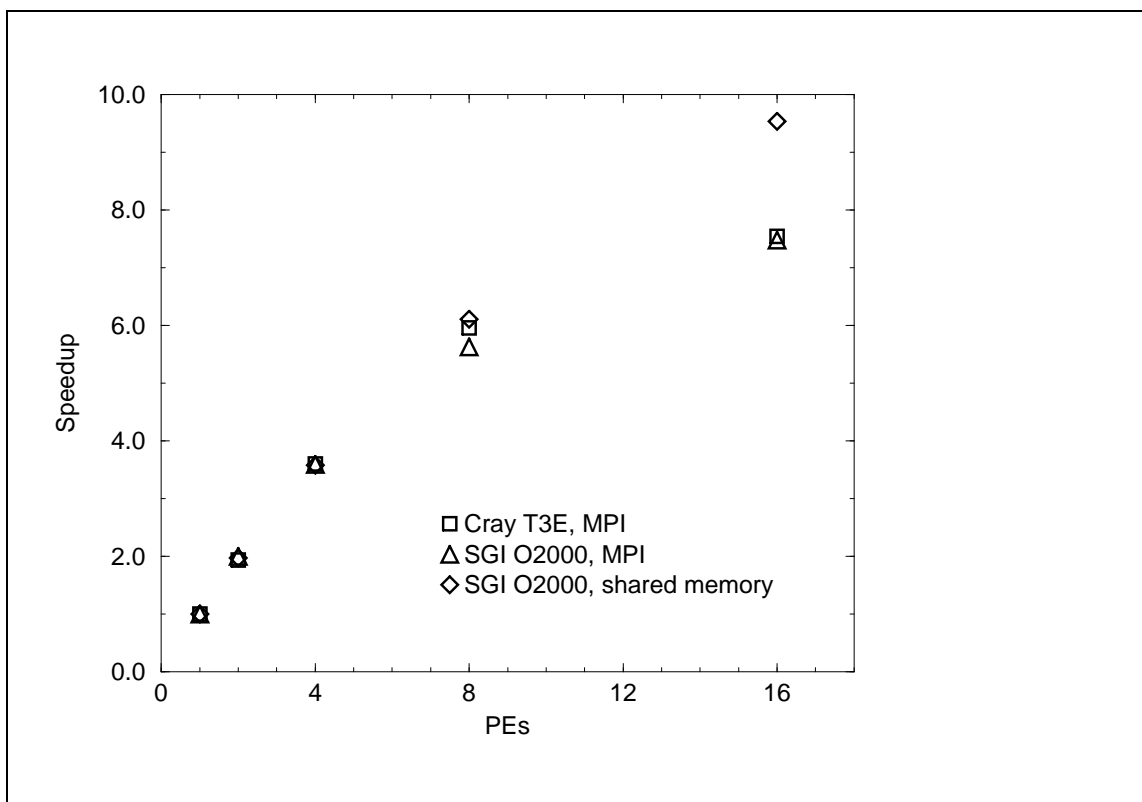


Figure 3.6: MD simulation of Ar–Kr mixture with 100,000 particles. Speedup on Cray T3E (MPI code) and SGI Origin2000 (MPI and shared memory code)

3.3 Remarks

The method of the program MCMD, to construct a large system from periodic copies of a subsystem, implies the danger of long range correlations. This could be problematic, especially for dipolar substances or ions, where electrostatic interactions are important. These correlations can be avoided by subsequent MC runs – with appropriate boundaries – for each subcell of the large system.

The question when equilibrium is reached has been the subject to many discussions. In principle every accessible property and its fluctuation could be used as a check for thermalization. Internal energy, pressure and kinetic temperatures are printed out during a run of MCMD. They should be checked, together with the structural and dynamical functions, which are given at the end of a simulation.

Bibliography

- [1] F. Vesely. *Computerexperimente an Flüssigkeitsmodellen*. Physik Verlag, Weinheim, 1978.
- [2] G. Corongiu. Molecular dynamics simulation for liquid water using a polarizable and flexible potential. *Int. J. Quant. Chem.*, 42:1209–1235, 1992.
- [3] J. A. Barker and D. Henderson. What is “liquid”? Understanding the states of matter.. *Rev. Mod. Phys.*, 48:587–671, 1976.
- [4] D. Frenkel and B. Smit. *Understanding molecular simulation*. Academic Press, Amsterdam, 1996.
- [5] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*. Clarendon Press, Oxford, 1987.