

Molecular Simulation Course: Exercises

Thijs J.H. Vlugt, Maddalena Venturoli, Daan Frenkel and Berend Smit

Department of Chemical Engineering, University of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam, The Netherlands

This document describes the exercises for the Molecular Simulation Course by Berend Smit and Daan Frenkel. This document and the accompanying programs are available on the web [1].

Version 4.0, January 15, 2001.

Contents

1	Introduction	1
2	Statistical Mechanics	3
2.1	Distribution of particles	3
2.2	Boltzmann distribution	3
2.3	Coupled harmonic oscillators	4
2.4	Random Walk on a 1D lattice	4
2.5	Random Walk on a 2D lattice	5
3	Basic Monte Carlo techniques	7
3.1	Calculation of π	7
3.2	The photon gas	7
3.3	Hard disks in a square	8
3.4	MC of a Lennard-Jones system	9
3.5	Scaling	9
3.6	Ewald summation	10
3.7	Parallel tempering	11
4	Basic Molecular Dynamics techniques	13
4.1	MD of a Lennard-Jones system	13
4.2	Parallel MD	15
5	MD and MC in various ensembles	17
5.1	Barrier crossing (Part 1)	17
5.2	MC in the NPT ensemble	18
5.3	Ising model	19
6	Phase equilibrium and free energy calculations	21
6.1	Vapor-liquid equilibrium	21
6.2	Umbrella sampling	22
7	Rare events	23
7.1	Barrier crossing (Part 2)	23
7.2	Transition Path Sampling	24
7.2.1	Introduction	24
7.2.2	MC sampling from the distribution $F(x_0, T)$	26
7.2.3	Model system	27
7.2.4	Questions	27

8	Configurational-Bias Monte Carlo	29
8.1	CBMC of a single chain	29
8.2	CBMC of a simple system	31
8.3	Overlapping distribution for polymers	32
9	Appendix A: Introduction to GNU/Linux	35
10	Appendix B: Introduction to FORTRAN77	39
11	Appendix C: Introduction to <i>MOLMOL</i>	43
12	Appendix D: Thermodynamic equations	45
13	Appendix E: Equations of motion from the Lagrangian or Hamiltonian	47
13.1	Introduction	47
13.2	Lagrangian	48
13.3	Example	50
13.4	Hamiltonian	50
14	Appendix F: Questions about the book “Understanding ..” [2]	53
14.1	Introduction to FORTRAN77	53
14.2	Chapter 2	53
14.3	Chapter 3	57
14.4	Chapter 4	57
14.5	Chapter 5	58
14.6	Chapter 6	59
14.7	Chapter 7	59
14.8	Chapter 8	60
14.9	Chapter 11	61
14.10	Chapter 13	61
15	Appendix G: Possible research projects	63
15.1	Adsorption in porous media	63
15.2	Transport properties in liquids	63
15.3	Parallel Molecular Dynamics	64
15.4	Phase diagram of ethane	65
15.5	Diffusion in a porous media	65
15.6	Multiple time step integrators	66
15.7	Thermodynamic integration	67
15.8	Hints for programming	68
16	Appendix H: Software	71
	Acknowledgements	73
	Bibliography	75

Chapter 1

Introduction

This document describes the exercises for the Molecular Simulation Course by Daan Frenkel and Berend Smit. To do the exercises, it is essential to have a copy of the book “Understanding Molecular Simulation” by Daan Frenkel and Berend Smit [2]. Some of the exercises in this document were taken from several standard textbooks [2–6].

In most of the exercises, some programming has to be done. One does not have to write a program from scratch, but rather one has to make some small modifications to an existing code^{*}. We have programmed the code in FORTRAN77 because it is quite easy to learn. However, the random number generator that is used in some of the programs is written in C [7]. See, for example, ref. [8] for more information about random number generators. All programs are provided with a makefile that has been prepared for GNU/Linux using the g77/gcc compilers [9, 10]. Compiling the programs on other Unix systems is straightforward. Solutions of the programming exercises as well as the \LaTeX source of this document are available on request.

The course and exercises are divided into several parts:

- Introduction to Unix and FORTRAN77 (0.5 day)
- Basic Statistical Mechanics (1.5 day)
- Basic Monte Carlo (MC) techniques (2 days)
- Basic Molecular Dynamics (MD) techniques (2 days)
- MD and MC in various ensembles (1 day)
- Phase equilibrium and free energy calculations (1 day)
- Rare events (1 day)
- Configurational-Bias Monte Carlo (1 day)

We have setup the course in such a way that lectures are given every morning (3 hours), while the exercises have to be done in the afternoon (5 hours); we do not expect that all students have enough time to complete all exercises. After all exercises have been completed, a special research project has to be done (duration: 2 weeks) to pass the exam. In Appendix G, we have given some suggestions for research projects. We have also included a list of questions about

^{*}Although the text may suggest otherwise, all programs do not have to be written from scratch. For more information, see the *README* file from the distribution.

the book of Daan Frenkel and Berend Smit [2] (Appendix F).

We have done our best to remove all errors from this document and the programs. However, we have to make the following statement:

“We make no warranties that this document or the programs accompanied by this document are free of error, or that they will meet your requirements for any particular application. The authors disclaim all liability for direct and consequential damages resulting from the use of this document or the programs accompanied by this document. It is not allowed to distribute this document and the accompanying programs.”

We are interested in what you think about this course. Any comment would be appreciated, even severe criticism. Comment can be send to Berend Smit.

Contact address:

Berend Smit
Department of Chemical Engineering
University of Amsterdam
Nieuwe Achtergracht 166
1018 WV Amsterdam, The Netherlands
email: smit@its.chem.uva.nl
web: <http://molsim.chem.uva.nl>

Chapter 2

Statistical Mechanics

2.1 Distribution of particles

Consider an ideal gas of N particles in a constant volume at constant energy. Let us divide the volume in p identical compartments. Every compartment contains n_i molecules such that

$$N = \sum_{i=1}^{i=p} n_i \quad (2.1)$$

An interesting quantity is the distribution of molecules over the p compartments. Because the energy is constant, every possible eigenstate of the system will be equally likely. This means that in principle it is possible that one of the compartments is empty.

1. Enclosed is a program that calculates the distributions of molecules along the p compartments. Run the program for different numbers of compartments (p) and total number of gas molecules (N). Note that the code has to be completed first (see the file *distribution.f*). The output of the program is the probability to find x particles in a particular compartment as a function of x . This is printed in the file *output.dat*, which can be plotted using *xmgr -nxy output.dat* (the option *-nxy* means that more than one column is plotted).
2. Why does it (almost) never happen that one of the compartments is empty ?
3. Consider the case of $p = 2$. The probability of finding n_1 molecules in compartment 1 and $n_2 = N - n_1$ molecules in compartment 2 is given by

$$P(n_1) = \frac{N!}{n_1! \times (N - n_1)! \times 2^N} \quad (2.2)$$

Compare your numerical results with the analytical solution for different values of N . Show that this distribution is a Gaussian for small n_1 . Hint: For $x > 10$, it might be useful to use Stirling's approximation:

$$x! \approx (2\pi)^{\frac{1}{2}} x^{x+\frac{1}{2}} \exp[-x] \quad (2.3)$$

2.2 Boltzmann distribution

Consider a system of N energy levels with energies $0, \epsilon, 2\epsilon, \dots, (N-1) \times \epsilon$ and $\epsilon > 0$.

Questions:

1. Calculate, using the given program, the occupancy of each level for different values of the temperature. What happens at high temperatures ?
2. Change the program in such a way that the degeneracy of energy level i equals $i + 1$. What do you see ?
3. Modify the program in such a way that the occupation of the energy levels as well as the partition function (q) is calculated for a linear rotor with moment of inertia I . Compare your result with the approximate result

$$q = \frac{2I}{\beta \hbar^2} \quad (2.4)$$

for different temperatures. Note that the energy levels of a linear rotor are

$$U = J(J + 1) \frac{\hbar^2}{2I} \quad (2.5)$$

with $J = 0, 1, 2, \dots, \infty$. The degeneracy of level J equals $2J + 1$.

2.3 Coupled harmonic oscillators

Consider a system of N harmonic oscillators with a total energy U . A single harmonic oscillator has energy levels $0, \epsilon, 2\epsilon, \dots, \infty$ ($\epsilon > 0$). All harmonic oscillators in the system can exchange energy.

Questions:

1. Invent a computational scheme for the update of the system at constant total energy (U). Compare your scheme with the scheme that is incorporated into the given computer code (see the file *harmonic.f*).
2. Make a plot of the energy distribution of the first oscillator as a function of the number of oscillators for a constant value of U/N (*output.dat*). Which distribution is recovered when N becomes large ? What is the function of the other $N - 1$ harmonic oscillators ? Explain.
3. Compare this distribution with the canonical distribution of a single oscillator at the same average energy (use the option NVT).
4. How does this exercise relate to the derivation of the Boltzmann distribution for a system at temperature T on page 12 of ref. [2] ?

2.4 Random Walk on a 1D lattice

Consider the random walk of a single particle on a line. The probability that after N jumps the net distance of the particle equals n , is:

$$\ln(P(n, N)) \approx \frac{1}{2} \ln\left(\frac{2}{\pi N}\right) - \frac{n^2}{2N} \quad (2.6)$$

Questions:

1. Derive this equation. The probability to jump to one direction equals the probability to jump to the other direction. For large x , $x!$ can be approximated by:

$$x! \approx (2\pi)^{\frac{1}{2}} x^{x+\frac{1}{2}} \exp[-x] \quad (2.7)$$

2. Compare this theoretical result with the computed root mean square displacement and the computed function $P(n, N)$ (see the file *output.dat*). What is the diffusivity of this system?
3. Modify the program in such a way that the probability to jump in one direction equals 0.8. What happens?

2.5 Random Walk on a 2D lattice

Consider the random walk of N particles on a $M \times M$ lattice. Two particles cannot occupy the same lattice site. On this lattice, periodic boundaries are used. This means that when a particle leaves the lattices it returns on the opposite side of the lattice.

Questions:

1. What is the fraction occupied sites (θ) of the lattice as a function of M and N ?
2. Make a plot of the diffusivity D as a function of θ for $M = 32$. For low values of θ , the diffusivity can be approximated by

$$D \approx D_0 (1 - \theta) \quad (2.8)$$

Derive this equation. Why is this equation not exact?

3. Modify the program in such a way that the probability to jump in one direction is larger than the probability to jump in the other direction. Explain the results.
4. Modify the program in such a way that periodic boundary conditions are used in one direction only. What happens?
5. Modify the program in such a way that a certain fraction of the particles are “frozen”. Investigate the influence of the fraction of frozen particles on the diffusivity.

Chapter 3

Basic Monte Carlo techniques

3.1 Calculation of π

Consider a circle of diameter d surrounded by a square of length l ($l \geq d$). Random coordinates within the square are generated. The value of π can be calculated from the fraction of points that fall within the circle.

Questions:

1. How can π be calculated from the fraction of points that fall in the circle ? Remark: the “exact” value of π can be computed numerically using $\pi = 4 \times \arctan(1)$.
2. Complete the small Monte Carlo program to calculate π using this method.
3. How does the accuracy of the result depend on the ratio l/d and the number of generated coordinates ? Derive a formula to calculate the relative standard deviation of the estimate of π .
4. Is it a good idea to calculate many decimals of π using this method ?

3.2 The photon gas

The average occupancy number of state j of the photon gas ($\langle n_j \rangle$) can be calculated analytically; see equation 14.13. However, it is also possible to compute this quantity using a Monte Carlo scheme. In this exercise, we will use the following procedure to calculate $\langle n_j \rangle$:

1. Start with an arbitrary n_j
2. Decide at random to perform a trial move to increase or decrease n_j by 1.
3. Accept the trial move with probability

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta(U(n) - U(o))]) \quad (3.1)$$

Of course, n_j cannot become negative !

Questions:

1. How can this scheme obey detailed balance when $n_j = 0$? After all, n_j can not become negative !
2. Is the algorithm still correct when trial moves are performed that change n_j with a random integer from the interval $[-5, 5]$? What happens when only trial moves are performed that change n_j with either -3 or $+3$?
3. Assume that $N = 1$ and $\epsilon_j = \epsilon$. Write a small Monte Carlo program to calculate $\langle n_j \rangle$ as a function of $\beta\epsilon$. Compare your result with the analytical solution.
4. Modify the program in such a way that the averages are updated only after an accepted trial move. Why does this lead to erroneous results ? At which values of β does this error become more pronounced ?
5. Modify the program in such a way that the distribution of n_j is calculated as well. Compare this distribution with the analytical distribution.

3.3 Hard disks in a square

Consider a system of N identical hard disks with diameter 1 in a square of size 10. No periodic boundary conditions are applied. The energy of this system (U) is either $U = 0$ (no overlaps) or $U = \infty$ (at least one overlap). We would like to calculate the radial distribution function of the disks ($g(r)$, $r < 5$) in the canonical ensemble. There are 2 methods of calculating this:

1. Dynamic scheme.
The disks are initially placed on a lattice and trial moves are performed to translate the disks. A trial move is accepted when there are no overlaps and rejected when there are one or more overlaps.
2. Static Scheme.
In every cycle, all disks are put at a completely random position in the system. The radial distribution function of this configuration is calculated and multiplied by the Boltzmann factor of the system (0 when there is at least one overlap and 1 when there are no overlaps).

Questions:

1. Write a program to calculate $g(r)$ using both methods (see *sample.f*). Make sure that both methods give the same result when N is small. Use the *MOLMOL* program to visualize the Monte Carlo simulation.
2. At which N does the second method starts to fail ? Why ?
3. For which method will there be a larger correlation between successive elements of the Markov chain ? Explain.

3.4 MC of a Lennard-Jones system

In this exercise, we will study a 3D Lennard-Jones system. See also Case Study 1 from ref. [2].

Questions:

1. In the present code, the pressure of the system is not calculated. Modify the code in such a way that the average pressure can be calculated. You will only have to make some modifications in the subroutine *ener.f*.
2. Perform a simulation at $T = 2.0$ and various densities. Up to which density does the ideal gas law

$$\beta p = \rho \quad (3.2)$$

hold ?

3. The program produces a sequence of snapshots of the state of the system. Try to visualize these snapshots using the program *MOLMOL*.
4. For the heat capacity at constant volume one can derive

$$C_v = \frac{\langle U^2 \rangle - \langle U \rangle^2}{k_B T^2} \quad (3.3)$$

in which U is the total energy of the system. Derive a formula for the dimensionless heat capacity. Modify the program (only in *mc.nvt.f*) in such a way that C_v is calculated.

5. Instead of performing a trial move in which only one particle is displaced, one can do a trial move in which all particles are displaced. Compare the maximum displacements of these moves when 50% of all displacements are accepted.
6. Instead of using a uniformly distributed displacement, one can also use a Gaussian displacement. Does this increase the efficiency of the simulation ?

3.5 Scaling

Consider a system in which the energy is a function of one variable (x) only:

$$\exp [-\beta U(x)] = \theta(x) \theta(1-x) \quad (3.4)$$

in which $\theta(x)$ is the Heaviside step function: $\theta(x < 0) = 0$ and $\theta(x > 0) = 1$. We would like to calculate the distribution of x in the canonical ensemble. We will consider two possible algorithms (we will use $\delta > 0$):

1. Generate a random change in x between $[-\delta, \delta]$. Accept or reject the new x according to its energy.
2. Generate a random number ϕ between $[1, 1 + \delta]$. With a probability of 0.5, decide to invert ϕ or not. The new value of x is obtained by multiplying x with ϕ .

Questions:

1. Derive the correct acceptance/rejection rules for both schemes.
2. Complete the computer code to calculate the probability density of x . The program writes this distribution to *distri.dat*.
3. What happens when the acceptance rule of method 1 is used in the algorithm of method 2 ? Why ?

3.6 Ewald summation

An example of long-range interactions is the coulombic potential between point charges:

$$U(r_{ij}) = \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \quad (3.5)$$

Special techniques like the Ewald summation are required to compute the total potential energy correctly. When the Ewald summation is used, the total potential energy (U_t) consists of three contributions: the fourier space part (U_f), the real space part (U_r), and the self energy (U_s):

$$\begin{aligned} U_t &= U_f + U_r + U_s \\ U_f &= \frac{1}{2V\epsilon_0} \sum_{\mathbf{k} \neq 0} \frac{\exp[-k^2/4\alpha^2]}{k^2} \left| \sum_{i=1}^{i=N} \exp[-i\mathbf{k} \cdot \mathbf{r}_i] \right|^2 \\ U_r &= \frac{1}{4\pi\epsilon_0} \sum_{i < j} \frac{q_i q_j \operatorname{erfc}(\alpha r_{ij})}{r_{ij}} \\ U_s &= -\frac{1}{4\pi\epsilon_0} \sum_{i=1}^{i=N} \frac{q_i^2 \alpha}{\sqrt{\pi}} \end{aligned} \quad (3.6)$$

in which N is the number of ions and V is the volume of the (rectangular) unit-cell. The positions of the ions in the unit-cell are indicated by \mathbf{r}_i . The complementary error function ($\operatorname{erfc}(x) = \frac{2}{\sqrt{\pi}} \times \int_x^\infty dt \exp[-t^2]$) falls to zero with increasing x . Details of the Ewald summation can be found in ref. [2].

Consider a simple cubic lattice (for example, NaCl) in which the cations are located at $[0, 0, 0]$, $[1, 1, 0]$, $[1, 0, 1]$, and $[0, 1, 1]$ and the anions are located at $[0, 0, 1]$, $[1, 1, 1]$, $[1, 0, 0]$, and $[0, 1, 0]$. The total energy of an infinitely large NaCl crystal equals:

$$U = -\frac{nq^2M}{4\pi\epsilon_0 r_0} \quad (3.7)$$

in which r_0 is the closest distance between ion pairs, n is the number of ion pairs and M is the Madelung constant.

Questions:

1. What is the relation between U_t and the coulombic virial ?
2. How many ion pairs are present in a unit-cell of NaCl ?

3. Compute the Madelung constant of NaCl using the given program. Make a plot of the total energy as a function of \mathbf{k} for different values of α . What is a good choice for α and \mathbf{k} ? You will have to program the real space part of the Ewald summation yourself (in *realspace.f*). Note: We will use $r_0 = 1$, $4\pi\epsilon_0 = 1$ and $q_{\text{Na}^+} = -q_{\text{Cl}^-} = 1$.
4. Rewrite the program in such a way that each ion is given a random displacement from the interval $[-0.3, 0.3]$ (this may correspond to a liquid structure). Make a plot of the total energy as a function of \mathbf{k} for different values of α .
5. Consider a system of dipolar molecules AB ($q_A = -q_B \neq 0$). What will happen with the radial distribution function when a truncated and shifted coulombic potential is used instead of the Ewald summation?

3.7 Parallel tempering

The method of parallel tempering [11–14] is a Monte Carlo scheme that has been derived to achieve good sampling of systems that have a free energy landscape with many local minima.

In parallel tempering we consider N systems. In each of these systems we perform a simulation in the canonical ensemble, but each system is in a different thermodynamic state. Usually, but not necessarily, these states differ in temperature. In what follows we assume that this is the case. Systems with a sufficiently high temperature pass all barriers in the system. The low temperature systems, on the other hand, mainly probe the local energy minima. The idea of parallel tempering is to include MC trial moves that attempt to “swap” systems that belong to different thermodynamic states, *e.g.*, to swap a high temperature system with a low temperature system. If the temperature difference between the two systems is very large, such a swap has a very low probability of being accepted. This is very similar to particle displacement in ordinary Monte Carlo. If one uses a very large maximum displacement a move has a very low probability of being accepted. The solution to this problem is to use many small steps. In parallel tempering we use intermediate temperatures in a similar way. Instead of making attempts to swap between a low and a high temperature, we swap between ensembles with a small temperature difference.

The total partition function of a system with N canonical subsystems (Q) equals

$$Q = \prod_{i=1}^{i=N} Q_i \quad (3.8)$$

in which Q_i is the canonical partition function of the individual system i

$$Q_i = \sum_{\mathbf{x}_i} \exp[-\beta_i U(\mathbf{x}_i)] \quad (3.9)$$

in which $\beta_i = 1/(k_B T_i)$. For each of these systems, individual trial moves are performed. After a randomly selected number of trial moves, an attempt is made to exchange configurations. Two systems (i and j , $|i - j| = 1$) are selected at random, the systems are exchanged by choosing $\mathbf{x}_i(n) = \mathbf{x}_j(o)$ and $\mathbf{x}_j(n) = \mathbf{x}_i(o)$. The ratio of acceptance probabilities equals

$$\frac{\text{acc}(o \rightarrow n)}{\text{acc}(n \rightarrow o)} = \exp[(\beta_i - \beta_j) \times (U(\mathbf{x}_i(o)) - U(\mathbf{x}_j(o)))] \quad (3.10)$$

Such trial moves will be accepted when there is enough overlap between the energies of systems i and j . To demonstrate this technique, consider a two dimensional system of 9 particles that are

confined in a square ($\rho = 2.5$). Each particle pair interact with a soft repulsive potential:

$$U(r) = \begin{cases} \epsilon (r - 1)^2 & r \leq 1 \\ 0 & r > 1 \end{cases} \quad (3.11)$$

We will use $\epsilon = 1$. At low temperatures, the particles are not able to pass each other and therefore a particle is confined at its original position. However, in principle the distribution of the position of a particle should be symmetrical.

Questions:

1. Derive equation 3.10.
2. Compute the distribution of the position of the first particle as well as the distribution of the total energy for $T = 0.001$ when no exchange moves are allowed ($N = 1$). At which temperature are the particles allowed to diffuse?
3. Complete the code for the exchange moves and find out how many systems are needed to ensure that for $T = 0.001$ the particle is allowed to diffuse. You will have to check the distribution of the total energy to ensure that there is enough overlap between the systems.
4. At high temperature, confirm that the equilibrium distributions with and without exchange moves are identical.

Chapter 4

Basic Molecular Dynamics techniques

4.1 MD of a Lennard-Jones system

Enclosed is a Molecular Dynamics (MD) program for a Lennard-Jones fluid in the NVE ensemble. Unfortunately, the program does not conserve the total energy because it contains three errors.

Questions:

1. Find the three errors in the code. The person that spots the errors first will receive a bottle of French wine from Berend Smit (people from Berend's group are excluded from this contest). Hint: there are two errors in *integrate.f* and one in *force.f*. See the file *system.inc* for documentation about some of the variables used in this code.
2. How is one able to control the temperature in this program ? After all, the total energy of the system should be constant (not the temperature).
3. To test the energy drift ΔU of the numerical integration algorithm for a given time step Δt after N integration steps, one usually computes [15]

$$\Delta U (\Delta t) = \frac{1}{N} \sum_{i=1}^{i=N} \left| \frac{U(0) - U(i\Delta t)}{U(0)} \right| \quad (4.1)$$

In this equation, $U(x)$ is the total energy (kinetic+potential) of the system at time x . Change the program (only in *mdloop.f*) in such a way that ΔU is computed and make a plot of ΔU as a function of the time step. How does the time step for a given energy drift change with the temperature and density ?

4. One of the most time consuming parts of the program is the calculation of the nearest image of two particles. In the present program, this calculation is performed using an *if-then-else-endif* construction. This works only when the distance between two particles is smaller than 1.5 and larger than -1.5 times the size of the periodic box. A way to overcome this problem is to use a function that calculates the nearest integer $nint$

$$x = x - box * nint(x * ibox) \quad (4.2)$$

in which $ibox = 1.0/box$. Which expression is faster ? (Hint: You only have to make some modifications in *force.f*) Which expression will be faster on a vector computer like a Cray

C90 ? Because the `nint` function is usually slow, you can write your own `nint` function. For example, when $x < -998$, we can use

$$\text{nint}(x) = \text{int}(x + 999.5) - 999 \quad (4.3)$$

What happens with the speed of the program when you replace the standard `nint` function ? Do you have an explanation for this ?

5. In equation 4.2, `ibox` is used instead of `1/box`. Why ?
6. An important quantity of a liquid or gas is the so called self diffusivity D . There are two methods to calculate D :

(a) by integrating the velocity autocorrelation function:

$$\begin{aligned} D &= \frac{1}{3} \int_0^\infty \langle \mathbf{v}(t) \cdot \mathbf{v}(t+t') \rangle dt' \\ &= \frac{\int_0^\infty \sum_{i=1}^N \langle \mathbf{v}(i,t) \cdot \mathbf{v}(i,t+t') \rangle dt'}{3N} \end{aligned} \quad (4.4)$$

in which N is the number of particles and $\mathbf{v}(i,t)$ is the velocity of particle i at time t . One should choose t in such a way that independent time origins are taken, *i.e.* $t = i\alpha\Delta t$, $i = 1, 2, \dots, \infty$ and $\langle \mathbf{v}(t) \cdot \mathbf{v}(t + \alpha\Delta t) \rangle \approx 0$ (why ?).

(b) by calculating the mean square displacement:

$$D = \lim_{t' \rightarrow \infty} \frac{\langle |\mathbf{x}(t+t') - \mathbf{x}(t)|^2 \rangle}{6t'} \quad (4.5)$$

One should be very careful with calculation of the mean square displacement when particles are always transformed to the central box (why ?).

Modify the program in such a way that the self diffusivity can be calculated using both methods. Only modifications in subroutine *sample_diff.f* are needed. Why is it important to use only independent time origins for the calculation of the mean square displacement and the velocity autocorrelation function ? What is the unit of D in SI units ? How can one transform D into dimensionless units ?

7. For Lennard-Jones liquids, Naghizadeh and Rice report the following equation for the self diffusivity (dimensionless units, $T^* < 1.0$ and $p^* < 3.0$) [16]

$$^{10}\log(D^*) = 0.05 + 0.07p^* - \frac{1.04 + 0.1p^*}{T^*} \quad (4.6)$$

Try to confirm this equation with simulations. How can one translate D^* to a diffusivity in SI units ?

8. Instead of calculating the average energy $\langle U \rangle$ directly, one can use the radial distribution function $g(r)$. Derive an expression for $\langle U \rangle$ using $g(r)$. Compare this calculation with a direct calculation of the average energy. A similar method can be used to compute the average pressure.

*The result will strongly depend on the computer/compiler that is used.

9. In the current version of the code, the equation of motion are integrated by the Verlet algorithm. Make a plot of the energy drift ΔU for the following integration algorithms [2]:
 - Verlet
 - Velocity Verlet [17]
 - Euler (never use this one except here !!!)

4.2 Parallel MD

Please find enclosed a parallel version of the code of the previous exercise. This code is written in FORTRAN77 using an MPI library. We will use LAM [18] for this. To run the program in parallel, several things have to be done:

1. Copy the file *mpif.h* from $\${LAMHOME}/h$. This file is strongly dependent on the version of MPI.
2. Compile the code using LAM. The commands *hf77* and *hcc* will use the GNU compilers *g77* and *gcc* to compile the source.
3. Find out on how many computers you would like to run the code. The user must be able to remotely execute on the machine with *rsh*. Remote host permission must be provided in either */etc/hosts.equiv* or the remote user's *~/.rhosts* file. The remote user's shell must have a search path that will locate LAM executables and the remote shell's startup file must not print anything to standard error when invoked non-interactively.
4. Type *recon -v hostnames* in which the file *hostnames* should contain the hostnames of all machines that will be used.
5. When no errors occur, you can start the LAM daemon by typing *lambboot -v hostnames*. When this is successful too, you can start the run-script. Beware that the command *mpirun* uses an absolute path !
6. After the execution is completed, the LAM daemon can be killed by typing *wipe -v hostnames*.

Questions:

1. Try to find out which part of the calculation is performed in parallel.
2. In the file *force.f*, there is a line:


```
do I=(Iirank+1),(Npart-1),Iisize
```

 In principle, the program also works correct when this line is replaced by:


```
do I=((Iirank*(Npart-1)/Iisize) + 1),((Iirank+1)*(Npart-1)/Iisize)
```

 What is the difference between these two methods and how is this reflected in the scaling of the code ? Hint: *Npart* is the number of particles, *Iisize* is the number of processors and *Iirank* is the index of a particular processor (*Iirank* = 0, 1, 2, \dots , *Iisize* - 1).
3. Run the code on different numbers of nodes and record the execution time as a function of the number of nodes. What happens when using a large number of nodes ?

4. Perform this calculation again for a much larger system at the same density. What do you see ?
5. Can you come up with a better strategy to parallelize this code ?
6. Can the same strategy be used for a MC simulation of the same system in the canonical ensemble ?

Chapter 5

MD and MC in various ensembles

5.1 Barrier crossing (Part 1)

Consider the movement of a single particle that moves on a 1D potential energy surface with the following functional form:

$$U(x) = \begin{cases} \epsilon B x^2 & x < 0 \\ \epsilon (1 - \cos(2\pi x)) & 0 \leq x \leq 1 \\ \epsilon B (x - 1)^2 & x > 1 \end{cases} \quad (5.1)$$

The energy, force and the derivative of the force are continuous functions of the position x and $\epsilon > 0$.

Questions:

1. Derive an expression for B . Make a sketch of the energy landscape.
2. A program is provided that integrates the equation of motion of the particle starting at $x(t=0) = 0$ using several methods:
 - (a) No thermostat (NVE ensemble). What do you expect the phase space trajectories to look like ?
 - (b) Andersen thermostat. In this method, the velocity of the particle is coupled to a stochastic heat bath which leads to a canonical distribution.
 - (c) A Nosé-Hoover chain [15, 19]. In this method, the motion of the particle is coupled to a chain of thermostats. The equations of motion are integrated using an explicit time-reversible algorithm that might look a little-bit complicated at first sight [15], see *integrate_res.f*. One can prove that this method yields a canonical distribution provided that the system is ergodic.
 - (d) No molecular dynamics, but a simple Monte Carlo scheme.

The Andersen thermostat and the NVE integration algorithm are not implemented yet, so you will have to do this yourself (see *integrate_nve.f* and *integrate_and.f*). Try to use all methods for a low temperature, $T = 0.05$, for which the system behaves like a harmonic oscillator. Pay special attention to the following:

- (a) Why does the phase space distribution of the MC scheme look so much different at low temperatures ?

- (b) Why does the phase space distribution of the NVE scheme look like a circle ?
 - (c) Compare the phase space distributions of the Nosé-Hoover chain method with distribution generated by the Andersen thermostat. How long has the Nosé-Hoover chain to be to obtain a canonical distribution ?
3. Investigate at which temperature the particle is able to cross the energy barrier.
 4. Another widely used algorithm is the “temperature coupling” of Berendsen [20]. It is important to note that this method does not produce a canonical ensemble and therefore we should never use it. In this algorithm, the temperature of the system is controlled by scaling the velocities every time step with a factor λ

$$\lambda = \left[1 + \frac{\Delta t}{\tau_T} \left(\frac{T_0}{T} - 1 \right) \right]^{\frac{1}{2}} \quad (5.2)$$

in which T_0 is the desired temperature, T is the actual temperature, Δt is the time step of the integration algorithm and τ_T is a constant. The temperature coupling algorithm can be used in combination with a Leap-Frog algorithm

$$\begin{aligned} v \left(t + \frac{\Delta t}{2} \right) &= \lambda \left(t - \frac{\Delta t}{2} \right) \times \left(v \left(t - \frac{\Delta t}{2} \right) + \frac{F(t)}{m} \Delta t \right) \\ x(t + \Delta t) &= x(t) + v \left(t + \frac{\Delta t}{2} \right) \Delta t \end{aligned} \quad (5.3)$$

Compare the distributions of the Berendsen temperature bath with the canonical distributions.

5. Modify the program in such a way that the potential energy function

$$U = \epsilon (1 - \cos(2\pi x)) \quad (5.4)$$

is used. Calculate the diffusion coefficient as a function of the temperature. Why is it impossible to calculate the diffusivity at low temperatures using ordinary molecular dynamics ? Why is the diffusion coefficient obtained by using the Andersen thermostat a function of the collision frequency ?

5.2 MC in the NPT ensemble

Enclosed is a program to simulate hard spheres (diameter 1) in the NPT ensemble using MC.

Questions:

1. Why is it impossible to calculate the virial for this system directly ?
2. In the current code, a random walk is performed in $\ln(V)$ instead of V . Change the code in such a way that a random walk in V is performed. Check that the average densities calculated by both algorithms are equal.
3. Make a plot of the acceptance ratio for volume displacements as a function of the maximum volume displacement for both algorithms.

5.3 Ising model

In this exercise we consider a 2D Ising model. In this model, N spins s (± 1) are arranged on a lattice. Every spin (i) has 4 neighbors ($j = 1, 2, 3, 4$). The total energy of the system equals

$$U = -\frac{\epsilon}{2} \sum_{i=1}^{i=N} \sum_{j=1}^{j=4} s_i s_j \quad (5.5)$$

in which $s_i = \pm 1$ and $\epsilon > 0$. The second summation is a summation over all spin pairs of spin i . The total magnetization M equals the sum over all spins:

$$M = \sum_{i=1}^{i=N} s_i \quad (5.6)$$

The 2D Ising model has a critical point close to $\beta_c \approx 0.44$.

Questions:

1. Complete the given simulation code for this system (see *ising.f*).
2. Calculate the distribution of M for $N = 32 \times 32$ and $\beta = 0.5$ in the canonical ensemble. In principle, this distribution should be symmetrical:

$$p(M) = p(-M) \quad (5.7)$$

Why does this not seem to be the case ?

3. Instead of a simulation in the canonical ensemble, one can perform the simulation in the ensemble π :

$$\pi \propto \exp[-\beta U + W(M)] \quad (5.8)$$

The average value of an observable O in the canonical ensemble equals:

$$\langle O \rangle = \frac{\langle O \exp[-W(M)] \rangle_{\pi}}{\langle \exp[-W(M)] \rangle_{\pi}} \quad (5.9)$$

in which $\langle \dots \rangle_{\pi}$ is an ensemble average in the ensemble π . Derive this equation.

4. Perform simulations with some given distributions $W(M)$ (*w.type1.dat* and *w.type2.dat*). Explain your results. How should one choose the function $W(M)$ to obtain the optimal efficiency ?
5. What happens when $W(M)$ is a Gaussian,

$$W(M) = A \exp \left[- \left(\frac{M}{\sigma} \right)^2 \right] \quad (5.10)$$

with $A > 0$?

6. What happens when $W(M) = W(U) = -\beta U$?

Chapter 6

Phase equilibrium and free energy calculations

6.1 Vapor-liquid equilibrium

In this exercise, we will use Widom's test particle method to locate a vapor-liquid equilibrium. This is compared with a Gibbs-ensemble simulation.

Questions:

1. Modify the Monte Carlo program of Lennard-Jones particles in the NVT ensemble (only in the file *mc.nvt.f*) in such a way that the chemical potential can be calculated using Widom's test particle method:

$$\mu = \mu_0 - \frac{\ln(\rho^{-1} \langle \exp[-\beta \Delta U^+] \rangle)}{\beta} \quad (6.1)$$

in which ρ is the number of particles per volume, U^+ is the energy of a test particle and

$$\mu_0 = \frac{-\ln(\Lambda^3)}{\beta} \quad (6.2)$$

- Make a plot of the chemical potential and pressure as a function of the density for $T = 0.8$.
 - Why is it more difficult to calculate the chemical potential at high densities than at low densities ?
 - How can you locate the vapor-liquid coexistence densities ?
2. Perform a Gibbs-ensemble simulation of the system at $T = 0.8$. In the Gibbs ensemble, the chemical potential of box i is equal to [2,21]

$$\mu_i = \mu_0 - \frac{\ln \left\langle \frac{V_i}{n_i + 1} \exp[-\beta \Delta U_i^+] \right\rangle}{\beta} \quad (6.3)$$

in which n_i is the number of particles in box i and V_i is the volume of box i . Do the vapor/liquid density and chemical potential agree with your previous results ?

6.2 Umbrella sampling

Consider a single particle on a one dimensional energy landscape. The energy as a function of the position is given by

$$U(x) = \epsilon x^2 \quad (6.4)$$

One would like to calculate the probability distribution of finding the particle at a position x ($p(x)$) by using a Monte-Carlo scheme (we will use $\beta = \epsilon = 1$).

Questions:

1. Why is this distribution difficult to calculate for large values of x when a conventional MC scheme is used ?
2. Alternatively, one can divide the x axis in overlapping slices and calculate the distribution $p(x, i)$ in all slices i . This scheme is often referred to as umbrella sampling. Show that $p(x, i) \propto p(x, j)$ when $i \neq j$ (This also means that $\ln(P(x, i)) = \ln(P(x, j)) + C$ in which C is a constant).
3. Compare the results from the conventional MC algorithm with the results from the umbrella sampling simulations, especially at large values of x . Different slices can be combined by using *xmgr*.

Chapter 7

Rare events

7.1 Barrier crossing (Part 2)

In one of the previous exercises, we have seen that sometimes there may be energy barriers in a system that are so high, that the crossing rate can not be calculated using conventional molecular dynamics techniques. A method that can be used to calculate the transition rate is the Chandler-Bennett approach [2,4,22,23]. We will use this approach to calculate the crossing rate of a single particle over an energy barrier

$$U(x) = \begin{cases} \infty & x < -5 \\ 0 & -5 \leq x < 0 \\ \epsilon (1 - \cos(2\pi x)) & 0 \leq x \leq 1 \\ 0 & 1 < x \leq 5 \\ \infty & x > 5 \end{cases} \quad (7.1)$$

We will call the region left from the barrier the reactant side A and the region right from the energy barrier the product side B. The time dependent rate constant $k_{A \rightarrow B}(t)$ can be written as [2]

$$k_{A \rightarrow B}(t) = \frac{\langle \dot{q}(0) \delta(q^* - q(0)) \theta(q(t) - q^*) \rangle}{\langle \delta(q^* - q(0)) \rangle} \times \frac{\langle \delta(q^* - q) \rangle}{\langle \theta(q^* - q) \rangle} \quad (7.2)$$

We are mainly interested at the plateau value of $k_{A \rightarrow B}(t)$, which is the hopping rate. At first sight, this equation might look pretty horrible. First of all, we need to define our symbols:

- q is the reaction coordinate. In this case, $q = x$ is a sensible choice. $q(0)$ is the reaction coordinate at $t = 0$.
- q^* is the position of the dividing surface. We will choose $0 < q^* < 1$. Note that the dividing surface does not always have to be on top of the energy barrier !
- $\langle \dots \rangle$ denotes an ensemble average
- $\delta(x)$ is Dirac's delta function: $\delta(x = 0) = \infty$, $\delta(x \neq 0) = 0$, $\int_{-\infty}^{\infty} \delta(x) dx = 1$ and

$$\delta(h(s)) = \frac{\delta(s - s_0)}{|h'(s)|} \quad (7.3)$$

when $h(s_0) = 0$.

- $\theta(x)$ is the Heaviside step function: $\theta(x < 0) = 0$ and $\theta(x > 0) = 1$. Note that the definition of θ in ref. [2] on page 251 is incorrect.

Now it is a lot easier to understand what the terms in equation 7.2 mean:

- The first term on the r.h.s. is the conditional average of the product $[\dot{q}(0) \theta(q(t) - q^*)]$ given that the initial position is at the top of the barrier ($q(0) = q^*$). The term $\theta(q(t) - q^*)$ is equal to 1 when the particle is in state B at time t and equal to 0 otherwise. $\dot{q}(0)$ is the initial velocity of the particle at the top of the barrier.
- The second term on the r.h.s is the probability of finding the system on top of the barrier divided by the probability that the system is on the reactant side of the barrier.

Questions:

1. What is the unit of $k_{A \rightarrow B}(t)$ in SI units and in dimensionless units ?
2. Is the force on a particle a continuous function of x ?
3. A program is provided to perform a Molecular Dynamics simulation for a single particle provided that $q(0) = q^*$. The sampling of the crossing rate has still to be programmed (see *mdloop.f* and *sample.f*).
4. Calculate the crossing rate for different temperatures and positions of q^* . Make a plot of $\ln(k_{A \rightarrow B})$ versus T^{-1} .
5. Check that the crossing rate is independent of the location of the dividing surface.

7.2 Transition Path Sampling

In this section, we will study the Transition Path Sampling method for the calculation of rate constants of rare events. This method is able to compute rate constants without having to assume a transition state. For more details about this method, the reader is referred to ref. [24,25], on which this exercise is based.

7.2.1 Introduction

Consider a dynamical system with two stable states, A and B, in which transitions from A to B are rare. The transition rate, k , from A to B can be calculated from the time derivative of an autocorrelation function $C(t)$,

$$k = \frac{dC(t)}{dt} \quad t_{\text{mol}} < t \ll t_{\text{rxn}} \quad (7.4)$$

$$C(t) = \frac{\langle h_A(x_0) h_B(x_t) \rangle}{\langle h_A(x_0) \rangle}, \quad (7.5)$$

provided that the reaction time t_{rxn} of the system [A, B] is much larger than the molecular relaxation time t_{mol} of the system in region A or B. In equation 7.5, x_t represents the momenta \mathbf{p} and positions \mathbf{q} of the system at time t . We will only consider deterministic trajectories for which x_t is completely determined by the initial conditions x_0 , i.e. $x_t = x_t(x_0)$. The functions h_A and

h_B characterize the regions A and B; $h_{A,B}(x) = 1$ when $x \in A, B$, respectively, and $h_{A,B}(x) = 0$ otherwise. Note that A and B must be chosen in such a way that $A \cap B = \emptyset$.

Since the function x_t is fully determined by the initial condition x_0 , the ensemble averages in equation 7.5 can be written as an integration over the initial conditions weighted with the equilibrium distribution $\mathcal{N}(x_0)$,

$$C(t) = \frac{\int dx_0 \mathcal{N}(x_0) h_A(x_0) h_B(x_t(x_0))}{\int dx_0 \mathcal{N}(x_0) h_A(x_0)}. \quad (7.6)$$

We can also look at this equation as the ensemble average of $h_B(x_t)$ weighted with the equilibrium distribution $\mathcal{N}(x_0) \times h_A(x_0)$. In other words, $C(t)$ is the fraction of trajectories that start in A with distribution $\mathcal{N}(x_0)$ and reach B after time t . Since we are sampling over paths this ensemble is called the path ensemble. A procedure to sample this ensemble would be to perform a MD simulation to generate a new path of length t and subsequently use a MC procedure to decide whether to accept or reject this new path. In this way, we generate an ensemble of paths which we can use to compute ensemble averages. We will only consider a micro-canonical ensemble of initial conditions x_0 , *i.e.*

$$\mathcal{N}(x_0) = \delta(\mathcal{N}(x_0) - E) \quad (7.7)$$

in which E is the total energy. In principle we could compute $C(t)$ from an “ordinary” path ensemble simulation. This would imply that we generate an ensemble of paths of length t that start at A and we would count all paths that are at time t in B. However, since the transition from A to B is a rare event, the number of paths that ends in B is so small that such an approach would require very long simulations. Therefore, we need to help the system explore the regions of interest.

Suppose that region B can be defined by the value of an order parameter λ ; $x_t \in B$ if $\lambda_{\min} \leq \lambda(x_t) \leq \lambda_{\max}$. For equation 7.5, we may write

$$C(t) = \frac{\int dx_0 \exp[-\beta \mathcal{H}(x_0)] h_A(x_0) h_B(x_t(x_0))}{\int dx_0 \exp[-\beta \mathcal{H}(x_0)] h_A(x_0)} = \int_{\lambda_{\min}}^{\lambda_{\max}} d\lambda P(\lambda, t), \quad (7.8)$$

in which

$$\begin{aligned} P(\lambda, t) &= \frac{\int dx_0 \exp[-\beta \mathcal{H}(x_0)] h_A(x_0) \delta[\lambda - \lambda(x_t(x_0))]}{\int dx_0 \exp[-\beta \mathcal{H}(x_0)] h_A(x_0)} \\ &= \frac{\int dx_0 f(x_0, t) \delta[\lambda - \lambda(x_t(x_0))]}{\int dx_0 f(x_0, t)}. \end{aligned} \quad (7.9)$$

$P(\lambda, t)$ can be interpreted as the probability for the system to be in a state with a certain λ after time t given that the system is in A at time 0. Because $P(\lambda, t)$ is quite small in B (*i.e.* transitions from A to B are rare), special techniques such as umbrella sampling [2,3] are required to compute $P(\lambda, t)$. As shown in refs. [24,25], it is advantageous to rewrite $C(t)$ as

$$C(t) = C(t') \times \frac{\langle h_B(t) \rangle_{F(x_0, T)}}{\langle h_B(t') \rangle_{F(x_0, T)}}, \quad (7.10)$$

in which $t, t' \in [0, T]$ and

$$\begin{aligned} F(x_0, T) &= \exp[-\beta \mathcal{H}(x_0)] h_A(x_0) H_B(x_0, T) \\ H_B(x_0, T) &= \max_{0 \leq t \leq T} h_B(x_t(x_0)). \end{aligned} \quad (7.11)$$

The distribution $F(x_0, T)$ can be interpreted as the ensemble of trajectories starting in A and visiting B at least once in the time interval $[0, T]$. In this way, one has to perform only a single transition path sampling calculation of $C(t')$ when calculating the time derivative of $C(t)$,

$$k = \frac{dC(t)}{dt} = \frac{C(t')}{\langle h_B(t') \rangle_{F(x_0, T)}} \times \frac{d \left[\langle h_B(t) \rangle_{F(x_0, T)} \right]}{dt}, \quad (7.12)$$

while the functions $\langle h_B(t) \rangle_{F(x_0, T)}$ and $\langle h_B(t') \rangle_{F(x_0, T)}$ can be calculated from a (single) separate simulation.

7.2.2 MC sampling from the distribution $F(x_0, T)$

In the transition path sampling method, transition pathways are harvested by sampling the path ensemble $F(x_0, T)$ with a MC procedure. In this subsection, we will present two types of MC trial moves to generate a new path from an existing one to sample the distribution $F(x_0, T)$. Sampling of the distribution $f(x_0, t)$ is similar. We will use the symbols n and o for the new and old configuration respectively.

Shooting

In a shooting move, first one picks a time t' randomly from the interval $[0, T]$ and one makes an attempt to rotate the old momenta vector \mathbf{p} in such a way that the total energy E is constant (see equation 7.7). The rotation angle is chosen at random from a uniform distribution in a finite interval $[-\Delta, \Delta]$. Second, one has to construct a new path by integrating backward and forward to obtain the new path. To obey detailed balance, the new path has to be accepted with a probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{F(x_0(n), T)}{F(x_0(o), T)} \right) = h_A(x_0(n)) H_B(x_0(n), T) \quad (7.13)$$

Shifting

In a shifting move, one translates the initial conditions in time by an amount Δt :

$$x_0(n) = x_{\Delta t}(x_0(o)). \quad (7.14)$$

To simplify the acceptance rule we choose a symmetric generation probability for Δt ,

$$P_g(\Delta t) = P_g(-\Delta t). \quad (7.15)$$

Due to energy conservation along a trajectory the acceptance rule for this trial move equals

$$\text{acc}(o \rightarrow n) = h_A(x_0(n)) H_B(x_0(n), T). \quad (7.16)$$

Although shifting trial moves do not sample the phase space ergodically because the energy of the path is not changed, they greatly improve statistics.

7.2.3 Model system

To illustrate this method, consider a two-dimensional system consisting of 15 WCA particles:

$$u_{\text{WCA}}(r) = \begin{cases} \epsilon (1 + 4 [r^{-12} - r^{-6}]) & r \leq r_{\text{WCA}} \\ 0 & r > r_{\text{WCA}} \end{cases} \quad (7.17)$$

in which r is the distance between two particles and $r_{\text{WCA}} = 2^{1/6}$. We will use $\epsilon = 1$. However, particles 1 and 2 interact via a double well potential:

$$u_{\text{dw}}(\lambda) = h \left[1 - \frac{(\lambda - w - r_{\text{WCA}})^2}{w^2} \right]^2 \quad (7.18)$$

This potential has stable minima at $\lambda = r_{\text{WCA}}$ and $\lambda = r_{\text{WCA}} + 2w$ that are separated by an energy barrier with height h . When h is large compared to the total energy E the transitions between these minima are rare.

In all simulations, we have used $w = 0.25$ and $E = 9$. We have confined this system of 15 particles in a circle of diameter 6, resulting in a density of 0.53. We have defined region A as all configurations for which $\lambda < 1.30$. We have defined region B as all configurations for which $\lambda > 1.45$.

7.2.4 Questions

1. What is the value of $C(t)$ for $t = 0$ and $t \rightarrow \infty$? Why?
2. Perform conventional MD simulations for different values of the barrier height h and compare the equilibrium distribution of λ . How is $C(t)$ related to this distribution?
3. Calculate the function $C(t)$ for $h = 2$ by transition path sampling and compare the result with a conventional MD simulation. To perform the calculation of $P(\lambda, t)$, you can divide the phase space in five overlapping regions:
 - $0.00 < \lambda_1 < 1.22$
 - $1.20 < \lambda_2 < 1.26$
 - $1.24 < \lambda_3 < 1.30$
 - $1.28 < \lambda_4 < 1.45$
 - $1.40 < \lambda_5 < \infty$

How can one match these distributions?

Chapter 8

Configurational-Bias Monte Carlo

8.1 CBMC of a single chain

In this exercise, we will look at the properties of a single chain molecule. We will compare various sampling schemes. Suppose that we have a chain molecule of length n in which there are the following interactions between beads:

- Two successive beads have a fixed bond-length l . We will use $l = 1$.
- Three successive beads have a bond-bending interaction

$$U = \frac{1}{2} k_t (\theta - \theta_0)^2 \quad (8.1)$$

in which θ is the bond-angle, θ_0 is the equilibrium bond angle and k_t a constant. We will use $\theta_0 = 2.0$ rad ($\approx 114.6^\circ$) and $k_t = 2.0$.

- Every pair of beads that is separated by more than two bonds has a soft repulsive interaction

$$U(r) = \begin{cases} \frac{\Lambda (r - r_{\text{cut}})^2}{r_{\text{cut}}^2} & r \leq r_{\text{cut}} \\ 0 & r > r_{\text{cut}} \end{cases} \quad (8.2)$$

in which r_{cut} is the cut-off radius (we will use $r_{\text{cut}} = 1.0$ and $\Lambda > 0$).

An interesting property of a chain molecule is the distribution of the end-to-end distance, which is the distance between the first and the last segment of the chain. There are several possible schemes to study this property (see also section 3.3):

1. Dynamic schemes.

In a dynamic scheme, a Markov chain of states is generated. The average of a property B is the average of B over the elements of the Markov chain

$$\langle B \rangle \approx \frac{\sum_{i=1}^{i=N} B_i}{N} \quad (8.3)$$

When $N \rightarrow \infty$ the expression is exact. Every new configuration is accepted or rejected using an acceptance/rejection rule:

- When unbiased chains are generated

$$\text{acc}(o \rightarrow n) = \min(1, \exp[-\beta (U(n) - U(o))]) \quad (8.4)$$

in which U is the total energy (soft repulsion and bond-bending) of a chain.

- When Configurational-Bias Monte Carlo (CBMC) [26–28] is used

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{W(n)}{W(o)} \right) \quad (8.5)$$

in which

$$W = \frac{\prod_{i=2}^{i=n} \sum_{j=1}^{j=k} \exp[-\beta U(i, j)]}{k^{n-1}} \quad (8.6)$$

In this equation, k is the number of trial positions and $U(i, j)$ is the energy of the j -th trial position of the i -th chain segment. The term $U(i, j)$ does not contain the bond-bending potential, because that potential has already been used for the generation of the trial positions.

2. Static schemes.

In a static scheme, all generated configurations contribute to the average. To obtain a canonical distribution, we will have to use a weight factor R

$$\langle B \rangle = \frac{\sum_{i=1}^{i=N} B_i \times R_i}{\sum_{i=1}^{i=N} R_i} \quad (8.7)$$

For R_i we can write

- When random chains are generated

$$R_i = \exp[-\beta U_i] \quad (8.8)$$

Here, U_i is the total energy of the chain.

- When CBMC is used

$$R_i = W \quad (8.9)$$

These equations are derived in ref. [2].

Questions:

1. The program is provided to calculate chain properties using these four methods. However, some additional programming has to be done in the file *growf*, which is a subroutine to grow a new chain using either CBMC or random insertion.
2. Compare the end-to-end distance distributions of the four methods. Which method will have the best performance? Investigate how the efficiency of CBMC depends on the number of trial directions (k).
3. Investigate the influence of chain-length on the end-to-end distance distribution. For which chain-lengths do the four methods start to fail?
4. For high temperatures (and for low k_t and Λ), the end-to-end distance distribution looks like the distribution of a non-self-avoiding random walk. This means that the chain segments are oriented completely random and the segments are allowed to overlap. For the mean square end-to-end distance, we can write

$$\frac{\langle r^2 \rangle}{l^2} = \left\langle \left(\sum_{i=1}^{i=n} x_i^2 \right) + \left(\sum_{i=1}^{i=n} y_i^2 \right) + \left(\sum_{i=1}^{i=n} z_i^2 \right) \right\rangle \quad (8.10)$$

in which (x_i, y_i, z_i) are the projection of each segment on the (x, y, z) axis

$$\begin{aligned} x_i &= \sin(\theta_i) \cos(\phi_i) \\ y_i &= \sin(\theta_i) \sin(\phi_i) \\ z_i &= \cos(\theta_i) \end{aligned} \quad (8.11)$$

This set of equations can be reduced to

$$\frac{\langle r^2 \rangle}{l^2} = n \quad (8.12)$$

Questions:

- Derive equation 8.12. Hint: the following equations will be very useful:

$$\begin{aligned} \cos^2(\theta_i) + \sin^2(\theta_i) &= 1 \\ \cos(\theta_i - \theta_j) &= \cos(\theta_i) \cos(\theta_j) + \sin(\theta_i) \sin(\theta_j) \\ \langle \cos(\theta_i - \theta_j) \rangle &= 0 \end{aligned} \quad (8.13)$$

The last equation holds because $\theta_i - \theta_j$ is uniformly distributed.

- Modify the program in such a way that $\langle r^2 \rangle$ is calculated for a non-self-avoiding random walk. Compare your results with the analytical solution.
- Does

$$\langle r^2 \rangle \propto n \quad (8.14)$$

hold for a chain with a potential energy function described in this exercise ? Investigate the influence of Λ on the end-to-end distance distribution.

8.2 CBMC of a simple system

Consider a system with three coordinates (x_1, x_2, x_3) and phase space density

$$\rho(x_1, x_2, x_3) = \exp \left[- \left(x_1^2 + x_2^2 + x_3^2 \right) \right] = \exp \left[-r^2 \right] \quad (8.15)$$

We would like to calculate the average $\langle r^2 \rangle$,

$$\langle r^2 \rangle = \frac{\iiint dx_1 dx_2 dx_3 r^2 \rho}{\iiint dx_1 dx_2 dx_3 \rho} \quad (8.16)$$

by using the CBMC algorithm of Falcioni and Deem [12]:

- Generate k sets of new coordinates B_1, \dots, B_k by adding a random vector to the old configuration (A_1) .
- Select one set (i) with a probability proportional to its Boltzmann factor,

$$p_i = \exp \left[-r_{B_i}^2 \right] \quad (8.17)$$

This leads to a Rosenbluth factor of

$$W(n) = \sum_{j=1}^{j=k} \exp \left[-r_{B_j}^2 \right] \quad (8.18)$$

- Starting from the selected configuration B_i , $k-1$ configurations (A_2, \dots, A_k) are generated by adding a uniform vector to B_i . A_1 is the old configuration. This leads to the Rosenbluth factor of the old configuration

$$W(o) = \sum_{j=1}^{j=k} \exp \left[-r_{A_j}^2 \right] \quad (8.19)$$

- The new configuration B_i is accepted with a probability

$$\text{acc}(o \rightarrow n) = \min \left(1, \frac{W(n)}{W(o)} \right) \quad (8.20)$$

Questions:

1. Make a schematic sketch of the configurations A_1, \dots, A_k and B_1, \dots, B_k . Show that for $k = 1$, this algorithm reduces to the standard Metropolis algorithm for particle displacements.
2. Proof that this algorithm obeys detailed balance.
3. Is it possible to calculate $\langle r^2 \rangle$ analytically ? Hint: for $a > 0$,

$$\int_0^\infty \exp \left[-a^2 x^2 \right] dx = \frac{\sqrt{\pi}}{2a} \quad (8.21)$$

4. Enclosed is a computer program for this CBMC sampling scheme. Unfortunately, the program has to be completed by you (see the file *cbmc.f*) ! Make sure that your estimate of $\langle r^2 \rangle$ is independent of the number of trial directions (k).
5. What happens with the fraction of accepted trial moves when the number of trial directions (k) is increased ? Make of plot of the fraction of accepted trial moves as a function of k for various maximum displacements. Explain your results.
6. Why is this CBMC method very useful when the system is far from equilibrium ?

8.3 Overlapping distribution for polymers

In this exercise, we would like to calculate the chemical potential of a chain of length n in a solvent of monomers. We will use the following potential:

- Two successive beads have a fixed bond-length of 1.
- Every pair of beads of the chain that is separated by more than one bond has a soft repulsive interaction

$$U(r) = \begin{cases} \frac{\Lambda(r-r_{\text{cut}})^2}{r_{\text{cut}}^2} & r \leq r_{\text{cut}} \\ 0 & r > r_{\text{cut}} \end{cases} \quad (8.22)$$

in which r_{cut} is the cut-off radius (we will use $r_{\text{cut}} = 1.0$ and $\Lambda > 0$). The same interactions are used for monomer-monomer and polymer-monomer pair interactions.

The chemical potential (μ) of the chain in the solvent can be calculated from the average Rosenbluth factor when test chains are grown using CBMC:

$$\beta\mu = -\ln \langle W \rangle \quad (8.23)$$

However the same thing can be achieved by removing a real chain from the simulation box. The principle of the overlapping distribution method is to calculate a histogram of the Rosenbluth weight when adding or removing the chain. If these two distributions overlap a reliable estimate of the chemical potential can be obtained. If these distributions do not overlap sampling problems may be expected [2, 29].

By constructing the following functions we can calculate the excess chemical potential *

$$\begin{aligned} f(-\ln(W)) &= \ln(p_0(-\ln(W))) + \frac{1}{2} \ln(W) \\ g(-\ln(W)) &= \ln(p_1(-\ln(W))) - \frac{1}{2} \ln(W) \end{aligned} \quad (8.24)$$

In which p_0 is the probability density of $-\ln(W)$ in the case of adding a chain and p_1 of removing one. The chemical potential is found by subtracting these two functions,

$$\beta\mu = g(-\ln(W)) - f(-\ln(W)) \quad (8.25)$$

Questions:

1. Complete the given program. Additions have to be made in subroutine *overdist.f*.
2. Why is the difference between the functions g and f not exactly constant?
3. Why do we still need two simulations when the length of the chain equals 1?
4. Compare the two methods to compute the chemical potential for different values of the density and the chain length. When does the test particle method fail?

*In the article of Mooij and Frenkel [29] the functions f and g contain some typographic errors and should read as equation 8.24.

Chapter 9

Appendix A: Introduction to GNU/Linux

GNU/Linux [30] is a Unix Operating System (OS) for Intel PC's and other architectures. There are several advantages to use GNU/Linux instead of Windows 98/NT:

- It is free software, including almost all applications such as compilers and editors. Popular GNU/Linux distributions, like Redhat, can be downloaded directly from the web [31].
- It is multiprocessor, multitasking and multiuser.
- There is a strict distinction between OS and user files.
- Unix is available on many different computer platforms, ranging from simple PC's to supercomputers.

There are however also some disadvantages:

- It is a little-bit more difficult to learn than Windows 98/NT.
- The commands can be rather cryptic.
- It is not very suitable if you use it only once in a while.
- It might sometimes be difficult to get immediate support.

To use a Unix computer, one always has to login first. To login, one has to provide a user name (let us assume that the user name is *tampert*) and user-specific password. One will end up in the directory */home/tampert*. This is called the home-directory of user *tampert*.

Several GNU/Linux commands or applications are:

acroread: displays an Adobe *.pdf* file [32]

bzip2: an alternative for *gzip*

cat <file>: display a file

cd directory: changes the directory to *directory*. The current directory is called *.* and the previous one is called *..* *cd tampert* will change the current directory to the directory *tampert*. The command *cd* will change the current directory to the home directory */home/tampert*

compress and *uncompress*: zip or unzip a *.Z* file

diff file1 file2: displays the differences between *file1* and *file2*

emacs: a very nice text editor, especially for writing C/Fortran code

fnchek: a program to detect logical errors in a FORTRAN77 code that the compiler does not detect [33]. To use more than 80 character on a single line, use the options *-columns=131 -portability=all -nopretty*.

g77: GNU FORTRAN77 compiler [9]. Four important compiler options are:

- *-O2* produces optimized code
- *-C -g* makes code suitable to the debugger and checks for array-bound over/underflow.
- *-ffixed-line-length-132* allows the use most than 80 character on a single line. Note that this particular option is strongly machine dependent.
- *-Wall* prints all warnings to the screen

gcc: GNU C compiler [10]

g++: GNU C++ compiler [10]

gdb: GNU debugger. Very useful to analyze core dumps.

ghostview: displays a postscript (*.ps*) file

gnuplot: traditional program to make graphs. We recommend you to use *xmgr* however.

grep text file: find the word *text* in the file *file*

gzip and *gunzip*: zip or unzip a *.gz* file

joe: a simple text editor to manipulate text files. *joe tampert* will edit the text-file *tampert*. Several useful commands are: *Ctrl-K-H* provides a help screen, *Ctrl-K-X* exits with saving the file, *Ctrl-C* exits without saving.

kedit and *nedit*: nice text editors for GNU/Linux

less: an alternative for *more*

logout: logout

ls: shows the content of a directory. *ls -l* shows more information. *ls t** will only show files or directories starting with the character *t*.

make: used to compile programs when a Makefile is provided. To compile a program, simply type *make*.

man command: looks for the manual page of *command*. *man -k keyword* will browse all manual pages for *keyword*. *man man* will show the manual of *man*.

mkdir directory: makes a new directory *directory*

more filename: displays the content of a text-file *filename*

netscape: especially <http://www.garfield.com> is interesting

pwd: displays the current directory

rm filename: removes the file *filename*. There is no undelete command for Unix, so be careful !!! Use *rm -rf* to delete a complete directory structure. Be very careful !!

rmdir directory: removes directory *directory*

startx: start the X (graphical) server of GNU/Linux

tar: tape archive utility. *tar cvf <file>* will archive and *tar xvf <file>* will unpack

vi: this is the standard Unix editor which is available on all Unix systems. There are still people that use it.

xmgr: a nice program to make graphs [34]. To plot more than one graph from a single file, use *xmgr -nxy <file>*. Recently, *xmgr* has evolved into *xmgrace*.

xv: a nice program to display and to convert all kinds of graphic files

Beware of the following:

1. Unix is case sensitive, this means that *TEST* is different from *test*.
2. Unix uses the no-news-is-good-news principle. If a command is executed and the action is successful, there is usually no information provided about the result of the action. Example: *rm t** will remove all files in a directory that start with the character *t*. The command *rm t < space > ** will first try to remove the file *t* and then all files (* means all files). This means that everything in the directory is lost !!! Again, this means lost forever !!!
3. One can use the standard redirection of input/output *> < |*. For example, *cat < file | more* will display *file* and redirect the output to the command *more*. Alternatively, one could use *more file*.

More information about GNU/Linux can be found on the webpage of this course.

Chapter 10

Appendix B: Introduction to FORTRAN77

FORTRAN77 is a very old language (1977). It is relatively simple to use, and the syntax is quite trivial. In this course we will use ANSI FORTRAN77 with a few extensions. We will only mention some of the very basics of FORTRAN. For more information, there are several sources:

- The Unix *man* command. For example, *man nint* will give information about the FORTRAN *nint* command. Unfortunately, the *man* pages of FORTRAN77 commands and functions are not always available.
- The book: Interactive Fortran 77: A Hands on Approach, by Ian Chivers and Jane Sleightholme. It is available on the website of this course.
- The Fortran Market on the web [35].

Some FORTRAN77 keywords/hints:

- All programs start with *PROGRAM program-name*, *SUBROUTINE subroutine-name* or *FUNCTION function-name* and end with *END*.
- The first five characters on a line are reserved for line numbers, the sixth character is reserved for a continuation character. If the first character is a C, the line is just comment.
- FORTRAN77 does not use $>$, \geq , $<$, \leq , $==$, $/=$, but *.gt.*, *.ge.*, *.lt.*, *.le.*, *.eq.*, *.ne.* instead. However, on most compilers $>$, \geq , $<$, \leq will work.
- Variable types: *integer* (-1 0 1 2 3 etc.), *double precision* (1.0d0 -1.0d0 5.0d0 etc., 1.0d7 means 1.0×10^7), logical (*.true.* or *.false.*). These variables are declared at the beginning of the program. If the statement *implicit none* is used, all variables must be declared. We strongly recommend the use of *implicit none*.
- All arrays start at 1 instead of 0 in C/C++. So *integer qq(5)* has the elements 1, 2, \dots , 5, while *integer qq(a:b)* has the elements $a \dots b$ ($b > a$). Negative values of *a, b* are allowed.
- Logical operators: *.and.*, *.or.* and *.not.*
- *if-then-else-endif*, *do-endif* and *do while-endif* construction
- *goto linenum*: jumps to *linenum*. We recommend you to avoid the use of many *goto* statements.

- *stop* terminates a program
- Subroutines: call of a subroutine: *call integrate(tampert)* will call the subroutine *integrate* with the argument *tampert* passed. If the function modifies *tampert* its value is returned. The use of functions is almost similar.
- Functions: $a**b$ equals a^b , *sqr*t (square-root), *sin* (sine), *asin* (arcsine), *cos* (cosine), *acos* (arccosine), *tan* (tangent), *atan* (arctangent), *int* (conversion to integer using truncation), *nint* (conversion to nearest integer), *dble* (conversion to a double precision), *exp* (exponential), *log* (natural logarithm), *log10* (common logarithm), *max* (maximum of two or more numbers), *min* (minimum of two or more numbers), *sinh* (hyperbolic sine), *cosh* (hyperbolic cosine), *tanh* (hyperbolic tangent), *mod* (integer remainder of its first argument divided by its second argument), *abs* (absolute value).
- *common* block. used for global variables.
- *include 'filename'* include a file. The *include* statement is replaced by the content of *filename*. We strongly recommend you to put all *common* blocks in *include* files.
- *save variable-name*: stores the value of variable *variable-name* when the execution of the subroutine is completed.
- The programs *ftnchek* is very useful to detect logical errors in your FORTRAN77 code that are not detected by the compiler [33].

When you use FORTRAN77, there is nearly always an emotional discussion with people that use C/C++. We completely agree that C/C++ is a much nicer language than FORTRAN77. The reasons that we (and with us many other people) still use FORTRAN77 are

- FORTRAN77 is very simple to learn and to use. It will generally take more time to learn C/C++.
- Only a few people that read this manual will be a computer scientist. As physicist or chemist, we want a program that works and not a program that is programmed in a nice way. Physics and chemistry are complicated enough !
- FORTRAN77 compilers still produce faster code than C/C++ compilers. This is due to the longer evolution of the compiler.

Other languages like Pascal, Delphi or Java are either too slow or not available on all Unix platforms so it is quite obvious why almost nobody uses them in the field of molecular simulations. Although FORTRAN77 has already evolved to FORTRAN90, FORTRAN95 and HPF, these compilers are not (yet) part of the GNU project [36] so we have chosen to use an older version.

program test1	this program will calculate
implicit none	sqrt(1.0d0) + ... + sqrt(5.0d0)
 integer i	 declare variables
double precision ee,dd	
 ee = 0.0d0	
do i=1,5	loop over i=1,2,3,4,5
dd = dble(i)	convert i to double precision
call tampert(dd)	call subroutine tampert
ee = ee + dd	
enddo	
write(*,*) ee	print final result
end	end of the program
 subroutine tampert(input)	 subroutine tampert
implicit none	
 double precision input	
input = sqrt(input)	call sqrt(input)
return	return to main program
end	

program test2	this program will calculate
implicit none	the sine of a given input
	all variables have to be declared
 integer i	
logical OK	
double precision value	
 1 write(*,*) 'Give a value'	print header
read(*,*) value	read in a value
if (value.le.0.0d0.or.	test if range is all right
+ value.ge.1.0d0) OK=.false.	
if(.not.ok) stop	if not all right, then terminate
write(*,*) sin(value)	write result
goto 1	return to line 1
end	

Chapter 11

Appendix C: Introduction to *MOLMOL*

Many of the programs used in this course produce *.pdb* files. A *.pdb* file contains a snapshot/movie of the system, which can be visualized using the *MOLMOL* program. To view the movie, one should start *molmol* first. Then choose *File, ReadMol, PDB, system.pdb*. Click *Select all* (this is on the right panel), *Ball/Stick* (right panel as well), *Options, Animation, Start*. Mouse actions: Left button (rotating the system), Middle button (translating the system), right button (zoom in/out). One can also zoom in/out by using the menus. More information about *MOLMOL* can be found on the *MOLMOL* website [37].

Chapter 12

Appendix D: Thermodynamic equations

This section summarizes some basic thermodynamic equations that might be useful while doing the exercises.

Fundamental Functions

$$\text{Energy} \quad U \quad (12.1)$$

$$\text{Enthalpy} \quad H = U + pV$$

$$\text{Helmholtz free energy} \quad F = U - TS$$

$$\text{Gibbs free energy} \quad G = H - TS$$

Sometimes the symbol A is used for the Helmholtz free energy.

Differentials of Fundamental Functions

$$dU = TdS - pdV + \sum_i \mu_i dn_i \quad (12.2)$$

$$dH = TdS + Vdp + \sum_i \mu_i dn_i$$

$$dF = -SdT - pdV + \sum_i \mu_i dn_i$$

$$dG = -SdT + Vdp + \sum_i \mu_i dn_i$$

Maxwell Relations

For example, from $dU = TdS - pdV + \sum_i \mu_i dn_i$ one can derive

$$\left(\frac{\partial T}{\partial V} \right)_{S, n_i} = \left(\frac{\partial p}{\partial S} \right)_{V, n_i} \quad (12.3)$$

and also

$$T = \left(\frac{\partial U}{\partial S} \right)_{V, n_i} \quad (12.4)$$

Gibbs-Helmholtz equations

$$\left(\frac{\partial F/T}{\partial T}\right)_{V,n_i} = -\frac{U}{T^2} \quad (12.5)$$

$$\left(\frac{\partial G/T}{\partial T}\right)_{p,n_i} = -\frac{H}{T^2} \quad (12.6)$$

Specific heat

$$C_v = \left(\frac{\partial U}{\partial T}\right)_{V,n_i} \quad (12.7)$$

$$C_p = \left(\frac{\partial U}{\partial T}\right)_{p,n_i} \quad (12.8)$$

$$C_p - C_v = T \left(\frac{\partial p}{\partial T}\right)_{V,n_i} \left(\frac{\partial V}{\partial T}\right)_{p,n_i} \quad (12.9)$$

Chapter 13

Appendix E: Equations of motion from the Lagrangian or Hamiltonian

13.1 Introduction

To demonstrate the basics of molecular dynamics it is sufficient to start with Newton's equations of motion. The more advanced techniques use the Lagrangian and the Hamiltonian to derive the equations of motion. Here we demonstrate the relation between these two approaches. We outline the principles (based on the famous Feynman lectures on physics [38]). For a more detailed and formal description of classical mechanics, the reader is referred to the book of Goldstein [39].

Consider the following statement: the equation of motion gives the path for which the action, S , is minimum. The action is defined as the integral over the difference in kinetic \mathcal{U}_K and potential \mathcal{U}_P energy:

$$S = \int_{t_b}^{t_e} dt [\mathcal{U}_K - \mathcal{U}_P] . \quad (13.1)$$

For any other path S is bigger. Let us see whether this statement is reasonable for a few simple cases.

The first case is a single particle in a zero potential $\mathcal{U}_P = 0$. Let us write the velocity of the particle as the sum of the average velocity v_{av} and the deviation from it $\eta(t)$:

$$v(t) = v_{av} + \eta(t) \quad (13.2)$$

where the average velocity is defined in such a way that:

$$\int dt v(t) = \int dt v_{av} \quad (13.3)$$

Since we have only kinetic energy, we obtain for equation 13.1:

$$S = \frac{1}{2} m \int dt [v_{av} + \eta(t)]^2 = S_{av} + \frac{1}{2} m \int dt \eta^2(t) \quad (13.4)$$

Since the last term is always greater than zero, the action has its minimum if $\eta(t) = 0$. This implies that if there is no force acting on a particle, the particle is moving with a constant velocity. This is equivalent to Newton's first law.

We now let our particle move in a one-dimensional potential $U(x)$. The action for this case is:

$$S = \int_{t_b}^{t_e} dt \left[\frac{1}{2} m \left(\frac{dx(t)}{dt} \right)^2 - U(x) \right]. \quad (13.5)$$

An arbitrary path, $x(t)$, can be written as the true path, $\bar{x}(t)$, plus a small deviation from the true path $\eta(t)$:

$$x(t) = \bar{x}(t) + \eta(t) \quad (13.6)$$

We assume that the boundary conditions are chosen such that the beginning and the end of the paths are fixed, or, $\eta(t_b) = \eta(t_e) = 0$. Furthermore, we assume that the deviation of the true path is small. The action is minimal if the difference of the action along path $x(t)$, S and the action along the true path, \bar{S} , is minimal. This is a problem that can be solved using calculus of variation. Since $\eta(t)$ is small, we can make an expansion of the action around the action of the true path:

$$\begin{aligned} S &= \int_{t_b}^{t_e} dt \frac{1}{2} m \left(\frac{d\bar{x}(t)}{dt} + \frac{d\eta(t)}{dt} \right)^2 - U[\bar{x}(t) + \eta(t)] \\ &= \int_{t_b}^{t_e} dt \frac{1}{2} m \left[\left(\frac{d\bar{x}(t)}{dt} \right)^2 + 2 \frac{d\bar{x}(t)}{dt} \frac{d\eta(t)}{dt} \right] - \left[U(\bar{x}(t)) + \frac{dU(\bar{x})}{dx} \eta(t) \right] \\ &= \bar{S} + \int_{t_b}^{t_e} dt \left[m \frac{d\bar{x}(t)}{dt} \frac{d\eta(t)}{dt} - \frac{dU(\bar{x})}{dx} \eta(t) \right] \\ &= \bar{S} + m \frac{d\bar{x}(t)}{dt} \eta(t) \Big|_{t_b}^{t_e} - \int_{t_b}^{t_e} dt \left[m \frac{d^2\bar{x}(t)}{dt^2} + \frac{dU(\bar{x})}{dx} \right] \eta(t) \end{aligned} \quad (13.7)$$

The last step of this equation has been obtained via partial integration. Since by definition $\eta(t) = 0$ at the boundaries, the second term on the right hand side is zero. The action has its minimum if the term under the integration is zero for all paths, *i.e.*, for all values of $\eta(t)$:

$$m \frac{d^2\bar{x}(t)}{dt^2} = - \frac{dU(\bar{x})}{dx} \quad (13.8)$$

which is exactly Newton's second law. This shows that Newton's equations of motion can be derived from our statement that a particle follows a path for which the action is the minimum.

13.2 Lagrangian

One may wonder whether we can use this path formulation of the equations of motion for something more useful than an elegant derivation of something we learned in high-school. The answer becomes clear if we realize that this alternative formulation is not limited to Cartesian coordinates but can be written in any sets of coordinates. Suppose that we would like to use some generalized coordinates q instead of the Cartesian x coordinate. For example, if we have a pendulum hanging on the ceiling we could use the angle with the vertical to describe the motion of the pendulum. Since the true path should be independent of the coordinates which we use to describe the path, the action should be the same:

$$S = \int dt \mathcal{L}(x, \dot{x}) = \int dt \mathcal{L}(q, \dot{q}) \quad (13.9)$$

where \mathcal{L} is called the Lagrangian. The Lagrangian is defined as the kinetic energy minus the potential energy*:

$$\mathcal{L} \equiv \mathcal{U}_K(\dot{q}) - \mathcal{U}_P(q) \quad (13.10)$$

We again introduce our ideal path $\bar{q}(t)$ and the deviation $\eta(t)$ from it:

$$q(t) = \bar{q}(t) + \eta(t) \quad (13.11)$$

$$\dot{q}(t) = \dot{\bar{q}}(t) + \dot{\eta}(t) \quad (13.12)$$

We can write for the Lagrangian:

$$\mathcal{L}(q, \dot{q}) = \mathcal{L}(\bar{q}, \dot{\bar{q}}) + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial \dot{q}} \dot{\eta}(t) + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial q} \eta(t) \quad (13.13)$$

Like in the previous section, we use calculus of variation to derive an expression for the true path. We substitute this Lagrangian in the expression for the action (equation 13.9). Next we write the actual path as the sum of the true path plus a correction and separate the action of the true path. Then we use partial integration, and use the fact that at the boundaries of the integration the deviation from the true path is zero. Finally, we find that the action has its minimum if:

$$\int dt \left[-\frac{d}{dt} \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial \dot{q}} + \frac{\partial \mathcal{L}(\bar{q}, \dot{\bar{q}})}{\partial q} \right] \eta(t) = 0 \quad (13.14)$$

To derive the equation of motion we need to introduce a momentum associate to the generalized coordinate q :

$$p_q \equiv \frac{\partial \mathcal{L}(q, \dot{q})}{\partial \dot{q}} \quad (13.15)$$

Substitution of this expression into equation 13.14 gives:

$$\dot{p}_q = \frac{\partial \mathcal{L}(q, \dot{q})}{\partial q} \quad (13.16)$$

which is the equation of motion in terms of the generalized coordinates (q, p_q) . If the above formulation is valid for any coordinate system, it should certainly hold for Cartesian coordinates. In these coordinates the Lagrangian reads:

$$\mathcal{L}(x, \dot{x}) = \frac{1}{2} m \dot{x}^2 - \mathcal{U}(x) \quad (13.17)$$

The momentum associated to x is:

$$p_x = \frac{\partial \mathcal{L}(x, \dot{x})}{\partial \dot{x}} = m \dot{x} \quad (13.18)$$

and the equation of motion is:

$$m \ddot{x} = - \frac{\partial \mathcal{U}(x)}{\partial x} \quad (13.19)$$

which is indeed the result we would obtain from Newton's equation of motion.

*The true definition is more restrictive; see ref. [39] for more details.

13.3 Example

Consider the simple pendulum of length l with mass m hanging on the ceiling. The gravitational force is acting on the pendulum and the potential energy is a simple function of the angle with the vertical θ :

$$U(\theta) = mgl [1 - \cos(\theta)] \quad (13.20)$$

We would like to write down the equations of motion in terms of the generalized coordinate θ . It is an exercise for the reader to do the same using the Cartesian coordinates x and y . The Lagrangian is:

$$\begin{aligned} \mathcal{L} &= \mathcal{U}_K - \mathcal{U}_P = \frac{1}{2}m \left(\dot{x}^2(t) + \dot{y}^2(t) \right) - U(\theta) \\ &= \frac{ml^2}{2} \dot{\theta}^2 - U(\theta) \end{aligned} \quad (13.21)$$

The generalized impulse is defined as:

$$p_\theta = \frac{\partial \mathcal{L}}{\partial \dot{\theta}} = ml^2 \dot{\theta} \quad (13.22)$$

and the equation of motion follows from equation 13.16

$$\dot{p}_\theta = -\frac{\partial U(\theta)}{\partial \theta} \quad (13.23)$$

or

$$\dot{\theta} = -\frac{1}{ml^2} \frac{\partial U(\theta)}{\partial \theta} \quad (13.24)$$

13.4 Hamiltonian

Using the Lagrangian, we have derived the equation of motions in terms of q and \dot{q} . In some applications one would like to write the equations of motion in terms of q and the conjugate momentum p_q . To do this we can perform a Legendre transformation:[†]

$$\mathcal{H}(q, p_q) \equiv p_q \dot{q} - \mathcal{L}(q, \dot{q}, t) \quad (13.27)$$

[†]In thermodynamics the Legendre transformation is used to derive the auxiliary functions. For example, the energy U is a function of the entropy S and volume V : $U = U(S, V)$. In some practical application it is more convenient to work with the temperature T instead of the volume. Since the temperature is the conjugate variable to the entropy, we can make a Legendre transformation to remove the S dependence:

$$A = U - TS \quad (13.25)$$

giving

$$dA = dU - d(TS) = -SdT - pdV \quad (13.26)$$

This equation is defining the *Hamiltonian* of the system. For the differential we can write

$$\begin{aligned}
 d\mathcal{H}(q, p_q) &= d(p_q \dot{q}) - d\mathcal{L}(q, \dot{q}) \\
 &= p_q d\dot{q} + \dot{q} dp_q - \left[\frac{\partial \mathcal{L}}{\partial q} dq + \frac{\partial \mathcal{L}}{\partial \dot{q}} d\dot{q} + \frac{\partial \mathcal{L}}{\partial t} dt \right] \\
 &= p_q d\dot{q} + \dot{q} dp_q - \dot{p}_q dq - p_q d\dot{q} - \frac{\partial \mathcal{L}}{\partial t} dt \\
 &= \dot{q} dp_q - \dot{p}_q dq - \frac{\partial \mathcal{L}}{\partial t} dt \\
 &= \frac{\partial \mathcal{H}}{\partial p_q} dp_q + \frac{\partial \mathcal{H}}{\partial q} dq + \frac{\partial \mathcal{H}}{\partial t} dt
 \end{aligned} \tag{13.28}$$

in which we have used the definitions of p_q and \dot{p}_q , equations 13.15 and 13.16, respectively. From this equation we read:

$$\dot{q} = \frac{\partial \mathcal{H}}{\partial p_q} \tag{13.29}$$

$$\dot{p}_q = -\frac{\partial \mathcal{H}}{\partial q} \tag{13.30}$$

Which are the desired equations of motion in terms of q, p_q . For most systems the Lagrangian does not explicitly depend on time. If this is the case the Hamiltonian is equal to the total energy, which gives a conservation law.

Also the Hamiltonian formulation is independent of the coordinate system. Of course, it should be valid for Cartesian coordinates, for which the Hamiltonian reads:

$$\begin{aligned}
 \mathcal{H}(x, p_x) &= \dot{x} p_x - \mathcal{L}(x, \dot{x}) \\
 &= m\dot{x}^2 - \frac{1}{2}m\dot{x}^2 + U(x) \\
 &= \frac{1}{2m}p_x^2 + U(x)
 \end{aligned} \tag{13.31}$$

The equations of motion are:

$$\dot{x} = \frac{\partial \mathcal{H}}{\partial p_x} = \frac{p_x}{m} \tag{13.32}$$

$$\dot{p}_x = -\frac{\partial \mathcal{H}}{\partial x} = -\frac{\partial U(x)}{\partial x} \tag{13.33}$$

Which are the familiar Newton's equation of motion again. Comparison with the Lagrangian formalism shows that with the Hamiltonian we obtain two first order differential equations, while the Lagrangian gives one second order equation. Of course, in both formalisms the physics is the same.

Chapter 14

Appendix F: Questions about the book “Understanding ..” [2]

14.1 Introduction to FORTRAN77

Question 1 (Standard deviation) Write a short FORTRAN77 program that computes the average and standard deviation of a sequence of N numbers. The average of a variable X is defined as

$$\langle X \rangle = \frac{\sum_{i=1}^{i=N} X_i}{N} \quad (14.1)$$

and the standard deviation as

$$\sigma = \sqrt{\frac{\sum_{i=1}^{i=N} (X_i - \langle X \rangle)^2}{N}} \quad (14.2)$$

Why is this expression not useful to use for a very large N ? Rewrite the expression for the standard deviation in such a way that X_i does not have to be stored.

Question 2 (Matrix Multiplication) Write a FORTRAN77 program that multiplies a $[a \times b]$ matrix with a $[b \times c]$ matrix. The result is a $[a \times c]$ matrix. By the way, what is the matrix [40] ?

14.2 Chapter 2

Question 3 (Number of configurations)

1. Consider a system A consisting of subsystems A_1 and A_2 , for which $\Omega_1 = 10^{20}$ and $\Omega_2 = 10^{22}$. What is the number of configurations available to the combined system ? Also, compute the entropies S , S_1 , and S_2 .
2. By what factor does the number of available configurations increase when 10 m^3 of air at 1.0 atm and 300K is allowed to expand by 0.001 per cent at constant temperature ?
3. By what factor does the number of available configurations increase when 150 kJ is added to a system containing 2.0 mol of particles at constant volume and $T = 300\text{K}$?
4. A sample consisting of five molecules has a total energy 5ϵ . Each molecule is able to occupy states of energy ϵ_j , with $j = 0, 1, 2, \dots, \infty$. Draw up a table with columns by the energy of the states and write beneath them all configurations that are consistent with the total energy. Identify the most probable configurations.

Question 4 (Thermodynamic variables in the canonical ensemble) *If one has an expression for the Helmholtz free energy (F) as a function of N , V , T*

$$F = \frac{-\ln(Q(N, V, T))}{\beta} \quad (14.3)$$

one can derive all thermodynamic properties. Show this by deriving equations for U , p , and S . You might want to take a look at Appendix D to refresh your memory.

Question 5 (Ideal Gas (Part 1)) *The canonical partition function of an ideal mono-atomic gas is equal to*

$$Q(N, V, T) = \frac{1}{h^{3N} N!} \int d\Gamma \exp[-\beta H] = \frac{V^N}{\lambda^{3N} N!} \quad (14.4)$$

in which $\lambda = h/\sqrt{2\pi m/\beta}$ and $d\Gamma = dq_1 \cdots dq_N dp_1 \cdots dp_N$. Derive expressions for the following thermodynamic properties:

- $F(N, V, T)$ (hint: $\ln(N!) \approx N \ln(N) - N$)
- $p(N, V, T)$ (which leads to the ideal gas law !!!)
- $\mu(N, V, T)$ (which leads to $\mu = \mu_0 + RT \ln \rho$)
- $U(N, V, T)$
- $S(N, V, T)$
- C_v (heat capacity at constant volume)
- C_p (heat capacity at constant pressure)

Question 6 (Ising model) *Consider a system of N spins arranged on a lattice. In the presence of a magnetic field, H , the energy of the system is*

$$U = - \sum_{i=1}^N H \mu s_i - J \sum_{i>j} s_i s_j \quad (14.5)$$

in which J is called the coupling constant ($J > 0$) and $s_i = \pm 1$. The second summation is a summation over all pairs ($D \times N$ for an infinitely large system, D is the dimensionality of the system). This system is called the Ising model.

Questions:

1. Show that for positive J , and $H = 0$, the lowest energy of the Ising model is equal to

$$U_0 = -DNJ \quad (14.6)$$

in which D is the dimensionality of the system.

2. Show that the free energy per spin of a 1D Ising model with zero field is equal to

$$\frac{F(\beta, N)}{N} = -\frac{\ln(2 \cosh(\beta J))}{\beta} \quad (14.7)$$

when $N \rightarrow \infty$. The function $\cosh(x)$ is defined as:

$$\cosh(x) = \frac{\exp[-x] + \exp[x]}{2} \quad (14.8)$$

3. Derive equations for the energy and heat capacity of this system.

Question 7 (The photon gas) A photon gas is an electromagnetic field in thermal equilibrium with its container. From the quantum theory of the electromagnetic field, it is found that the total energy of the system (\mathcal{U}) can be written as the sum of energies of harmonic oscillators:

$$\mathcal{U} = \sum_{j=1}^N n_j \omega_j \hbar = \sum_{j=1}^N n_j \epsilon_j \quad (14.9)$$

in which ϵ_j is the characteristic energy of oscillator j , $n_j = 0, 1, 2, \dots, \infty$ is the so called occupancy number of oscillator j and N is the number of oscillators.

Questions:

1. Show that the canonical partition function of the system can be written as

$$Q = \prod_{j=1}^N \frac{1}{1 - \exp[-\beta \epsilon_j]} \quad (14.10)$$

Hint: you will have to use the following identity for $|x| < 1$

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x} \quad (14.11)$$

For the product of partition functions of two independent systems A and B we can write

$$Q_A \times Q_B = Q_{AB} \quad (14.12)$$

when $A \cap B = \emptyset$ and $A \cup B = AB$.

2. Show that the average occupancy number of state j , $\langle n_j \rangle$, is equal to

$$\langle n_j \rangle = \frac{\partial \ln Q}{\partial (-\beta \epsilon_j)} = \frac{1}{\exp[\beta \epsilon_j] - 1} \quad (14.13)$$

3. What happens with $\langle n_j \rangle$ when $T \rightarrow \infty$ and $T \rightarrow 0$?

Question 8 (Ideal Gas (Part 2)) An ideal gas is placed into a gravitational field $\phi(z) = mgz$. The temperature in the system is uniform and the system infinitely large. The magnitude of the field, g is constant. We assume that the system is locally in equilibrium, so we are allowed to use a local partition function.

Questions:

1. Show that the grand-canonical partition function at height z is equal to

$$Q(\mu, V, T, z) = \sum_{N=0}^{\infty} \frac{\exp[\beta\mu N]}{h^{3N} N!} \int d\Gamma \exp[-\beta(H_0 + mgz)] \quad (14.14)$$

in which H_0 is the Hamiltonian of the system at $z = 0$.

2. Explain that a change in z is equivalent to a change in chemical potential, μ . Use this to show that the pressure of the gas at height z is equal to

$$p(z) = p(z=0) \times \exp[-\beta mgz] \quad (14.15)$$

(Hint: you will need the formula for the chemical potential of an ideal gas).

3. How does this formula change when the gravitational force g is not constant but a function of z ? Use the symbol g_0 for the gravitation at the surface of the Earth and R for the radius of the Earth.

Question 9 (Chemical Reaction) Consider the chemical reaction



in the (ideal) gas phase.

1. The equilibrium ratio of the A and B populations is given by

$$\frac{\langle n_A \rangle}{\langle n_B \rangle} = \frac{q_A}{q_B} = \frac{g_A}{g_B} \exp[-\beta \Delta \epsilon] \quad (14.17)$$

in which q_i is the partition function and g_i the degeneracy of component i and $\Delta \epsilon$ the energy difference between A and B. Show how the same result follows from the condition of chemical equilibrium, $\mu_A = \mu_B$.

2. The partition function Q of this system can be written as

$$Q = \frac{1}{N!} (q_A + q_B)^N \quad (14.18)$$

in which $N = n_A + n_B$. Show that the condition of chemical equilibrium is identical to finding the partitioning of A and B that minimizes the Helmholtz free energy

$$\frac{\partial F}{\partial n_A} = \frac{\partial F}{\partial n_B} = 0 \quad (14.19)$$

Hint: Use $\ln n! \approx n \ln n - n$ and remember that $F = -k_B T \ln Q$.

3. Show that

$$\langle [n_A - \langle n_A \rangle]^2 \rangle = \frac{\langle n_A \rangle \langle n_A \rangle}{N} \quad (14.20)$$

Hint: Use

$$\langle n_A \rangle = q_A \left(\frac{\partial \ln Q}{\partial q_A} \right)_{q_B, N} = \frac{N q_A}{q_A + q_B} \quad (14.21)$$

14.3 Chapter 3

Question 10 (Reduced units) A typical set of Lennard-Jones parameters for Argon and Krypton is $\sigma_{Ar} = 3.41\text{\AA}$, $\epsilon_{Ar}/k_B = 119.8\text{K}$ and $\sigma_{Kr} = 3.38\text{\AA}$, $\epsilon_{Kr}/k_B = 164.0\text{K}$ [3].

1. At the reduced temperature $T^* = 2.0$, what is the temperature of Argon and Krypton ?
2. A typical time step for MD is $\Delta t^* = 0.001$. What is this in SI units for Argon and Krypton ?
3. If we simulate Argon at $T = 278\text{K}$ and density $\rho = 2000\text{ kg/m}^3$ with a Lennard-Jones potential, for which conditions of Krypton can we use the same data ? If we assume ideal gas behavior, compute the pressure in reduced and normal units.
4. What are the main reasons to use reduced units anyway ?

Question 11 (Heat capacity)

1. The heat capacity can be calculated from differentiating the total energy of a system with respect to the temperature. Is it a good idea to calculate the heat capacity from energies of different simulations at different temperatures ? Explain.
2. The heat capacity can also be calculated from fluctuations in the total energy in the canonical ensemble:

$$C_v = \frac{\langle U^2 \rangle - \langle U \rangle^2}{k_B T^2} \quad (14.22)$$

Derive this equation.

3. In a MC-NVT simulation, one does not calculate fluctuations in the total energy but in the potential energy. Is it then still possible to calculate the heat capacity ? Explain.

Question 12 (A new potential) The authors were getting bored with the Lennard-Jones (12 – 6) potential and decided to introduce the 10 – 5 potential

$$U(r) = 5\epsilon \left[\left(\frac{\sigma}{r} \right)^{10} - \left(\frac{\sigma}{r} \right)^5 \right] \quad (14.23)$$

But the authors are also lazy and it is therefore up to you to derive the tail corrections for the energy, pressure, and chemical potential (see Chapter 7 of [2]). If we use this potential in an MD simulation in the truncated and shifted form we still have a discontinuity in the force. Why ? How should the potential be modified to remove this discontinuity ? If you compare this potential with the Lennard-Jones potential, will there be any difference in efficiency of the simulation ? (Hint: there are two effects !)

14.4 Chapter 4

Question 13 (Integrating the equations of motion)

1. If you do an MD simulation of the Lennard-Jones potential with a time step that is too big you will find an energy drift. This drift is towards a higher energy. Why ?
2. Why don't we use Runge-Kutta methods to integrate the equations of motion of particles in MD ?
3. Which of the following quantities are conserved in the MD simulation of Case Study 4: potential energy, total impulse, center of mass of the system and angular momentum ?

4. Show that the Verlet and velocity Verlet algorithms have identical trajectories.
5. Derive the Leap-Frog algorithm by using Taylor expansions for $v(t + \frac{\Delta t}{2})$, $v(t - \frac{\Delta t}{2})$, $x(t + \Delta t)$ and $x(t)$.

Question 14 (Correlation functions)

1. To which quantity is the velocity autocorrelation function ($vacf$) related for $t = 0$?
2. Calculate the limit of the $vacf$ for $t \rightarrow \infty$.
3. What is the physical significance if $vacf < 0$?
4. When you calculate the mean square displacement for particles in a system in which periodic boundary conditions are used and in which particles are placed back in the box, you should be very careful in calculating the displacement. Why ?
5. What is more difficult to calculate accurately: the self-diffusion coefficient or the viscosity ? Explain.

14.5 Chapter 5

Question 15 (Trial moves)

1. Explain why a large fraction of particle swap trial moves in the grand-canonical ensemble is bad for the equilibration of the system.
2. Which trial move in Chapter 5 of ref. [2] will be computationally most expensive ? Why ?
3. In a simulation of a molecule that consists of more than one interaction site, a trial move that rotates the molecule around its center of mass is usually included. Why ? What is the acceptance/rejection rule for this trial move ?
4. When a particle is added in the grand-canonical ensemble, there might be an energy change due to a change in the tail-corrections. Why ? Derive an expression for this energy change when a Lennard-Jones potential is used.

Question 16 (Multicomponent simulation) We consider a grand-canonical scheme for a mixture of two components. Assume the temperature is T and the chemical potential of the components μ_1 and μ_2 .

1. To add or remove particles the following scheme is used:

- Select at random to add or remove a particle.
- Select at random a component.
- Add or remove a particle of this component.

Derive the acceptance rules for these trial moves.

2. An alternative scheme would be:

- Select at random to add or remove a particle.
- Select at random a particle (either in the system or in the reservoir), independent of its identity.

Is this scheme obeying detailed balance if the previous acceptance rules are used ? If not can this be corrected ? Hint: you might want to see ref. [41].

14.6 Chapter 6

Question 17 (Andersen thermostat)

1. Why do static properties calculated by NVT-MD using the Anderson thermostat do not depend on ν ?
2. Suppose we simulate a single particle using the Andersen thermostat. How do we have to choose the frequency to obtain exactly the same temperature fluctuations as in the canonical ensemble ?
3. Why does the self-diffusivity decrease with increasing ν ?

Question 18 (Nosé-Hoover thermostat)

1. Explain when we have to use $g = 3N + 1$ and $g = 3N$ in the Nosé-Hoover thermostat.
2. Often, the friction term in equation 6.1.25 of [2] is calculated iteratively. Are there some disadvantages when you do this ?
3. Instead of a single Nosé-Hoover thermostat, one can also use a chain of thermostats. Does this increase the total CPU time of the simulation for a big system ? Explain.
4. Another widely used thermostat is the “temperature coupling” of Berendsen [20]. It is important to note that this method does not produce a canonical ensemble and therefore we should never use it. In this algorithm, the temperature of the system is controlled by scaling the velocities every time step with a factor λ

$$\lambda = \left[1 + \frac{\Delta t}{\tau_T} \left(\frac{T_0}{T} - 1 \right) \right]^{\frac{1}{2}} \quad (14.24)$$

in which T_0 is the desired temperature, T is the actual temperature, Δt is the time step of the integration algorithm and τ_T is a constant.

- Show that this scaling is equivalent with a temperature coupling of the system with a heat bath at $T = T_0$

$$J = \alpha (T_0 - T) \quad (14.25)$$

in which J is the heat flux and α is the heat transfer coefficient.

- How do α and τ_T relate ?

14.7 Chapter 7

Question 19 (Free energy)

1. Why does equation 7.2.11 of ref. [2] (although in principle correct) does not work for hard spheres ? Is there something wrong with the theory ?
2. Derive an expression for the error in estimate of the chemical potential obtained by Widom’s test particle method for a system of hard spheres. The probability of generating a position with at least one overlap is equal to p .

3. An alternative method to calculate the free energy difference between state A and state B is the use of the difference in Hamiltonian:

$$F_A - F_B = \frac{-\ln [\langle \exp [-\beta (H_A - H_B)] \rangle_{N,V,T,B}]}{\beta} \quad (14.26)$$

Derive this equation. What are the limitations of this method ? Show that the Widom's test particle method is just a special case of this equation.

Question 20 (Ghost volume) The virial equation is not very convenient to compute the pressure of a hard sphere fluid. Why ? It is much more convenient to perform a constant pressure simulation and compute the density. An alternative way to compute the pressure of a hard sphere fluid directly is to use a ghost-volume change. In this method, a virtual displacement of the volume is performed and the probability that such a (virtual) move it is accepted has to be computed. Derive that this is indeed a correct way of calculating the pressure. (Hint: consider the analogy of Widom's test particle method).

14.8 Chapter 8

Question 21 (Gibbs ensemble)

1. When one of the boxes in the Gibbs ensemble is infinitely large and the molecules in this box do not have intermolecular interactions, the acceptance/rejection rule for particle swap becomes identical to the acceptance/rejection rule for particle swap in the grand-canonical ensemble. Derive this result.
2. From this, it is also possible to derive an equation for the particle exchange in the grand-canonical ensemble when the gas-phase is not ideal. Proof that in this case, we have to replace the pressure p by the fugacity f . How is the fugacity defined anyway ?
3. Why does the Gibbs ensemble not work for solid-liquid equilibria ?

Question 22 (Scaling of the potential) When an attempt is made to change the volume in the Gibbs ensemble, for some systems the energy of the new configuration can be calculated efficiently when the scaling properties of the potential are used. Consider a system of Lennard-Jones particles. The total energy \mathcal{U} of the system is equal to

$$\mathcal{U} = \sum_{i < j} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \quad (14.27)$$

Suppose that the box-size of the system is changed from L to L' and $s = L'/L$.

1. Why is this scheme so efficient ?
2. What is the expression for the total virial of this system ?
3. Why does this method only work when the cut-off radius is scaled as well ?
4. Derive expressions for the new energy \mathcal{U}' and new virial V' as a function of s , the old energy (\mathcal{U}) and virial (V).

14.9 Chapter 11

Question 23 (Barrier crossing) Consider the barrier crossing problem of exercise 7.1.

1. Why does $k_{A \rightarrow B}(t)$ always decreases as a function of time when the dividing surface is not located at the top of the barrier ?
2. How will $k_{A \rightarrow B}(t)$ depend on ν when NVT-MD with an Andersen thermostat is used instead of NVE-MD to compute trajectories ? In this question, you can assume that the dividing surface is on top of the energy barrier.

14.10 Chapter 13

Question 24 (Biased CBMC) In CBMC, trial positions are selected with a probability that is proportional to the Boltzmann factor of each trial segment. However, in principle one can use another probability function [42] to select a trial segment. Suppose that the probability of selecting a trial segment i is proportional to

$$p_i \propto \exp[-\beta^* u_i] \quad (14.28)$$

in which $\beta^* \neq \beta$.

1. Derive the correct acceptance/rejection rule for this situation.
2. Derive an expression for the excess chemical potential when this modified CBMC method is used to generate configurations of test particles.
3. What will happen when $\beta^* \rightarrow \infty$ and $\beta^* \rightarrow 0$?

Chapter 15

Appendix G: Possible research projects

15.1 Adsorption in porous media

In this project we will investigate the adsorption behavior in porous media. As a model we use a slit-like pore. The interactions with the pore are given by

$$U(z) = \begin{cases} 0 & 0 < z < L \\ \infty & \text{otherwise} \end{cases} \quad (15.1)$$

where L is the width of the slit. We will investigate the adsorption of methane which we model with a Lennard-Jones potential. The starting point is Case Study 9 of ref. [2] which includes a program to simulate the Lennard-Jones fluid in the grand-canonical ensemble. The project is to develop a program to simulate an adsorption isotherm of methane in the slit-like pore. Before starting to program you may want to think about the following points:

1. What is the geometry of the system and how should one apply the periodic boundary conditions?
2. What is the type of Lennard-Jones potential (truncated, truncated and shifted, with or without tail corrections)? And what are the parameters to model methane?
3. How is an adsorption isotherm defined thermodynamically?

With your program, try to answer the following questions.

1. Compute the density profile (density as function of the distance z between the plates) for $L = 1, 2, 5$, and 10 for $\rho = 0.7$ and $T = 2.0$. The total number of particles should be of the order of 100 to 500 for the widest slit.
2. Compute the excess chemical potential and chemical potential as a function of the distance between the plates for $L = 1, 2, 5$, and 10 for $\rho = 0.6$ and $T = 2.0$. Try to explain the differences.
3. Compute the adsorption isotherms for $L = 2$ and $L = 5$ for $T = 2.0$ and $T = 0.8$. Try to explain the results.

15.2 Transport properties in liquids

Molecular dynamics simulations can be used to compute the transport properties of liquids. Examples of these transport properties are the (self) diffusion coefficient and the viscosity. In

Case Study 5 the results for the diffusion coefficient are shown. In this project we extend these results to mixtures of Lennard-Jones (LJ) molecules. Before starting to program you may want to think about the following points:

1. The generalization of the diffusion coefficient from a pure component to a mixture is not trivial. Try to find in the literature how one should define a diffusion coefficient of a mixture and how can one compute this in a simulation. See, for example, refs. [43,44].
2. What is the type of LJ potential (truncated, truncated and shifted, with or without tail corrections)? And what are the parameters to model Argon and Krypton?

With your program, try to answer the following questions:

1. Compute the pressure, viscosity, and diffusion coefficient of the LJ fluid at $T = 1.0, 1.5$, and 2.0 for $\rho = 0.7$.
2. Compute the diffusion coefficients D_{11} , D_{12} , and D_{22} for a mixture of 50% -50% LJ particles in which the components 1 and 2 have the same interactions ($\sigma_{12} = \sigma_{11} = \sigma_{22}$ and $\epsilon_{12} = \epsilon_{11} = \epsilon_{22}$) but carry a different color. This means that the particles are labelled. Experimentally this could be done by radioactive labelling.
3. Compute the pressure, viscosity, and diffusion coefficient of the LJ fluid at $T = 1.0, 1.5$, and 2.0 for $\rho = 0.7$. Compute the diffusion coefficients D_{11} , D_{12} , and D_{22} for a mixture of 50-50% LJ particles, but now for a system of 50% Argon and 50% Krypton (use the parameters of Argon to compute the reduced temperatures). Does the Einstein equation for the diffusivity hold for this system?

15.3 Parallel Molecular Dynamics

In this project, we will test the scaling of a parallel Molecular Dynamics program on a Beowulf PC cluster [45,46]. It is interesting to compare the performance of such a cluster with off-the-shelf components with an expensive supercomputer. For simplicity, we will do MD in only two dimensions. We will use a soft short-ranged repulsive potential of equation 8.2. Successively, the following things have to be done:

- Write the sequential MD code using a linked-cell list (see, for example, refs. [2,3]). How does your code scale with the number of particles? Make sure that your code calculates the average energy drift, the pressure and the radial distribution function.
- Make yourself familiar with MPI [47]. This will take you a few days. We will strongly advise you to use MPI instead of other parallel libraries like, for example, PVM [48]. Popular MPI implementations are LAM [18] and MPI-CH [49].
- The program will be parallelized using domain decomposition, which means that every processor controls a fixed part of the simulation box. For simplicity, we will use a domain decomposition in slabs. In every domain, the forces are calculated using a linked-cell method. In a simulation, the following steps are executed:
 - Particles are assigned to a processor.
 - Every processor has to know the positions and velocities of the particles in its domain, and also the positions and velocities of the particles in a cell of length r_{cut} next to its domain (This is the so-called halo region).

- Every time step, the halo region has to be swapped between neighboring slabs.
- To calculate quantities like pressure and temperature a global summation over all processors is necessary.
- Make a detailed work-plan about how you are going to program this. Make a list of all key variables and arrays. This is necessary because writing this program won't be easy.
- Discuss this work-plan with your supervisor before you start programming.
- Make a detailed investigation of the scaling of this code with the number of processors. How does the scaling change with the number of processors ?
- Compare your results with a replicated data algorithm.

15.4 Phase diagram of ethane

Ethane can be modeled as two Lennard-Jones particles that are connected with a spring. In literature, Gibbs ensemble simulations have been performed of such a model. In this exercise you will have to modify Case Study 16 for using this model. Because an ethane molecule is quite small, we can still use conventional Monte Carlo techniques to insert or remove molecules. Try to find LJ parameters that correctly describe the vapor-liquid part of the phase diagram of ethane. You might want to take a look at refs. [50,51] for some details about a model for ethane.

Questions:

1. What kind of potential are you going to use (truncated, truncated and shifted, with or without tail corrections) ?
2. Is it a good idea to include a new type of trial move (rotation of the molecule around its center of mass) ? Explain.
3. Do you take a fixed C – C bond-length or do you use a harmonic potential ? Is there any difference in the algorithm between a fixed or flexible C – C bond ? Explain !
4. What is the bond-length of ethane in dimensionless units ?
5. How can you make an accurate estimate of the critical temperature and density ?
6. How does the fraction of accepted swap moves change with the temperature ?

15.5 Diffusion in a porous media

The behavior of liquid in confined geometries is different from the behavior in the bulk liquid. In the project we investigate the diffusion coefficient in a cylindrical pore. The starting point is Case Study 4 of [2] in which the dynamic properties of a Lennard-Jones fluid are simulated. This Case Study can be used as a starting point for our study. The interactions with the walls are described with a repulsive potential:

$$U(r) = \begin{cases} \epsilon \left(\frac{\sigma}{L-r} \right)^{10} & 0 \leq r \leq L \\ \infty & r > L \end{cases} \quad (15.2)$$

where L is a radius characterizing the size of the pore and the center of the cylinder is located at $r = 0$ and $\epsilon > 0$, $\sigma > 0$. Some questions that one should answer before one starts programming:

1. Is the potential for the interactions with the walls appropriate for a Molecular Dynamics simulation ?
2. What is the volume of the pore as a function of the parameter L ?
3. What is the dimension of our problem ? Do we have diffusion in 1, 2, or 3 dimensions ?

In the first part of the project we study the diffusion in a smooth pore as defined by the above potential as a function of the pore diameter.

1. Compute the diffusion coefficient of a bulk Lennard-Jones liquid for $\rho = 0.6$ and $T = 2.0$ and $T = 1.5$. Since the program uses an NVE ensemble it is not possible to simulate at exactly the requested temperature. However, one can ensure to be close to this temperature by an appropriate equilibration of the system (this is also the case for the following 2 questions) during the first part of the MD simulation.
2. Compute the density as a function of the distance from the center of the pore for $\rho = 0.6$ and $T = 2.0$ and $T = 1.5$ and $L = 5$ and $L = 2$. Interpret the results.
3. Compute the diffusion coefficient for $\rho = 0.6$ and $T = 2.0$ and $T = 1.5$ and $L = 5$, $L = 2$, and $L = 1$. Interpret the results. The interpretation is not trivial.
4. The above calculations have been performed using the NVE ensemble. This implies that there is no coupling with the atoms of the walls. In a real system the walls are not smooth and can exchange heat with the adsorbed molecules. A possible way of modeling this is to assume that we have a Andersen thermostat in the boundary layer with the wall. Investigate how the results depend on the thickness of the boundary layer and the constant ν of the Andersen algorithm.

The next step is to model the corrugation caused by the atoms. This corrugation could be a term:

$$U(z, r) = A \sin^2(\pi z / \sigma_w) \exp \left[- \left(\frac{r - L}{L_0} \right)^2 \right] \quad (15.3)$$

where z is the σ_w is a term characterizing the size of the atoms of the wall and A the strength of the interaction. The exponential is added to ensure that the potential is localized closed to the walls of cylinder. Investigate the diffusion coefficient as a function of the parameters σ_w and A both in the NVE and in the Anderson thermostat case.

15.6 Multiple time step integrators

The time step in a Molecular Dynamics simulation strongly depends on the steepness of the potential energy surface. However, most potentials like the Lennard-Jones potential are steep at short distances. As short-range interactions can be computed very fast, it would be interesting to use a multiple time step integration algorithm, in which short-range (computationally cheap) interactions are computed every time step and in which long-range (computationally expensive) interactions are evaluated every n time steps ($n > 1$). Recently, there has been a considerable effort to construct time-reversible multiple time step algorithms [15, 52].

1. Why is it important to use time-reversible integration schemes in MD ?
2. Modify Case Study 4 in such a way that pairwise interactions are calculated using a Verlet neighbor list [2, 3]. For every particle, a list is made of neighboring particles within a distance of $r_{\text{cut}} + \Delta$. All lists only have to be updated only when the displacement of a single particle is larger than $\Delta/2$. Hint: The algorithm on micro-fiche F.19 of the book of Allen and Tildesley [3,53] is a good starting point.
3. Investigate how the CPU time per time step depends on the size of Δ for various system sizes. Compare your results with table 5.1 from ref. [3].
4. Modify the code in such a way that the NVE multiple time step algorithm of ref. [15] is used to integrate the equations of motion. You will have to use separate neighbor-lists for the short-range and the long-range part of the potential.
5. Why does one have to use a switching function in this algorithm ? Why is it a good idea to use a linear interpolation scheme to compute the switching function from ref. [15] ?
6. Make a detailed comparison between this algorithm and the standard Leap-frog integrator (with the use of a neighbor-list) at the same energy drift.

15.7 Thermodynamic integration

The difference in free energy between state A and state B can be calculated by thermodynamic integration [2]

$$F_A - F_B = \int_{\lambda=0}^{\lambda=1} d\lambda \left\langle \frac{\partial U(\lambda)}{\partial \lambda} \right\rangle_{\lambda} \quad (15.4)$$

in which $\lambda = 1$ in state A and $\lambda = 0$ in state B. In order to calculate the excess chemical potential of a Lennard-Jones system, we might use the following modified potential [54]

$$U(r, \lambda) = 4\epsilon \left(\lambda^5 \left(\frac{\sigma}{r} \right)^{12} - \lambda^3 \left(\frac{\sigma}{r} \right)^6 \right) \quad (15.5)$$

Recall that the excess chemical potential is the difference in chemical potential between a real gas ($\lambda = 1$) and an ideal gas ($\lambda = 0$).

Questions:

1. Make a plot of the modified LJ potential for various values of λ .
2. Show that

$$\left(\frac{\partial^2 F}{\partial \lambda^2} \right) < 0 \quad (15.6)$$

when

$$U = U_0 \lambda + U_1 (1 - \lambda) \quad (15.7)$$

3. Derive equation 15.4.

4. Modify the code of Case Study 1 for this modified potential.
5. Perform the thermodynamic integration and compare your results with the conventional particle insertion method.
6. Calculate the chemical potential as a function of the density by scaling σ .

15.8 Hints for programming

The official rules for writing a program are that it should involve three persons. One for the design, one for the implementation, and one for the testing of a program. In our case you are doing all steps. The reason why there is a need for three persons is that if you do not realize a certain aspect in the design phase, it is very likely that you do not think about this either while implementing or testing a program. The following tips may be useful to develop a program:

- First try to understand every line of the starting code. If you do not understand the starting point it is impossible to make modifications.
- Try to develop a modification plan consisting of the following steps:
 1. Make a copy of the program in a new directory.
 2. Describe in words what and why a certain subroutine needs to be modified.
 3. Start the programming by writing comment lines what the modifications are and make the modifications according to your written instructions. If you find during the implementations that the original ideas were not correct describe this in your notes. This is to ensure that you think before you do. This may sound obvious and it is, but as you may find out it is a very difficult rule to stick to.
 4. Do not start modifying the entire program. Try to do it in steps. Try to test the modifications as soon as possible. For example, if the program is a Gibbs ensemble simulation you probably have to modify, the particle displacement routine, the volume change, and the particle exchange. However, there is no need to do this in all subroutines at the same time. First modify the particle displacement and make all the test you can do (for example, energy conservation) without use the volume displacement and particle exchange subroutines, before you continue to the next subroutine.
 5. The first time you run the modified code, you might want to use array-bound checking (usually, this option is not used by default). This will slow down the code, but it is very useful to detect errors. It is also very useful to use code checkers like for example `ftnchek` [33].
 6. Try to find limiting cases for which your program gives known results. For example, if you have written a program for a mixture the results should be identical to the pure component results if all interaction parameters are taken to be identical. Or does the program give in the ideal gas limit the same results as the ideal gas results that can be obtained from theory. This is, of course, not always possible, but if it is possible it is a good test case.
 7. For a Molecular Dynamics program, it is a good idea to check the conservation of the total energy. In most systems, the total impulse should also be conserved (usually zero).

8. In a Monte Carlo program, one has to calculate energy differences for each trial move. It is a good idea to write a subroutine that rigorously calculates the total energy of the system. In this way, the sum of the initial energy and all energy differences of all accepted trial moves should equal the total energy, any difference should only be due to the limited accuracy of the computer.
 9. Especially for lattice simulations, the result of a MC simulation might be dependent on the random number generator that is used. In principle, all random number generators are bad, some are less worse than others. It is always a good idea to do a MC simulation with two different random number generators. See, for example, ref. [55] for a discussion about this topic.
- If you see numbers try to understand them and check whether they are reasonable. Since we use reduced units most properties are in the range $[-1; 1]$ this implies that if we find -2.4 it is probably all right, but if we find 2.4×10^{18} you should be suspicious.
 - If you are writing a parallel code, we strongly recommend you to make a detailed work-plan before you start programming. A common mistake is that you forget to pass system variables (for example, temperature or masses of the particles) to all processors.

Chapter 16

Appendix H: Software

barrier1	5.1	MD of a single particle in various ensembles
barrier2	7.1	Barrier crossing of a single particle using rare events
boltzmann	2.2	Boltzmann distribution
cbmc	8.2	CBMC program for a simple system
distribution	2.1	Calculate the distribution of particles among compartments
ewald	3.6	The Ewald summation
gibbs	6.1	Gibbs ensemble of LJ particles
harddisks	3.3	Simulation of hard disks in a square
ising	5.3	2D Ising Model using multi-canonical simulations
mc-lj	3.4/6.1	MC simulation of LJ particles in the NVT ensemble
mc-npt	5.2	MC simulation of hard spheres in the NPT ensemble
md-lj	4.1	MD program of LJ particles
md-mpi	4.2	Parallel MD program of LJ particles
oscillators	2.3	System of coupled harmonic oscillators
overlap	8.3	Overlapping distribution for polymers
photon	3.2	MC simulation of a photon gas
pi	3.1	Calculation of PI using brute force MC
pt	3.7	Parallel Tempering
polymer	8.1	CBMC of a single chain molecule
random1d	2.4	1D random walk of a single particle
random2d	2.5	2D random walk of multiple particles
scaling	3.5	MC algorithm in which coordinates are scaled
tps	7.2	Transition Path Sampling
umbrella	6.2	Umbrella sampling for a particle over an energy surface

Acknowledgements

We would like to thank Roland van Vliet for a critical reading of the manuscript and all students from the March 1999 and January 2000 courses in Amsterdam for their useful comment. Thanks to Marieke Kranenburg for her contribution to the January 2000 course.

Bibliography

- [1] <http://molsim.chem.uva.nl/course>
- [2] Frenkel, D.; Smit, B., *Understanding Molecular Simulations: from Algorithms to Applications* Academic Press; San Diego, 1996.
- [3] Allen, M.P.; Tildesley, D.J., *Computer Simulation of Liquids* Clarendon Press; Oxford, 1987.
- [4] Chandler, D., *An Introduction to Modern Statistical Mechanics* Oxford University Press; New York, 1987.
- [5] Rapaport, D.C., *The art of molecular dynamics simulation* Cambridge University Press; Cambridge, 1995.
- [6] Atkins, P.W., *Physical Chemistry* Oxford University Press; New York, 1998.
- [7] Matsumoto, M.; Nishimura, T. *ACM Trans. on Modeling and Computer Simulation* **1998**, 8, 3–30.
- [8] <http://random.mat.sbg.ac.at>
- [9] <http://egcs.cygnus.com>
- [10] <http://www.gnu.org/software/gcc/gcc.html>
- [11] Geyer, C.J.; Thompson, E.A. *J. Am. Stat. Assoc.* **1995**, 90, 909–920.
- [12] Falcioni, M.; Deem, M.W. *J. Chem. Phys.* **1999**, 110, 1754–1766.
- [13] Wu, M.G.; Deem, M.W. *Mol. Phys.* **1999**, 97, 559–580.
- [14] Yan, Q.L.; Pablo, J.J.de *J. Chem. Phys.* **1999**, 111, 9509–9516.
- [15] Martyna, G.J.; Tuckerman, M.; Tobias, D.J.; Klein, M.L. *Mol. Phys.* **1996**, 87, 1117–1157.
- [16] Naghizadeh, J.; Rice, S.A. *J. Chem. Phys.* **1962**, 36, 2710–2720.
- [17] Swope, W.C.; Andersen, H.C.; Berens, P.H.; Wilson, K.R. *J. Chem. Phys.* **1982**, 76, 637–649.
- [18] <http://www.mpi.nd.edu/lam/>
- [19] Martyna, G.J.; Klein, M.L.; Tuckerman, M. *J. Chem. Phys.* **1992**, 97, 2635–2645.
- [20] Berendsen, H.J.C.; Postma, J.P.M.; Gunsteren, W.F. van; DiNola, A.; Haak, J.R. *J. Chem. Phys.* **1984**, 81, 3684–3690.
- [21] Smit, B.; Frenkel, D. *Mol. Phys.* **1989**, 68, 951–958.
- [22] Bennett, C.H., in *Diffusion in Solids: Recent Developments*, edited by Nowick, A.S.; Burton, J.J. Academic Press; New York, 1975, pp. 73–113.
- [23] Chandler, D. *J. Chem. Phys.* **1978**, 68, 2959–2970.
- [24] Bolhuis, P.G.; Dellago, C.; Chandler, D. *Faraday Discuss.* **1998**, 110, 421–436.
- [25] Dellago, C.; Bolhuis, P.G.; Chandler, D. *J. Chem. Phys.* **1999**, 110, 6617–6625.
- [26] Siepmann, J.I.; Frenkel, D. *Mol. Phys.* **1992**, 75, 59–70.
- [27] Laso, M.; Pablo, J.J.de ; Suter, U.W. *J. Chem. Phys.* **1992**, 97, 2817–2819.
- [28] Mooij, G.C.A.M.; Frenkel, D.; Smit, B. *J. Phys.: Condens. Matter* **1992**, 4, L255–L259.
- [29] Mooij, G.C.A.M.; Frenkel, D. *J. Phys.: Condens. Matter* **1994**, 6, 3879–3888.
- [30] <http://www.linux.org>

- [31] <http://www.redhat.com>
- [32] <http://www.adobe.com/acrobat>
- [33] <http://www.dsm.fordham.edu/ftnchek/>
- [34] <http://plasma-gate.weizmann.ac.il/Xmgr/>
- [35] <http://www.fortran.com/fortran/market.html>
- [36] <http://www.gnu.org>
- [37] <http://www.mol.biol.ethz.ch/wuthrich/software/molmol/>
- [38] Feynman, R.P.; Leighton, R.B.; Sands, M., *The Feynmann lectures on physics* Addison-Wesley; Reading, 1965.
- [39] Goldstein, H., *Classical Mechanics*, 2nd ed. Addison-Wesley; Reading, 1980.
- [40] <http://www.whatisthematrix.com>
- [41] Rull, L.F.; Jackson, G.; Smit, B. *Mol. Phys.* **1995**, *85*, 435–447.
- [42] Vlugt, T.J.H.; Martin, M.G.; Smit, B.; Siepmann, J.I.; Krishna, R. *Mol. Phys.* **1998**, *94*, 727–733.
- [43] Hansen, J.P.; McDonald, I.R., *Theory of Simple Liquids*, 2nd ed. Academic Press; London, 1986.
- [44] Ven-Lucassen, I.M.J.J.van de ; Vlugt, T.J.H.; Zanden, A.J.J.van der ; Kerkhof, P.J.A.M. *Mol. Phys.* **1998**, *94*, 495–503.
- [45] <http://www.beowulf.org>
- [46] <http://molsim.chem.uva.nl/cluster>
- [47] MPI: A Message-Passing Interface Standard (<http://www.mpi-forum.org>)
- [48] <http://www.epm.ornl.gov/pvm/>
- [49] <http://www-unix.mcs.anl.gov/mpi/mpich/>
- [50] Smit, B.; Karaborni, S.; Siepmann, J.I. *J. Chem. Phys.* **1995**, *102*, 2126–2140, erratum: *J. Chem. Phys.* **1998**, *109*, 352.
- [51] Martin, M.G.; Siepmann, J.I. *J. Phys. Chem. B* **1998**, *102*, 2569–2577.
- [52] Tuckerman, M.; Berne, B.J.; Martyna, G.J. *J. Chem. Phys.* **1992**, *97*, 1990–2001.
- [53] <http://www.dl.ac.uk/CCP/CCP5/main.html>
- [54] Ilario, G.; Tironi, G.; Gunsteren, W.F. van *Mol. Phys.* **1994**, *83*, 381–403.
- [55] Hellekalek, P. *Mathematics and Computers in Simulation* **1998**, *46*, 485–505.