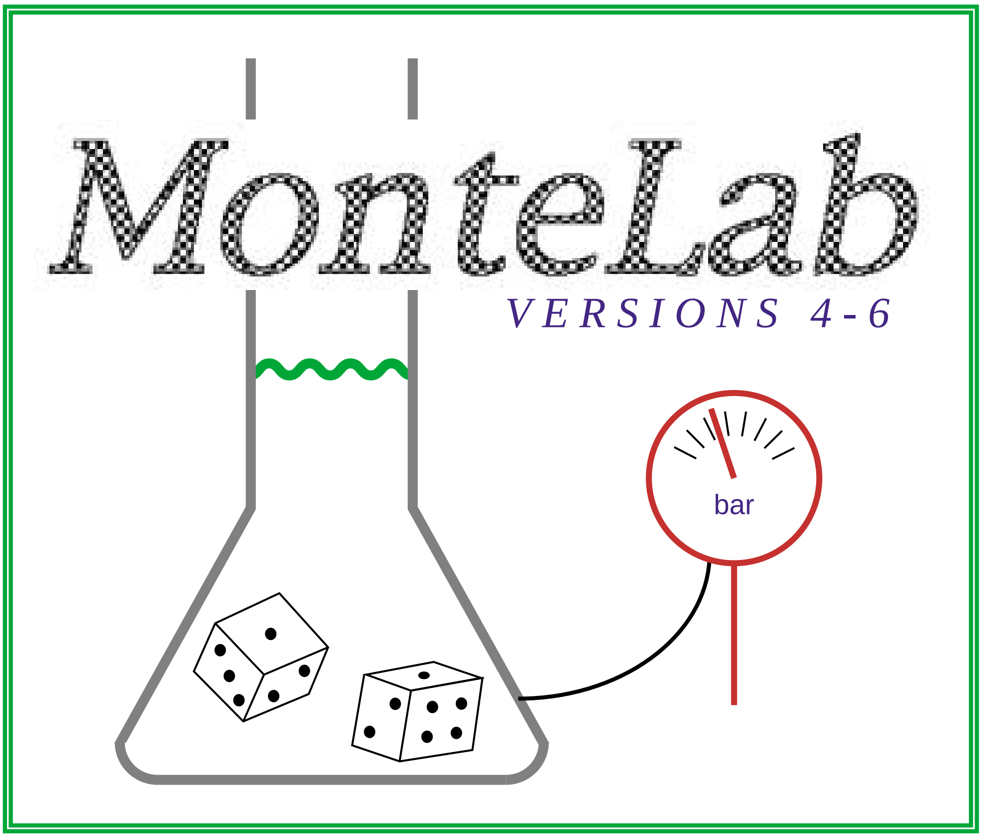


# User's Guide

Jan Westergren Sture Nordholm
Department of Experimental Physics - Chalmers University of Technology
Department of Physical Chemistry - Göteborg University
Göteborg - Sweden

April 2005

# About MonteLab

MonteLab is a program that employs the Metropolis Monte Carlo simulation method for calculating thermodynamical properties in bulk or particle systems (clusters). It was originally developed as an educational program under the VisAb project ("A Visualization of Abstract Concepts in Physical Chemistry by Computer" [1]) which was financed by Högskoleverket (the Swedish National Agency for Higher Education) but it has now also found its way into scientific research. MonteLab is now introduced as free software and might be downloaded from the MonteLab webpage *www.phc.gu.se/~janw/MonteLab.htm*. The program is written in standard Fortran and should be compilable for Fortran 77 and later version. Bugs reports and improvement suggestions are very welcome to Sture.Nordholm@phc.gu.se.

We believe that MonteLab can serve as a Monte Carlo tool for many people. A necessity when distributing software to new users is a good and extensive documentation. Hopefully, with this User's Guide and with a well-commented code, MonteLab will be easy to use and understand.

The number of possible areas of use for a Monte Carlo program is of course enormous, and a program that is not easy to modify has no future outside the group of original users. Therefore, much effort has been put into writing a clear and organized code with many comments. The program is organized as a main program that should suit many different systems and modules that may be modified for different systems. A Programmer's Guide is available as a help when modifying the program. The Programmer's Guide is downloadable from the MonteLab webpage. Every reasonable effort has been made to eliminate errors in the program, but the authors cannot assume responsibility for the consequences of their use.

## MonteLab version 6

www.phc.gu.se/~janw/MonteLab.htm

Jan Westergren
Department of Experimental Physics
Molecular Physics Group
Chalmers University of Technology and
Göteborg University
Göteborg - Sweden

Professor Sture Nordholm
Department of Physical Chemistry
Göteborg University
Göteborg - Sweden

Any contact about MonteLab should be taken with Professor Sture Nordholm
*e-mail* Sture.Nordholm@phc.gu.se

## About the User's Guide

In this User's Guide the function of MonteLab as well as the underlying statistical thermodynamics are described. To each system, the theory is first given followed by special features of MonteLab and then instructions are given how to give input and how to interpret output data. Finally, we present some hints for the simulations and obstacles that accompany the chosen system. For anyone who wishes to understand the fundamentals of the Metropolis Monte Carlo simulation method [2], an excellent description is given by Smit and Frankel [3].

This version of the User's Guide covers the following systems:

1. Simple fluids with Lennard-Jones 12-6 (LJ) or Morse pairwise interaction.
   The canonical ensemble (NVT).
2. Simple fluids with cell constraints with LJ or Morse potential.
   The canonical ensemble (NVT).
3. Clusters of atoms interacting with the MBA (Many-Body Alloy) potential.
   The canonical ensemble (NVT).
4. Clusters of atoms interacting with the MBA (Many-Body Alloy) potential.
   The microcanonical ensemble (NVE).

Each new chapter is based on the previous ones.

## News in MonteLab versions 4, 5 and 6

The version 4 of MonteLab was a totally rewritten program compared to version 3.2. The program was reorganized in a main program and exchangeable modules as mentioned above. The original function was not changed but complemented. One error has been found in version 4, namely that the new atom to be moved was not chosen randomly but in an ordered sequence (See Sec. 1.8). With this error modified to the proper random choice, we introduced version 5. In a few tests we observed no difference in results between version 4 and 5.

In version 6, the random numbers are generated in a new order and registered in a file so that an interrupted simulation may be continued. It is also possible to zoom in a simulation. The $g(r)$ is in version 6 replaced by the $s(r)$ function for particle systems (clusters).

The policy of version number is that the same input data (as e.g. the initial random number) should always generate exactly the same output (as long as the same computer is used) when the same version of the program has been used.

*Jan Westergren*
*April 2005*

# SIMPLE FLUIDS WITH LENNARD-JONES 12-6 (LJ) OR MORSE PAIR-POTENTIALS IN THE CANONICAL ENSEMBLE (NVT)

## THEORY

### 1.1. Periodic boundary constraints

The huge number of particles in for instance a liquid can naturally not be simulated by a computer and the common way to simulate a bulk-like system is to use periodic boundary constraints (PBC). An active cube is cut out of the bulk system and the particles in that cube are rcalled active particles (Fig. 1.1). That cube is then repeated infinitely many times in all dimensions (Fig. 1.2). In the copies of the cube, the particles are at exactly the same positions as in the active cube. If a particle moves out of the active cube, another image of that particle enters the active cube and becomes the active image of that particle.

Is the system with PBC a realistic representation of a fluid? Well, at least the surface particles are eliminated which is the major problem. In a real fluid, the overwhelming number of particles is far from the surface and that is also true for the PBC system. However, by using PBC, the particles are forced to a certain periodicity. If the length of the active cube is $L$, the system must repeat itself after the distance $L$. If there is no structure in the real system, for instance in an ideal gas, this forced periodicity probably has a minor effects on the thermodynamic properties. But at temperatures and pressures where the fluid should segregate into droplets in gas or bubbles in liquid, the PBC might hinder realistic configurations. Say that in a realistic fluid, droplets of diameter $2L$ are surrounded by gas. Such configurations are of course impossible to create in the PBC system. Hence, the PBC should be used with the awareness that they are more a way to avoid surface effects than representing a true bulk.

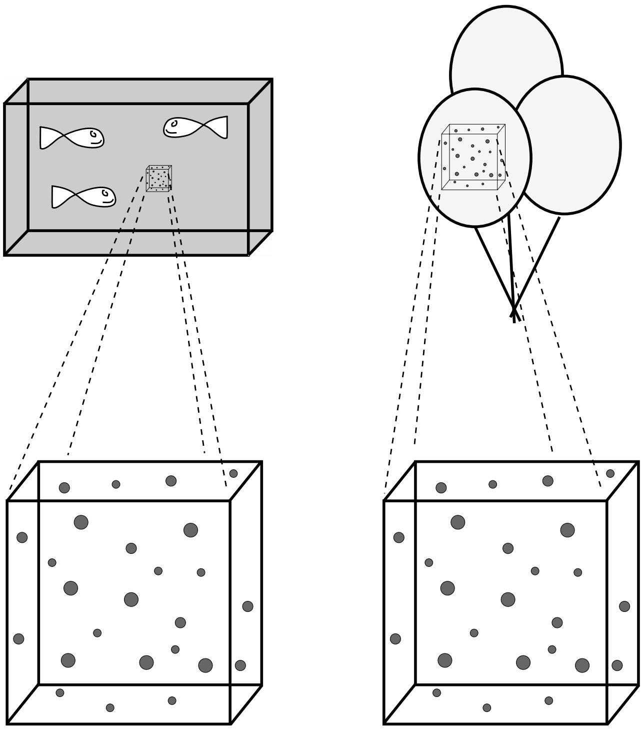


Fig. 1.1 The active cube is cut out from a liquid and a gas.

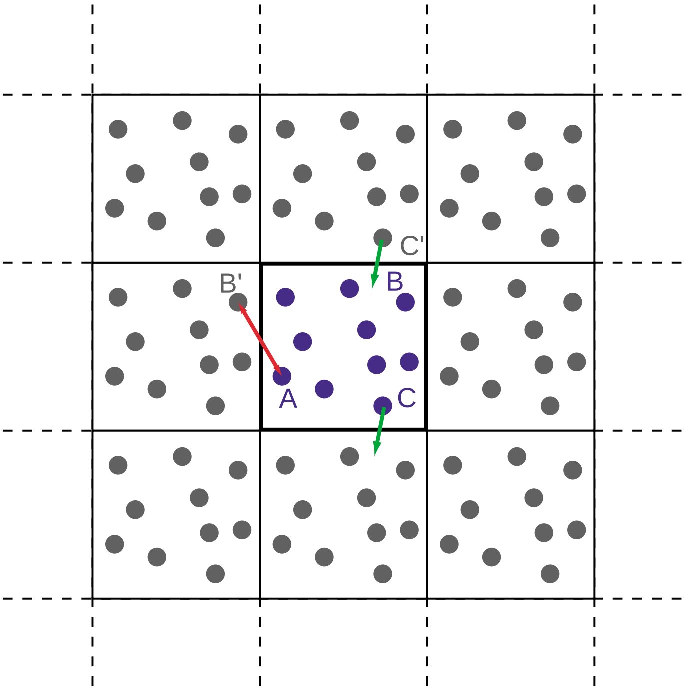


Fig. 1.2. The distance between particles A and B is the distance between the nearest images of the particles. When particle C moves out of the active cube, the image C' becomes the active image.

### 1.2. Energy calculations

In a system that is considered to be canonical, the number of particles in the active cube, *N*, the temperature, *T*, and the volume of the cube, *V*, are fixed. At a certain moment the spatial coordinates and the momenta of particle *i* are $\mathbf{r}_i$ and $\mathbf{p}_i$, respectively. If there are *N* active particles we say that the system has the configuration

$$\mathbf{R} = \left[\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\right]$$

and the momenta vector

$$\mathbf{P} = \left[\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_N\right]$$

The total energy of the system at *T* and *V* is just the time average of the total energy when the particles move around in the liquid. However, according to the *ergodic hypothesis,* instead of taking the time average, we can as well calculate the average of the total energy over all possible configurations and momenta. The average should be weighted with the Boltzmann factor. Hence,

$$\left\langle E_{total} \right\rangle_{time} = \left[\text{ergodic hypothesis}\right] = \frac{\displaystyle\iint\limits_{\text{all }[\mathbf{R},\mathbf{P}]} H\left(\mathbf{R},\mathbf{P}\right)\cdot e^{-H(\mathbf{R},\mathbf{P})/k_bT} d\mathbf{R}d\mathbf{P}}{\displaystyle\iint\limits_{\text{all }[\mathbf{R},\mathbf{P}]} e^{-H(\mathbf{R},\mathbf{P})/k_bT} d\mathbf{R}d\mathbf{P}}.$$

(Here and in the text below double integral symbol will be used although the integral is multi-dimensional. In this case the integral is 6*N*-dimensional. For the PBC fluid, the integrations should be over all momenta and over all configurations in the active cube, i.e. $0 \le x_1, y_1, z_1, x_2, \ldots, z_N \le L,$ where *L* is the side length of the active cube.) *H* is the Hamiltonian and $e^{-H(\mathbf{R},\mathbf{P})/k_bT}$ is the Boltzmann factor. In the canonical ensemble, the probability for the system to be in $\left[\mathbf{R},\mathbf{P}\right]$ is proportional to the Boltzmann factor.

In the canonical ensemble we can separate **R** and **P** as the kinetic energy only depends on **P** and the potential energy only depends on **R**. The integral might then be simplified

$$\begin{aligned}\left\langle E_{total} \right\rangle &= \frac{\displaystyle\iint\limits_{\text{all }[\mathbf{R},\mathbf{P}]} \left(E_{kin}\left(\mathbf{P}\right)+U\left(\mathbf{R}\right)\right)\cdot e^{-\left(E_{kin}(\mathbf{P})+U(\mathbf{R})\right)/k_bT} d\mathbf{R}d\mathbf{P}}{\displaystyle\iint\limits_{\text{all }[\mathbf{R},\mathbf{P}]} e^{-\left(E_{kin}(\mathbf{P})+U(\mathbf{R})\right)/k_bT} d\mathbf{R}d\mathbf{P}} \\ &= \frac{\displaystyle\iint\limits_{\text{all }\mathbf{P}} E_{kin}\left(\mathbf{P}\right)\cdot e^{-E_{kin}(\mathbf{P})/k_bT} d\mathbf{P}}{\displaystyle\iint\limits_{\text{all }\mathbf{P}} e^{-E_{kin}(\mathbf{P})/k_bT} d\mathbf{P}} + \frac{\displaystyle\iint\limits_{\text{all }\mathbf{R}} U\left(\mathbf{R}\right)\cdot e^{-U(\mathbf{R})/k_bT} d\mathbf{R}}{\displaystyle\iint\limits_{\text{all }\mathbf{R}} e^{-U(\mathbf{R})/k_bT} d\mathbf{R}} \\ &= \left\langle E_{kin} \right\rangle + \left\langle U \right\rangle, \end{aligned} \tag{1.1}$$

where *U* is the potential energy.

The average kinetic energy is simply

$$\langle E_{kin}\rangle = \frac{3N}{2} k_b T,$$

thus the difficult task is to calculate $\langle U\rangle$. First of all we must know how to calculate the potential energy for a single configuration, $U(\mathbf{R})$. In the simplest case we use a pair potential to describe the interaction between the particles. With such a potential the total potential energy is

$$U(\mathbf{R}) = \sum_{i=1}^{N}\sum_{j=i+1}^{N}\theta(r_{ij}) = \frac{1}{2}\sum_{i,j\neq i}\theta(r_{ij}),$$

where $r_{ij}$ is the distance between the particle $i$ and the nearest image of particle $j$. That image might be an image in a neighbour cube and not in the active cube (see Fig. 1.2). The most common of all potentials is the Lennard-Jones (12-6) potential that is given by

$$\theta(r_{ij}) = 4\varepsilon\left(\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right).$$

This potential is drawn as the solid line in Fig. 1.3. The well depth is set by the parameter $\varepsilon$, and the equilibrium distance $r_0$, i.e. the bond length in a very cold dimer, is set by $\sigma$ ($r_0 = \sqrt[6]{2}\sigma$). In Appendix 1.1, potential parameters are found for some common atoms and molecules. Another commonly used pair potential is the Morse potential:

$$\theta(r_{ij}) = D\left(e^{-2\alpha(r-r_e)} - 2e^{-\alpha(r-r_e)}\right).$$

Three parameters are to be specified for this potential. Besides the well depth, $D$, and the equilibrium distance, $r_e$, the curvature at equilibrium distance is given by $\kappa = 2D\alpha^2$. The LJ and Morse potentials have the same well depth, equilibrium distance and curvature at the equilibrium distance when $D=\varepsilon$, $r_e = 2^{1/6}\sigma$ and $\alpha = 2^{-1/6}\cdot 6/\sigma\cdot\sqrt{\varepsilon/D}$. This Morse curve is drawn as the dashed line in Fig. 1.3. Since the interaction between particles fall off rapidly with $r$ it is common to truncate the potential at a certain so-called cut-off distance in order to save computing time. In MonteLab, the potential is set to zero for distances longer than $r_{cut\text{-}off}$, which is a user-defined parameter.

The LJ potential is especially handy for simulations as the units of the parameters (energy and length units) can be chosen independently of each other. Thus, regardless of the kind of particles in the fluid we can choose the units so that $\varepsilon = 1$ energy unit and $\sigma = 1$ length unit. Consequently, all fluids can be described by the parameters $\varepsilon = 1$ and $\sigma = 1$, only the units differ. The advantage is that a simulation is valid for all fluids, except that both input and output data will convert to different values in SI units. For instance, the input temperature $k_b T = 1$ energy unit corresponds to different SI temperatures depending on if the fluid is neon or argon. These varying units are called reduced units and the recipe to convert reduced units into SI units is given in Appendix 1.2.

The situation is different for the Morse potential. We can still let the energy and length units vary so that for all fluids, $D_e = 1$ reduced energy unit and $r_e = 2^{1/6}$ reduced length units. The third parameter $\alpha$ has however the dimension length$^{-1}$ and since the length unit is already set to define $r_e$, the parameter $\alpha$ must be given a numerical value before a simulation starts. A simulation can thus only represent fluids with the predefined ratio $\alpha / r_e$.

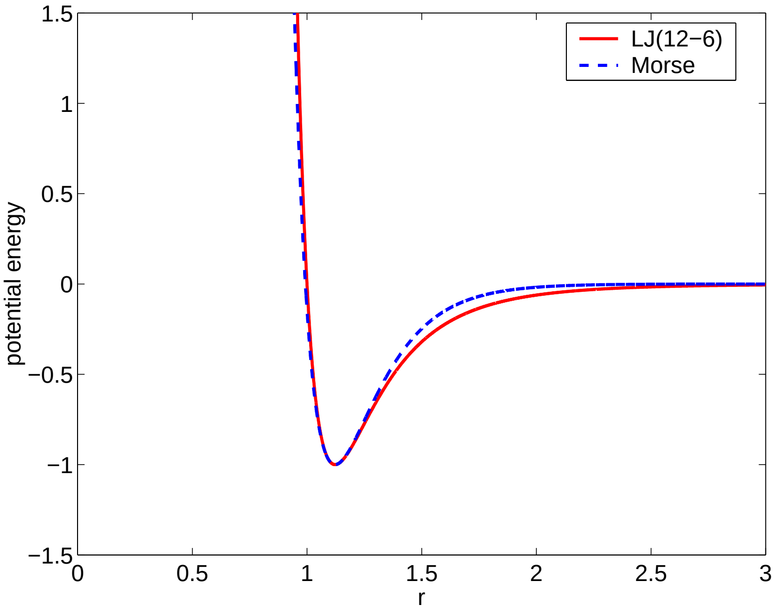


Fig. 1.3 The Lennard-Jones and the Morse potentials. $\alpha$ is chosen so that the curvature of the two curves is equal at $r_0$. Distances and energies are given in reduced units.

When the potential is given, $U(\mathbf{R})$ can be calculated for any configuration $\mathbf{R}$. According to Eq. 1.1, only an accurate numerical integration method is needed in order to calculate the average potential energy. Unfortunately, the integral should be over all spatial coordinates. Say that we have 512 active particles, then the integral is 1536-dimensional. A numerical evaluation of such an integral is not doable. Instead, let us randomly generate configurations and calculate the average over them. Let us imagine that we have an algorithm that would generate a set of random configurations $\{\mathbf{R}_k\}_{k=1}^{W}$ according to the Boltzmann distribution,

$$\text{Probability}(\mathbf{R}_k) \propto e^{-U(\mathbf{R}_k)/k_bT} d\mathbf{R}.$$

Then the average potential energy would be just the average over that set of configurations,

$$\langle U\rangle = \frac{\iint\limits_{\text{all }\mathbf{R}} U(\mathbf{R})\cdot e^{-U(\mathbf{R})/k_bT}\,d\mathbf{R}}{\iint\limits_{\text{all }\mathbf{R}} e^{-U(\mathbf{R})/k_bT}\,d\mathbf{R}} = \frac{1}{W}\sum_{k=1}^{W} U(\mathbf{R}_k) \quad \text{as } W\to\infty. \tag{1.2}$$

The method to generate $\{\mathbf{R}_k\}_{k=1}^{W}$ is the Metropolis Monte Carlo method that will be described in Sec. 1.8. This method to calculate the integral has proved to be superior to a normal numerical integration for thermodynamical systems.

## 1.3. Heat capacity calculations

The heat capacity at constant volume is easily obtained in the canonical ensemble, as it is the fluctuation in energy,

$$C_V = \langle E^2\rangle - \langle E\rangle^2 = \frac{3N}{2}k_B + \langle U^2\rangle - \langle U\rangle^2 .$$

The first term in the last expression is the heat capacity of the momenta. The total heat capacity is obtained in a Monte Carlo simulation by

$$C_V = \frac{3N}{2}k_B + \frac{1}{W}\sum_{k=1}^{W} U^2(\mathbf{R}_k) - \left(\frac{1}{W}\sum_{k=1}^{W} U(\mathbf{R}_k)\right)^2 \quad \text{as } W\to\infty.$$

## 1.4. The pressure

In the canonical ensemble it is the temperature, volume and number of particles that are fixed. The pressure, in contrast, fluctuates. The average pressure is however a quantity of great importance and by simulations we will try to calculate it for different temperatures and particle densities. The pressure of a canonical ensemble is generally expressed as

$$P = -\left(\frac{\partial F}{\partial V}\right)_{T,V}$$

where Helmholtz free energy is derived from the partition function:

$$F = -k_B T\ln Q = -k_B T\left(\ln Q_{kin} + \ln Q_{pot}\right).$$

The partition function for the kinetic energy is

$$Q_{kin} = \frac{1}{N!}\left(\frac{V}{\Lambda^3}\right)^N = \frac{V^N}{N!\left(\frac{h}{\sqrt{2\pi m k_B T}}\right)^{3N}} \Rightarrow \qquad P_{kin} = P_{ideal} = \frac{Nk_BT}{V}$$

($\Lambda$ is called the *thermal de Broglie wavelength*.) In an ideal gas, where the potential energy is constant, $\ln Q_{pot} = 0$ and the pressure solely originates from the kinetic energy. In the general case however, the partition function for the potential energy of a PBC fluid is

$$Q_{pot} = \frac{1}{V^N} \iint\limits_{\text{all } \mathbf{R}} e^{-U(\mathbf{R})/k_bT} d\mathbf{R} = \frac{1}{V^N} \int_0^L dx_1 \int_0^L dy_1 ... \int_0^L dz_N e^{-U(x_1,y_1,...,z_N)/k_bT} .$$

The side length of the active cube is $L$ and the volume is $V = L^3$. In order to calculate the contribution to the pressure from the potential energy we rescale the coordinates to $\mathbf{S} = \mathbf{R}/L = \mathbf{R}/V^{1/3}$ :

$$P_{pot} = -\frac{\partial}{\partial V}\left[-k_BT \ln Q_{pot}\right] = k_BT \frac{\frac{\partial Q_{pot}}{\partial V}}{Q_{pot}}$$

$$\begin{aligned}
\frac{\partial Q_{pot}}{\partial V} &= \frac{\partial}{\partial V}\left[\frac{\left(V^{1/3}\right)^{3N}}{V^N} \int_0^1 ds_{x_1} \int_0^1 ds_{y_1} ... \int_0^1 ds_{z_N}\, e^{-U\left(V^{1/3}s_{x_1}, V^{1/3}s_{y_1},...,V^{1/3}s_{z_N}\right)/k_BT}\right] \\
&= -\frac{1}{k_BT} \int_0^1 ds_{x_1} \int_0^1 ds_{y_1} ... \int_0^1 ds_{z_N}\, e^{-U/k_BT} \cdot \frac{\partial}{\partial V}\left[U\left(V^{1/3}s_{x_1}, V^{1/3}s_{y_1},...,V^{1/3}s_{z_N}\right)\right] \\
&= \left[\text{the chain rule for derivatives}\right] \\
&= -\frac{1}{k_BT} \int_0^1 ds_{x_1} \int_0^1 ds_{y_1} ... \int_0^1 ds_{z_N}\, e^{-U/k_BT} \cdot \left(\frac{\partial\left(V^{1/3}s_{x_1}\right)}{\partial V} \cdot \frac{\partial U}{\partial\left(V^{1/3}s_{x_1}\right)} + \frac{\partial\left(V^{1/3}s_{y_1}\right)}{\partial V} \cdot \frac{\partial U}{\partial\left(V^{1/3}s_{y_1}\right)} + ...\right) \\
&= -\frac{1}{k_BT} \int_0^1 ds_{x_1} \int_0^1 ds_{y_1} ... \int_0^1 ds_{z_N}\, e^{-U/k_BT} \cdot \left(\frac{V^{1/3}s_{x_1}}{3V} \cdot \frac{\partial U}{\partial\left(V^{1/3}s_{x_1}\right)} + \frac{V^{1/3}s_{y_1}}{3V} \cdot \frac{\partial U}{\partial\left(V^{1/3}s_{y_1}\right)} + ...\right) \\
&= -\frac{1}{k_BT \cdot 3V \cdot V^N} \int_0^L dx_1 \int_0^L dy_1 ... \int_0^L dz_N e^{-U/k_BT} \cdot \left(x_1 \cdot \frac{\partial U}{\partial x_1} + y_1 \cdot \frac{\partial U}{\partial y_1} + ...\right)
\end{aligned}$$

$$\Rightarrow P_{pot} = -\frac{1}{3V} \frac{\int_0^L dx_1 \int_0^L dy_1 ... \int_0^L dz_N e^{-U/k_BT} \cdot \left(x_1 \cdot \frac{\partial U}{\partial x_1} + y_1 \cdot \frac{\partial U}{\partial y_1} + ...\right)}{\int_0^L dx_1 \int_0^L dy_1 ... \int_0^L dz_N e^{-U/k_BT}} \tag{1.3}$$

This formula is unfortunately exactly correct only when $L$ and $N$ are infinite. When periodic boundary constraints are introduced, the derivatives should be with respect to $x_i - L$ instead of $x_i$ when nearest neighbours are not in the active cube. Fortunately, the errors cancel totally if we regard the potential as being a function of distances between particles instead of the coordinates. We set

$$r_{ij} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2} .$$

Recall that the distance is between the nearest images of particles *i* and *j*. Now, let us substitute the variables in Eq. 1.3:

$$\frac{\partial r_{ij}}{\partial x_i} = \frac{x_i - x_j}{r_{ij}} = -\frac{\partial r_{ij}}{\partial x_j} \Rightarrow$$

$$x_1 \frac{\partial}{\partial x_1} = x_1 \sum_{i,j>i} \frac{\partial r_{ij}}{\partial x_1} \cdot \frac{\partial}{\partial r_{ij}} = x_1 \sum_{j=2}^{N} \frac{\partial r_{1j}}{\partial x_1} \cdot \frac{\partial}{\partial r_{1j}} = \sum_{j=2}^{N} \frac{x_1 \left( x_1 - x_j \right)}{r_{1j}} \cdot \frac{\partial}{\partial r_{1j}}, \quad \text{etc.} \tag{1.4}$$

If we now rearrange all the derivatives in Eq. 1.4 we can conclude that

$$\sum_{i=1}^{N} x_i \cdot \frac{\partial}{\partial x_i} + y_i \cdot \frac{\partial}{\partial y_i} + z_i \cdot \frac{\partial}{\partial z_i} =$$

$$\sum_{i,j>i} \left( \frac{x_i \left( x_i - x_j \right)}{r_{ij}} + \frac{x_j \left( x_j - x_i \right)}{r_{ij}} + \frac{y_i \left( y_i - y_j \right)}{r_{ij}} + \frac{y_j \left( y_j - y_i \right)}{r_{ij}} + \frac{z_i \left( z_i - z_j \right)}{r_{ij}} + \frac{z_j \left( z_j - z_i \right)}{r_{ij}} \right) \cdot \frac{\partial}{\partial r_{ij}} =$$

$$\sum_{i,j>i} r_{ij} \frac{\partial}{\partial r_{ij}}$$

We might now allow also $i > j$, only remembering to divide the sum by two. Eq. 1.3 is equivalent to

$$P_{pot} = -\frac{1}{3V} \frac{\int_0^L dx_1 \int_0^L dy_1 ... \int_0^L dz_N e^{-U/k_b T} \cdot \frac{1}{2} \sum_{i,j \neq i}^{N} r_{ij} \frac{\partial U}{\partial r_{ij}}}{\int_0^L dx_1 \int_0^L dy_1 ... \int_0^L dz_N e^{-U/k_b T}}.$$

Defining the virial energy

$$E_{vir}\left(\mathbf{R}\right) = \frac{1}{2} \sum_{i,j \neq i}^{N} -\frac{1}{3} r_{ij} \frac{\partial U}{\partial r_{ij}},$$

we conclude

$$P_{pot} = \frac{1}{V} \cdot \frac{\iint_{\text{all } \mathbf{R}} E_{vir}\left(\mathbf{R}\right) \cdot e^{-U(\mathbf{R})/k_b T} d\mathbf{R}}{\iint_{\text{all } \mathbf{R}} e^{-U(\mathbf{R})/k_b T} d\mathbf{R}}, \tag{1.5}$$

and, using the set of configurations $\left\{ \mathbf{R}_k \right\}_{k=1}^{W}$, the pressure of the system is

$$P = \frac{N k_B T}{V} + \frac{1}{V} \cdot \frac{1}{W} \sum_{k=1}^{W} E_{vir}\left(\mathbf{R}_k\right) \quad \text{as } W \to \infty.$$

## 1.5. The radial distribution function

The radial distribution function gives information about the structure of the particles. In a perfect crystal, the particles are ordered in a three-dimensional pattern. Looking at the neighbours with the eyes of one of the particles, the neighbours will be at specific distances, *r*. The radial distribution function, $g(r)$, tells how many neighbours there are per volume in the spherical shell at the distance *r* from a specific particle.

In a totally homogeneous fluid, $g(r)$ will be the same for all particles but in, for instance, a droplet/gas mixture the surrounding of a particle depends on whether it is in the droplet or in the gas. In such a case $g(r)$ is the average for all particles in the fluid. The radial distribution function is in other words the average local density around a particle. As a last step the distribution is normalized with the global particle density $\rho = N/V$. We end up at the definition

$$g(r) = \frac{\text{average number of atoms in the shell } [r, r+dr] / 4\pi r^2 dr}{\rho}.$$

In a liquid, the particles do not form a perfect pattern but there is still a structure on a local scale. Each particle tends to be surrounded by a few layers of particles. Further out, however, the particles are found totally at random (in the eyes of the particle we have picked) and $g(r)$ approaches unity. The definition of $g(r)$ is illustrated on a liquid in Fig. 1.4.

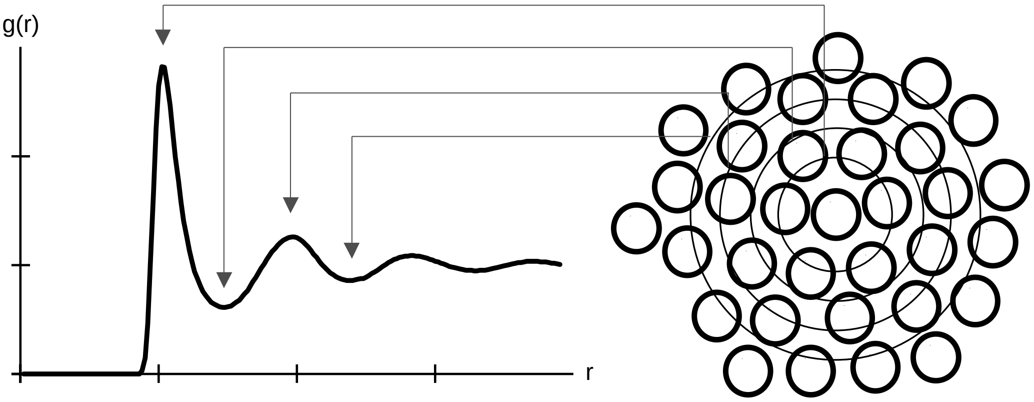


Fig. 1.4. The definition of the radial distribution function. In a liquid, wells and peaks follow each other until *g*(*r*) finally approaches unity. The illustration is inspired by Roland Kjellander, Göteborg University.

In a gas, the wells and peaks weaken and $g(r)$ is close to unity already for small *r*. In Figs. 1.5a-c, typical radial distribution functions are drawn for a solid, liquid and a gas, respectively. From the shape of a curve it is possible to tell the phase of a fluid and hence $g(r)$ is a very useful function.

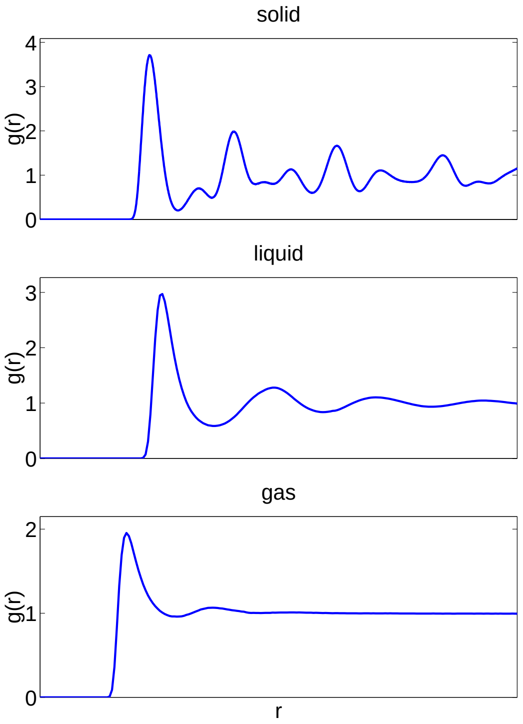


Fig. 1.5. The radial distribution function from simulations of a solid, liquid and a gas. Note that the $r$ scales are different in the three figures.

The $g(r)$ is obtainable from a Monte Carlo simulation. Let us divide the $r$ axis into $M$ equal parts from $r = 0$ to $r = L/2$. Since the largest sphere that might be placed within the active cube has the radius $L/2$, which is the largest $r$ for which a correct value of $g(r)$ can be obtained. The average number of neighbours in an interval is

$$\rho \int_{r_{I-1}}^{r_I} g(r) \cdot 4\pi r^2 dr = \left\langle \frac{\text{number of } r_{ij} \text{ in the interval } [r_{I-1}, r_I]}{N} \right\rangle .$$

where the average is over the set of configurations $\{\mathbf{R}_k\}_{k=1}^{W}$ that were generated in the simulation.

In the case of pair potentials, $g(r)$ contains enough information to calculate the average potential energy and the pressure. With the pair potential $\theta(r)$ and the "virial pair function" $\varphi(r) = -r\theta'(r)/3$, we find the following equations that are equivalent to Eqs. 1.2 and 1.5, respectively:

$$\langle U \rangle = \frac{N}{2} \int_0^{\infty} \theta(r) \rho g(r) \cdot 4\pi r^2 dr$$

$$P_{pot} = \frac{N}{2V} \int_0^{\infty} \varphi(r) \rho g(r) \cdot 4\pi r^2 dr .$$

The radial distribution function is only sampled up to $L/2$. The common procedure is to approximate $g(r) \equiv 1$ when $r > L/2$. The integrals beyond $L/2$ are for the LJ and Morse potentials analytical and the results are presented in Sec. 1.12, "Tail correction for the potential cut-off".

## 1.6. Pressure derivatives

(The derivation in this section is only valid for pair-potentials.)

The critical point of a fluid is defined as the point where

$$\frac{\partial P}{\partial V} = \frac{\partial^2 P}{\partial V^2} = 0$$

MonteLab calculates these two derivatives as one method to find the critical point. We have however seen that the relative accuracy is often not good enough.

Let us first define two functions with the dimension energy in terms of the "virial pair function"

$$\psi(r) = \frac{r \cdot \varphi'(r)}{3}$$
$$\xi(r) = \frac{r \cdot \psi'(r)}{3}.$$

The pressure is, as seen above,

$$P = \frac{Nk_BT}{V} + \frac{k_BT}{Q_{pot}}\left(\frac{\partial Q_{pot}}{\partial V}\right)_{T,V} \Rightarrow$$

$$\frac{\partial P}{\partial V} = -\frac{Nk_BT}{V^2} + k_BT\left(\frac{\frac{\partial^2 Q_{pot}}{\partial V^2}}{Q_{pot}} - \frac{\left(\frac{\partial Q_{pot}}{\partial V}\right)^2}{Q^2_{pot}}\right)_{T,V} \Rightarrow$$

$$\frac{\partial^2 P}{\partial V^2} = \frac{2Nk_BT}{V^3} + k_BT\left(\frac{\frac{\partial^3 Q_{pot}}{\partial V^3}}{Q_{pot}} - 3\frac{\frac{\partial^2 Q_{pot}}{\partial V^2} \cdot \frac{\partial Q_{pot}}{\partial V}}{Q^2_{pot}} + 2\frac{\left(\frac{\partial Q_{pot}}{\partial V}\right)^3}{Q^3_{pot}}\right)_{T,V}$$

The derivatives may now be expressed in the functions $\varphi(r)$, $\psi(r)$ and $\xi(r)$

$$
\begin{aligned}
\frac{\partial P}{\partial V} &= -\frac{Nk_BT}{V^2}+\frac{1}{V^2}\left(\left\langle\{\psi\}\right\rangle-\left\langle\{\varphi\}\right\rangle\right)+\frac{1}{k_BT\cdot V^2}\left(\left(\left\langle\{\varphi\}\{\varphi\}\right\rangle-\left\langle\{\varphi\}\right\rangle^2\right)\right) \\
\frac{\partial^2 P}{\partial V^2} &= \frac{2Nk_BT}{V^3}+\frac{1}{V^3}\left(2\left\langle\{\varphi\}\right\rangle-3\left\langle\{\psi\}\right\rangle+\left\langle\{\xi\}\right\rangle\right) \\
&+\frac{3}{k_BT\cdot V^3}\left(\left\langle\{\varphi\}\right\rangle^2-\left\langle\{\varphi\}\{\varphi\}\right\rangle+\left\langle\{\varphi\}\{\psi\}\right\rangle-\left\langle\{\varphi\}\right\rangle\left\langle\{\psi\}\right\rangle\right) \\
&+\frac{1}{\left(k_BT\right)^2\cdot V^3}\left(2\left\langle\{\varphi\}\right\rangle^3+\left\langle\{\varphi\}\{\varphi\}\{\varphi\}\right\rangle-3\left\langle\{\varphi\}\{\varphi\}\right\rangle\left\langle\{\varphi\}\right\rangle\right)
\end{aligned}
$$

Here the averages should be interpreted according to

$$
\begin{aligned}
&\left\langle\{a\}\right\rangle=\frac{1}{W}\sum_{k=1}^{W}\left(\{a\}\right) \quad \text{as } W\to\infty, \\
&\{a\}=\frac{1}{2}\sum_{i\neq j}^{N}a\left(r_{ij}\right).
\end{aligned}
$$

Unfortunately, there is no simple expression for the tail correction of averages $\left\langle\{\varphi\}\{\varphi\}\right\rangle$, $\left\langle\{\varphi\}\{\psi\}\right\rangle$ or $\left\langle\{\varphi\}\{\varphi\}\{\varphi\}\right\rangle$. Therefore a large $r_{\text{cut-off}}$ must be used when the derivatives are of interest. The importance of a large $r_{\text{cut-off}}$ is emphasized by the fact that the $\varphi,\psi$ and $\xi$ all reach further out than $\theta$.

## 1.7. The number of active particles

The larger the number of active particles, the more realistic is of course the system. On the other hand, it takes shorter time to get a representative set of configurations if the number of particles is small. Thus, with a given computer capacity, few active particles might give a more accurate result under conditions that should lead to a homogeneous fluid.

Some suggestions might be given to the choice of number of active particles. A larger cube should be used if inhomogeneity is suspected be present. The range for which the $g\left(r\right)$ can be calculated is limited to $L/2$. Furthermore, if the cut-off distance $r_{\text{cut-off}}>L/2$, the potential energy will start to directly sense the forced periodicity of the system. See also Sec. 1.13 "The $Q_g$ test".

## 1.8. The standard Monte Carlo simulation

In a Monte Carlo simulation, we want the particles to move around in the active cube in such a way that the configurations are generated according to the

$$\text{Probability}(\mathbf{R}) \propto e^{-U(\mathbf{R})/k_bT} d\mathbf{R}.$$

The particles are let to move randomly instead of according to Newton's equation of motion. They are however moved according to a special scheme that will lead to a correct probability distribution. There are many different schemes that will work, but the method employed in MonteLab is the commonly used algorithm which was developed by Metropolis *et. al.* in 1953 [3].

Let us start with a configuration $\mathbf{R}_k$ with a structure that is representative for the fluid in equilibrium at the given volume and particle density. A new configuration is then generated by a small random displacement of one particle. The particle to move is picked randomly. The new configuration we call a trial configuration $\mathbf{R}_t$. Now the Boltzmann factor is compared for the trial configuration and the original one:

$$\gamma = \frac{\text{Probability}(\mathbf{R}_t)}{\text{Probability}(\mathbf{R}_k)} = e^{-(U(\mathbf{R}_t)-U(\mathbf{R}_k))/k_bT}$$

If the potential energy for the trial configuration is lower than for the original one, i.e. $\gamma > 1$, the new configuration is accepted as the next configuration in the set, $\mathbf{R}_{k+1}$. Otherwise the trial configuration should be accepted with the probability $\gamma$. This is done by comparing $\gamma$ with a uniform random number between 0 and 1, $U[0,1]$. Thus, if $\gamma > U[0,1]$ the trial configuration should be accepted. If $\gamma < U[0,1]$, the trial configuration is rejected and the new configuration should be the same as the original one. Note that for a pair potential the energy difference $U(\mathbf{R}_t) - U(\mathbf{R}_k)$ is equal to the difference in removal energy of the trial particle after and before the trial move.

Now we know how to accept/reject a trial configuration. But how should the displacement of a particle be done? There is no unique recipe for that but there is one restriction: the move of a particle should satisfy so called detailed balance. That is that the probability for *generating* (not accepting) the trial configuration should equal the probability for generating the reverse move. The common way, which is used in MonteLab, is that the *x* coordinate of the particle to move is changed by a uniform random number $U[-\Delta, \Delta]$. The *y* and *z* coordinates are changed analogously, but with new random numbers. Such a move satisfies detailed balance.

### THE MONTELAB MONTE CARLO ALGORITHM

1. Start with a configuration $\mathbf{R}_k$ and pick randomly one of the particles in the active cube.
2. Generate a trial configuration by changing the coordinates of the chosen particle:
$$\begin{aligned} x_t &= x_k + U_1\left[-\Delta,\Delta\right] \\ y_t &= y_k + U_2\left[-\Delta,\Delta\right] \\ z_t &= z_k + U_3\left[-\Delta,\Delta\right] \end{aligned}$$
3. Calculate $\gamma = \dfrac{\text{Probability}\left(\mathbf{R}_t\right)}{\text{Probability}\left(\mathbf{R}_k\right)} = e^{-\left(U(\mathbf{R}_t)-U(\mathbf{R}_k)\right)/k_bT}$
4. If $\gamma > U_4\left[0,1\right]$, then accept the trial configuration as the new $\mathbf{R}_{k+1}$, else let $\mathbf{R}_{k+1} = \mathbf{R}_k$.
5. Increase *k* by one a go back to 1.

This scheme will in principle give the correct distribution of configurations. But is it *ergodic*, i.e. is it possible to go from any configuration to any other configuration within a finite number of steps? There are at least situations when it might require so many steps to go from one configuration to another that it is not possible to run such a simulation on today's computers. One example is the transition from a solid structure to a molten one. The probability to find the way from one phase to another is small which can be thought of as a barrier between the phases that hinders the transition. Another example is the segregation of a homogeneous fluid into bubbles in a liquid. In such cases, the user of a simulation program must find out himself whether a true equilibrium has been established. This is currently a major research field.

Not until MonteLab version 5 was the particle to be moved chosen randomly. In earlier versions the particles were instead picked according to a list so that a specific particle was to be moved regularly every $N^{\text{th}}$ trial. Although such a procedure might give a correct simulation, it is guaranteed to be correct with the random procedure. A few tests were done and no differences were found between the results from MonteLab version 4 and version 5.

The Monte Carlo simulation should give the same result independently of the value of $\Delta$ as the number of generated configurations $W \to \infty$. In a real simulation, however, we would like to set a value that minimizes the statistical error. In order to reduce the statistical error, we would like the particles to probe the phase space as much as possible. Let us define the $\textit{steplength} = \sqrt{3}\Delta$ which is the maximal move a particle can make in one step. At least for liquids, the particles will only be in narrow valleys in the phase space. For configurations outside the valleys, some particles will come too close to each other with a subsequent (very) low probability. Thus, in general, the rate of rejection of trial configurations will increase with the steplength since long steps will take the particles out of the valley. With a small steplength most trial configurations will be accepted but each step is short. With a large steplength most trial configurations will be rejected but a single move may take a particle far away. There is no golden rule how to maximize the probe of the phase space, but a rule of thumb that about 50% of the trial configurations should be rejected has often been used. However, a good rejection rate might vary with kind of system, temperature and particle density. An algorithm is included in MonteLab in which the steplength is adjusted to a desired rejection rate.

## FEATURES OF MONTELAB

### 1.9. Initial configuration and meltdown

The initial configuration of the fluid could either come from an input file or be generated by MonteLab. The initial configuration generated by MonteLab is a simple cubic lattice. The particles will be spread to form an infinite crystal with periodic boundary constraints. If the number of active particles is not an even cubic number, an empty hole will arise in the active cube.

If the initial configuration is given from a file, any configuration is accepted, only with the restriction that $0 < x, y, z < L = \sqrt[3]{N / \rho}$ for all particles. The minimal energy configuration is fcc (face centered cubic) for both the Lennard-Jones and Morse potentials. In order to make a perfect such lattice, the number of particles must correspond to an even cubic number of fcc unit cells. Since the number of particles per fcc unit cell is four, the number of particles must satisfy $N / 4 = \text{integer}^3$.

When the initial configuration is the simple cubic configuration it is clearly not an equilibrium configuration. However, when the simulation proceeds, the fluid will approach equilibrium (although it might take a very long time when there is a high barrier to the equilibrium structure as discussed in Sec 1.8). It would therefore be wise to delay accepting sampling configurations for the energy average etc until the system has approached equilibrium. That is possible in MonteLab by using a so-called Meltdown period. During this period, the random motion of the particles proceeds without sampling of averages. Use of a meltdown period is also possible when the initial configuration is read from an input file. The part of the simulations when sampling is done is called the *real simulation*.

### 1.10. Steplength adjustment

The steplength, i.e. the maximal distance a particle can be displaced in one move, might be selected to be fixed or be subject to adjustment. The adjustment algorithm will adjust the steplength to give a preset rejection ratio in the equilibrium fluid. The adjustment takes place during the entire meltdown period but after the meltdown the steplength will be fixed to the estimated optimal one. The algorithm has given the desired ratio within a few percentage units in all the simulations that we have executed. The algorithm is designed so that the steplength gradually approaches the desired one simultaneously as the structure approaches equilibrium.

THE MONTELAB STEPLENGTH ADJUSTMENT ALGORITHM

The simulation length, i.e. the number of generated configurations, during the meltdown is $W_{melt}$. This number must be divisible by ten. The length of each of the ten adjustment periods is $W_{adj} = W_{melt} / 10$. The desired rejection ratio is $q_d$, $5\% \le q_d \le 95\%$.

1. The initial steplength, $S_1$, is given by the user. A good initial steplength might be 10% to 50% of the lattice constant in the (corresponding) simple cubic lattice. The initial steplength must be less than 90 % of the side length of the active cube, *L*.

2. The rejection ratio of the first adjustment period, $q_1 = \text{number of rejected configurations} / W_{adj}$, is calculated. Calculate the steplength in the second adjustment period by

   $$S_2 = \min\left[\min\left[\frac{q_d}{q_1};2\right]\cdot S_1;90\%\cdot L\right]$$

   If $S_2$ happen to equal $S_1$, we set $S_2 = 0.99S_1$.

3. The rejection ratio of the second adjustment period, $q_2 = \text{number of rejected configurations during the second adjustment period} / W_{adj}$, is calculated.

   a. If $q_1$ and $q_2$ are on different sides of $q_d$, $S_3$ is calculated by linear interpolation,

   $$S_3 = S_1 + \left(S_2 - S_1\right)\frac{q_d - q_1}{q_2 - q_1}$$

   b. If $q_1$ and $q_2$ are on the same side of $q_d$, $S_3$ is extrapolated. However, in order to avoid too big changes in steplength, the change is maximized to $2\left|S_2 - S_1\right|$.

   $$\begin{cases} S_3 = \min\left[S_1 + \min\left[\left(S_2 - S_1\right)\dfrac{q_d - q_1}{q_2 - q_1};2\left|S_2 - S_1\right|\right];90\%\cdot L\right] & \text{if } q_1 < q_d \\ S_3 = S_1 - \min\left[-\left(S_2 - S_1\right)\dfrac{q_d - q_1}{q_2 - q_1};2\left|S_2 - S_1\right|\right] & \text{if } q_1 > q_d \end{cases}$$

   In the last case there is a risk that $S_3$ will be less than zero. In that case we set

$$S_3 = S_2 \frac{q_d}{q_2}$$

If $S_3$ happen to equal $S_2$, we set $S_3 = 0.99 S_2$.

4. The steplengths $S_4, \ldots, S_{10}$ are calculated analogously.

5. In the beginning of the meltdown period, the rejection rates were not related to the equilibrium structure. Therefore only the six last pairs $\left(S_i, q_i\right)$ are used for a final estimation of the optimal steplength. The estimated optimal steplength is a weighted average

$$S_{opt} = \frac{\sum_{i=5}^{10} \exp\left(-30\left|q_d - q_i\right|\right) \cdot S_i}{\sum_{i=5}^{10} \exp\left(-30\left|q_d - q_i\right|\right)}$$

## 1.11. Random number generator

A random number generator should not just generate unpredictable random numbers according to a certain distribution. As a computer can handle only finite integers, the sequence of random numbers will eventually repeat itself. A good random number generator has a long such period and for this random number generator the period is $2^{31} - 2$. Furthermore, the generated random number should not be correlated to the preceding ones. This random number generator has a slight low-order correlation but should work for most applications. The random number generator could easily be replaced if one so desires. This random number generator is taken from Numerical Recipes [4].

## 1.12. Tail correction for the potential cut-off

Since the potential and virial energies (and thus also the pressure) are calculated only to $r_{\text{cut-off}}$, an approximate correction for the missing tail should be added. The tail is divided into two parts.

1. The interval $r_{\text{cut-off}} < r < L/2$ where the sampled $g\left(r\right)$ function can be used.
2. The interval $L/2 < r < \infty$ where we approximate $g\left(r\right) \equiv 1$.

The tail corrections are then

$$\begin{aligned}
\langle U\rangle_{\text{incl. tail}} &= \langle U\rangle_{\text{excl. tail}} &&+\frac{4\pi N\rho}{2}\left(\int_{r_{\text{cut-off}}}^{L/2}\theta(r)g(r)r^2dr+\int_{L/2}^{\infty}\theta(r)r^2dr\right)\\
\langle E_{vir}\rangle_{\text{incl. tail}} &= \langle E_{vir}\rangle_{\text{excl. tail}} &&+\frac{4\pi N\rho}{2}\left(\int_{r_{\text{cut-off}}}^{L/2}\varphi(r)g(r)r^2dr+\int_{L/2}^{\infty}\varphi(r)r^2dr\right)\\
\langle P\rangle_{\text{incl. tail}} &= \langle P\rangle_{\text{excl. tail}} &&+\frac{4\pi N\rho}{2V}\left(\int_{r_{\text{cut-off}}}^{L/2}\varphi(r)g(r)r^2dr+\int_{L/2}^{\infty}\varphi(r)r^2dr\right)
\end{aligned}$$

Lennard-Jones

$$\int_{L/2}^{\infty}\theta(r)r^2dr=\frac{2048\varepsilon\sigma^{12}}{9L^9}-\frac{32\varepsilon\sigma^6}{3L^3}$$

$$\int_{L/2}^{\infty}\varphi(r)r^2dr=\frac{8192\varepsilon\sigma^{12}}{9L^9}-\frac{64\varepsilon\sigma^6}{3L^3}$$

Morse

$$\begin{aligned}
\int_{L/2}^{\infty}\theta(r)r^2dr=\frac{D_e e^{\alpha(r_e-L)}}{8\alpha^3}\Big\{&(\alpha L)^2\left(e^{\alpha r_e}-4e^{\alpha L/2}\right)+\\
&2\alpha L\left(e^{\alpha r_e}-8e^{\alpha L/2}\right)+\\
&2\left(e^{\alpha r_e}-16e^{\alpha L/2}\right)
\end{aligned}$$

$$\begin{aligned}
\int_{L/2}^{\infty}\varphi(r)r^2dr=\frac{D_e e^{\alpha(r_e-L)}}{24\alpha^3}\Big\{&(\alpha L)^3\left(e^{\alpha r_e}-2e^{\alpha L/2}\right)+\\
&3(\alpha L)^2\left(e^{\alpha r_e}-4e^{\alpha L/2}\right)+\\
&6\alpha L\left(e^{\alpha r_e}-8e^{\alpha L/2}\right)+\\
&6\left(e^{\alpha r_e}-16e^{\alpha L/2}\right)
\end{aligned}$$

How reasonable the assumption of a constant $g(r)$ beyond $L/2$ is, might be tested with the $Q_g$ test (See sec. 1.13).

### 1.13. The $Q_g$ test

There are two tests in MonteLab for finding inhomogenities in the system. The first one is the $Q_g$ test ("Quality of $g(r)$ tail"), which is based on all the configurations during the real simulation. The $Q_g$ test tests whether the assumption that $g(r)\equiv 1$ when $r>L/2$ is acceptable. If it is, then the average particle density for $r<L/2$ must be equal to the global one (minus the central atom which is not a neighbour). This can be written as

$$\int_0^{L/2} \rho g(r)\cdot 4\pi r^2 dr = \text{number of neighbours in the sphere} = \rho\frac{4\pi(L/2)^3}{3} - 1$$

The $Q_g$ number is defined by

$$Q_q = \frac{\int_0^{L/2} \rho g(r)\cdot 4\pi r^2 dr}{\rho\frac{4\pi(L/2)^3}{3} - 1}$$

which should be unity if $g(r)\equiv 1$ when $r > L/2$. If $Q_g$ deviates greatly from unity we can suspect that there are inhomogenities in the system of sizes that are at least comparable to $L$. Furthermore, the tail compensations are not reliable. The first measure to take in order to improve the $Q_g$ number is to increase the number of active particles.

## 1.14. The Bubble search method

By looking at the two figures in Figs. 1.6a-b, it is easy to find inhomogenities with the naked eye. In a three-dimensional plot it is however more difficult. One way to find inhomogenities in a configuration would be to register the maximal empty sphere that could be placed in the fluid. Unfortunately, a lonely gas atom placed in the middle of an empty space would drastically reduce the estimation of empty region size.

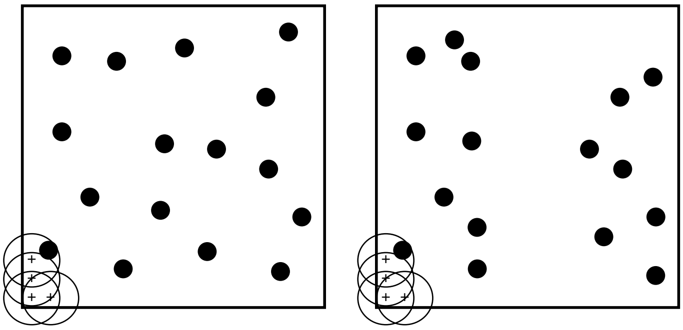


Fig. 1.6. Spheres are placed in the active cube. To the right is an inhomogeneous system.

The method that we have developed for MonteLab, instead searches for large regions with high density and large regions with low density. The method is as follows:

1. Spread spheres in a cubic lattice in the active cube. There are at least 64 times as many spheres as there are particles in the active cube. All spheres have the same radius and the radius is at least so large that all points in the cube are covered.
2. Make a histogram over the number of particles found in each sphere. The abscissa is normalized to

$$\rho_n = \frac{\text{number of particles in the sphere}}{4\pi r_{sphere}^3 / 3}.$$

3. Keep the number of spheres constant but change the radius of the spheres and draw new histograms.

This method will only test a single configuration, in MonteLab it is the final configuration that undergoes the Bubble search, and in contrast to the $Q_g$ test, it does not say anything about all the configurations during the simulation.

As an illustration of the method, let us look at a case that is clearly a two-phase system. The number of particles is 729 and the input file might be found on MonteLab's webpage (Input.Inhomogeneous). This simulation is compared with a homogeneous system (Input.Homogeneous on the webpage). The final configurations are in Figs 1.7a-b and some histograms are found in Fig.1.8.

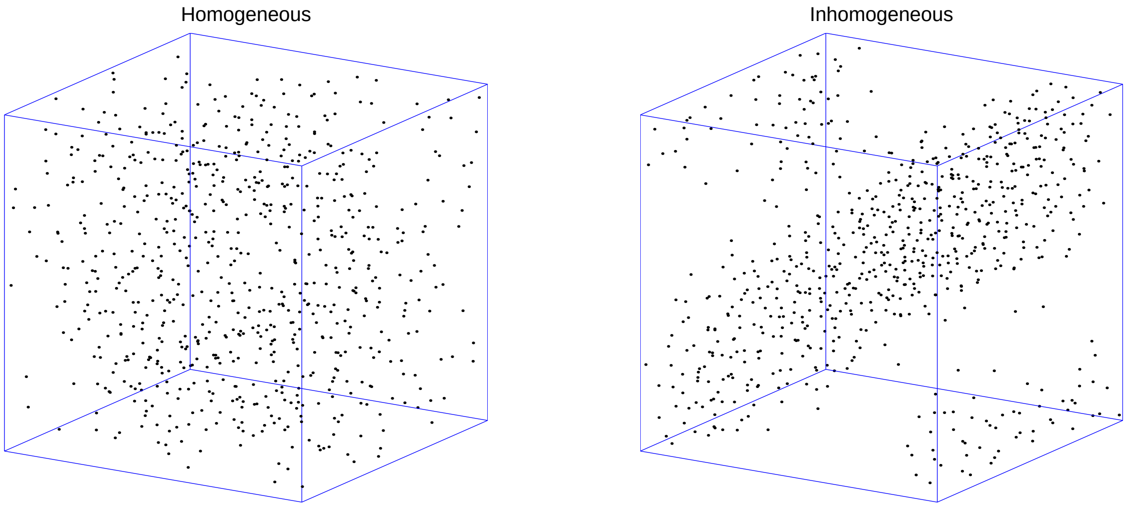


Fig. 1.7. Snapshots of the configuration of a homogeneous and an inhomogeneous fluid.

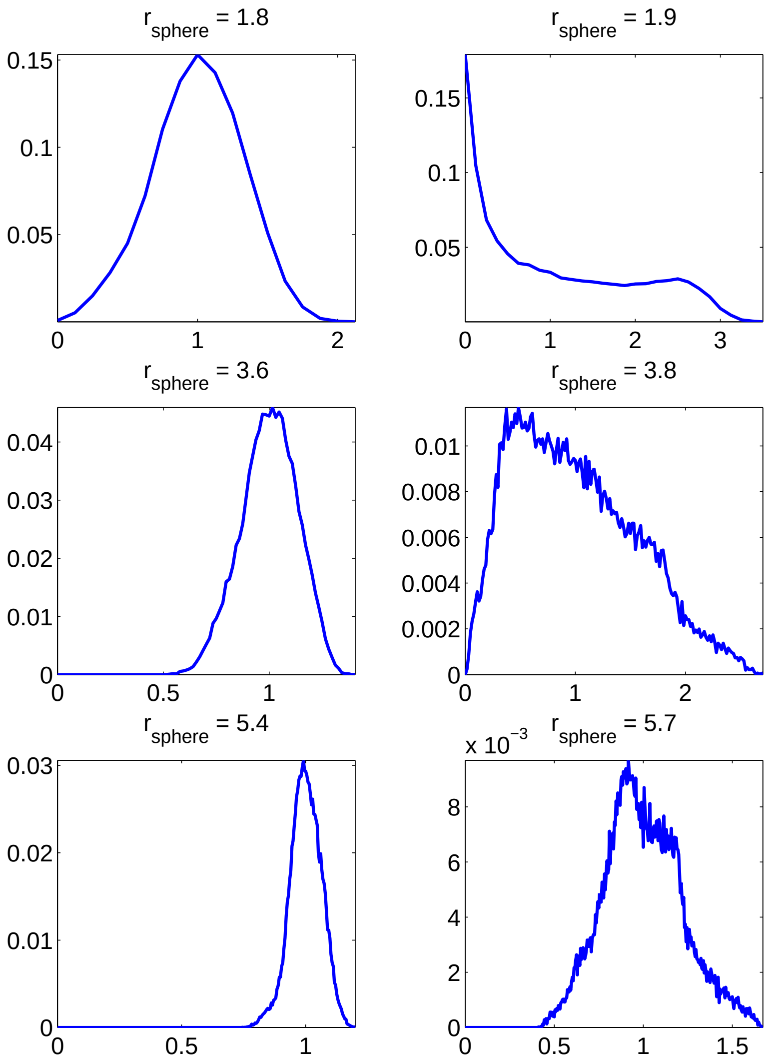


Fig. 1.8. Histograms over normalized density for different sphere radii. The left figures are for the homogeneous fluid in Fig 1.7a. and the right ones for the inhomogeneous fluid in Fig 1.7b.

For the homogeneous system, the distribution is symmetric around unity and the variance decreases with sphere radius. For the inhomogeneous system with a sphere radius of 1.9 reduced units, many spheres are almost empty, and the distribution is not at all symmetric. Not until radii larger than 6 does the distribution become relatively symmetric. We can thus conclude that the size of inhomogeneities is about 4-6 reduced units. The $Q_g$ numbers for the simulations are 1.005 for the homogeneous fluid and 1.01 for the inhomogeneous one.

## 1.15. Restarting a simulation

In some cases it might be desirable to restart a simulation from a point somewhere within the simulation. It could be that a time-consuming simulation has been interrupted and that you want to continue it as if nothing has happened. It could also be that you want to study a part of a very long simulation in more detail, i.e. data is written to files more often for this specific part. It is in MonteLab version 6 possible to restart a simulation using the same set of random numbers.

The procedure to start at a position in an old simulation so that the new simulation continues identically is as follows.

Copy the coordinates at the desired position (Monte Carlo step) in the *Conf file* of the old simulation to a new configuration input file. Find the corresponding Monte Carlo step in the *Data file*. Use the random number at that position as the seed random number in the new simulation. Look in the Data file what steplength that was used at that Monte Carlo step. Use this steplength in the new simulation. This procedure must be taken care of manually by the user.

Note that the averages might be a little different in the old and the new simulations. Say that configuration number 10000 in the old simulation is the initial one in the new one. The next local average in the old one might be over configurations 10001 to 10100 but the first local average in the new simulation is over configurations 1 to 100 which correspond to 10000 to 10099 in the old simulation.

Note that since that heat capacity is a variance, the total heat capacity of two simulations is not simply the average of the two heat capacities.

By one *Monte Carlo step* we mean one trial move of a particle. When comparing simulation lengths for systems with different number of particles, it is rather the number of *Monte Carlo steps per particle* that should be used.

### 1.16. Online help

```
Do you want some help and information ( y / n ) ?
```

*A short description to MonteLab may be given. If zero is given to any input, a short help is given and once again MonteLab asks for input.*

### 1.17. Number of particles

```
Enter Number of Particles ( <8001 )
```

*The cubic numbers up to 8000 are 8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728, 2197, 2744, 3375, 4096, 4913, 5832, 6859 and 8000.*

### 1.18. Particle density

```
Enter the Particle Density
```

*The particle density, i.e. particles per volume, should be given in reduced units. The critical density for the reduced Lennard-Jones fluid is approximately 0.3.*

### 1.19. Temperature

```
Enter kb*T. (kb=The Boltzmann constant)
```

$k_bT$ *should be given in reduced energy units. The critical temperature for the reduced Lennard-Jones is approximately 1.3.*

### 1.20. Kind of potential

```
Choose potential function for the interaction between
the particles
```

*The options are Lennard-Jones or Morse pair-potentials.*

```
Enter the Morse parameter alpha.
```

*The alpha parameter should be given in Morse reduced inverse length units.*

### 1.21. Cut-off distance

```
Enter Cut-off Distance for the potential
```

*The cut-off distance must be smaller than* $L/2$. *The reason is to avoid interaction between a particle and many images of another particle. The cut-off should be given in reduced length units. It is recommended to use a cut-off of at least 4.*

### 1.22. Initial configuration

```
Enter 'G' for 'generated' or 'F' for 'file'
```

*Decide whether the initial configuration should be generated as a simple cubic crystal by MonteLab or fetched from an input file.*

```
From which file should the configuration be fetched.
```

*A configuration input file should end with N lines of particle coordinates. The first column should be numbered 1 to N. The second to fourth columns should give the x, y and z coordinates of the particles, respectively. If anything goes wrong with the reading of the file, the simulation will be interrupted. Furthermore, all the coordinates must be within the interval* $[0, L]$. *The output configuration file of a simulation is written so that it might be used in another simulation as an input file.*

### 1.23. Number of sampling intervals for the radial distribution function

```
Into how many g(r) Sampling Intervals do you want to divide
the r axis?
```

*The* $g(r)$ *function will be sampled in equally long intervals in* $r = 0$ *to* $r = L/2$. *The maximal number of intervals is 10000.*

### 1.24. Open output files

```
Enter a Name of this Simulation.
 (Maximum thirty letters and digits.)
```

*The name of the simulation will be given as suffix to all the output files.*

```
Enter Date (or just press enter).
(Maximum ten letters and digits.)
```

*Optional input of date that will be registered in all the output files.*

```
Do you want to perform bubble search (Y/N) ?
```

*Enter your choice.*

### 1.25. Use of meltdown

```
Do you want to use a 'Meltdown'? ( Y/N )
Data sampled during the meltdown period will not be
included in the final averages.
```

*Enter your choice.*

### 1.26. Use of steplength adjustment

```
Choose between fixed or adjusting Steplength.
Enter 'F' for 'fixed', 'A' for 'adjusting'.
```

*Enter your choice. This question will only be asked if meltdown was chosen in the previous question.*

```
Enter the rejection ratio you want in percentage.
(Within [5%,95%].)
```

*The steplength will be adjusted to give a rejection rate close to the one entered here. If you do not know of anything better, enter 50. The percentage should be given as an integer.*

### 1.27. Steplength

```
Enter (fixed/initial) Steplength for the Simulation.
```

*If the steplength is chosen to be adjusted, this steplength is the one that is used during the first tenth of the meltdown simulation. The steplength must be less than 0.9L.*

### 1.28. Seed random number

```
Enter a positive integer seed random number.
```

*The seed random number to the random number generator should be any positive integer.*

### 1.29. Setup of the meltdown simulation

```
How many Monte Carlo Steps do you want to perform between
every Evaluation of the Pressure, etc.?'
```

*Enter any number.*

`Enter Number of Monte Carlo Steps during the meltdown.`

*The meltdown is divided into ten parts and there must be at least one printing of the pressure in each part. Therefore the total number of Monte Carlo steps must be a multiple of ten times the number of steps between every pressure evaluation.*

`How many Monte Carlo Steps do you want to perform between every Storing of the g(r) function?`

*Output of the g(r) function can be done at maximally every pressure printing. This number must therefore be a multiple of the number of steps between every pressure evaluation. If you do not want any g(r) output, enter a number greater than the total number of meltdown steps.*

`How many Monte Carlo Steps do you want to perform between every Writing of particle coordinates?`

*Output of the particle coordinates can be done at maximally every pressure printing. This number must therefore be a multiple of the number of steps between every pressure evaluation. If you do not want any output of the coordinates, enter a number greater than the total number of meltdown steps.*

### 1.30. Setup of the real simulation

*The same questions about number of steps are asked for the real simulation.*

`How many Monte Carlo Steps do you want to perform between every Evaluation of the Pressure, etc.?`

`Enter Number of Monte Carlo Steps during the real simulation.`

`How many Monte Carlo Steps do you want to perform between every Storing of the g(r) function?`

`How many Monte Carlo Steps do you want to perform between every Writing of particle coordinates?`

## USER INSTRUCTIONS TO MONTELAB - OUTPUT

### 1.31. The Conf file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date and then number of active particles are stated. Then for the desired Monte Carlo steps, the coordinates of the configurations are written. The first column gives the number of the particle, the second to the fourth give the *x*, *y* and *z* coordinates. If a meltdown simulation is used, the end of that one is indicated. The file is ended with the final configuration of the simulation. The configuration file can be used as a configuration input file to MonteLab.

## 1.32. The g.r file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date are stated. Then for the desired Monte Carlo steps, the $g(r)$ function is written. The $g(r)$ function is sampled in equal intervals from $r=0$ to $r=L/2$. In this file the first column is the middle point in each interval *I*, the second interval is the $g_I$ numbers, defined by

$$g_I \int_{\text{interval } I} 4\pi r^2 dr = \int_{\text{interval } I} g(r) 4\pi r^2 dr.$$

The third column is the number of the interval, *I*. If a meltdown simulation is used, the end of that one is indicated. All printings of $g(r)$ are averages from the start of the simulation (excluding the meltdown). The $g(r)$ function is always printed after the simulation is completed.

## 1.33. The Data file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date and potential type are stated. By a local average we mean the average of a quantity since the last Monte Carlo step for which the quantity was written to the file. With the total average we mean the average since the start of the simulation. Note that the total average is reset after the meltdown.

If (tail) is added to a quantity it includes tail correction. The quantities are for the whole system and not per particle.
Local averages are then given for

1. The potential energy, the squared potential energy and the potential energy including tail correction (tail)
2. Virial energy, squared virial energy
3. Virial energy (tail) and the cubic virial energy
4. Pressure based on the virial energy and the pressure based on the virial energy (tail)
5. The PV1 (tail), PV2 (tail) and the $\text{virial energy}\times\text{PV1}$ energies. These energies correspond to $\langle\{\Psi\}\rangle$, $\langle\{\xi\}\rangle$ and $\langle\{\varphi\}\{\Psi\}\rangle$, respectively.

The heat capacity divided by the Boltzmann constant is a total variance from the start of the simulation.

The last used random number is also given to enable restart of simulation (Sec. 1.15.). Whenever the steplength is changed that is indicated in the file. If a meltdown simulation is used, the end of that one is indicated.

This file ends with a list of the input data and the final results. The steplengths and corresponding ratios of rejection are listed. The steplength that was used in the real simulation is displayed as the *final steplength*. The number of rejected trial moves in the real simulation is given so that a check of the rejection ratio can be calculated. Total averages and squared total averages are given so that error estimates can be done. Some quantities are given for the final configuration. The $Q_g$ number of Sec. 1.13. is given.

### 1.34. Standard output

In the standard output the total averages of the potential energy and the pressure are given. The name of the simulation and the line "`THE SIMULATION WAS PROPERLY FINISHED.`" is found at the end of the line when the simulation is over.

### 1.35. The Bubble file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date are stated. The $Q_g$ number of Sec. 1.13. is given. The histograms over normalized density are given. In column one the number of particles found in a sphere is given. In column two, the number of particles is translated into normalized density. The third column is the normalized frequency.

In the end of the file the sphere radius that were used for the histograms and the variance of the histograms are written.

## OBSTACLES AND SUGGESTIONS

The main obstacle with the simulation is the phase segregation in the coexistence region. For such fluids the MonteLab program with cell constraints in chapter 2 might be used.

We have used 512 particles as standard and a cut-off of 4. For the meltdown we have used 1 million Monte Carlo steps and for the real simulation 10 million steps. By checking if the local average of e.g. the potential energy has converged during the meltdown one can check if the meltdown has been long enough.

## PUBLICATION

**On the role of density fluctuations in the equation of state of a simple fluid**
J. Westergren, S. Nordholm and R. Pendfold, Mol. Sim., **27**, 17 (2001)

## APPENDIX 1.1

### Table of Lennard-Jones parameters

| Particles | $\varepsilon$ | $\sigma$ |
|---|---|---|
| Ne-Ne | 35.7 K $k_b$ | 279 pm |
| Ar-Ar | 124 K $k_b$ | 341.8 pm |
| Kr-Kr | 190 K $k_b$ | 361 pm |
| Xe-Xe | 229 K $k_b$ | 405.5 pm |
| $H_2$- $H_2$ | 33.3 K $k_b$ | 297 pm |
| $N_2$- $N_2$ | 91.5 K $k_b$ | 368.1 pm |
| $O_2$- $O_2$ | 113 K $k_b$ | 343 pm |
| $CO_2$- $CO_2$ | 190 K $k_b$ | 399.6 pm |
| $CH_4$- $CH_4$ | 137 K $k_b$ | 382.2 pm |
| $Cl_2$- $Cl_2$ | 357 K $k_b$ | 412 pm |
| $Br_2$- $Br_2$ | 530 K $k_b$ | 427 pm |

Data from McQuarie: Statistical Mechanics and Atkins: Physical Chemistry.

### Table of Morse parameters

| Particles | $D$ | $r_0$ | $\alpha$ |
|---|---|---|---|
| Ar-Ar | 170 K $k_b$ | 3.71 Å | 1.63 Å$^{-1}$ |
| Kr-Kr | 236 K $k_b$ | 397 Å | 1.56 Å$^{-1}$ |
| Xe-Xe | 332 K $k_b$ | 4.32 Å | 1.38 Å$^{-1}$ |

Data from V. V. Kondratev, Sov.Phys. Solid State, Vol 16 (1974).

## APPENDIX 1.2

### Conversion between Lennard-Jones reduced units and SI units

Let us take argon as an illustrating example. According to Appendix 1.1, the parameters for argon are $\varepsilon = 124 \cdot 1.381 \cdot 10^{-23}$ J and $\sigma = 3.42 \cdot 10^{-10}$ m.

$$1 \text{ energy unit} = 124 \cdot 1.381 \cdot 10^{-23}\ \text{J} = 1.71 \cdot 10^{-21}\ \text{J}$$

$$1 \text{ length unit} = 3.42 \cdot 10^{-10}\ \text{m}$$

$$1 \text{ density unit} = \left(1 \text{ length unit}\right)^{-3} = \left(3.42 \cdot 10^{-10}\ \text{m}\right)^{-3} = 2.50 \cdot 10^{28}\ \text{m}^{-3}$$

$$1 \text{ pressure unit} = 1 \text{ energy unit} \cdot \left(1 \text{ length unit}\right)^{-3} = \frac{124 \cdot 1.381 \cdot 10^{-23}\ \text{J}}{\left(3.42 \cdot 10^{-10}\ \text{m}\right)^{3}} = 42.8\ \text{MPa}$$

The temperature always occurs multiplied by the boltzmann constant.

$$\left(k_\text{b} T\right)_\text{reduced units} = \frac{T_\text{SI} \cdot 1.381 \cdot 10^{-23}\ \text{J/K}}{124 \cdot 1.381 \cdot 10^{-23}\ \text{J}/\left(\text{energy unit}\right)} = \frac{T_\text{SI}}{124\ \text{K}} \text{energy units}$$

The heat capacity is always presented divided by $k_b$ and is independent of units

$$\left(\frac{C_\text{v}}{k_\text{b}}\right)_\text{reduced units} = \left(\frac{C_\text{v}}{k_\text{b}}\right)_\text{SI}$$

### Conversion between Morse reduced units and SI units

$$1 \text{ energy unit} = 170 \cdot 1.381 \cdot 10^{-23}\ \text{J} = 2.35 \cdot 10^{-21}\ \text{J}$$

$$1 \text{ length unit} = 3.71 \cdot 10^{-10}/2^{1/6}\ \text{m=3.31 m}$$

$$1 \text{ density unit} = \left(1 \text{ length unit}\right)^{-3} = \left(3.31 \cdot 10^{-10}\ \text{m}\right)^{-3} = 2.76 \cdot 10^{28}\ \text{m}^{-3}$$

$$1 \text{ pressure unit} = 1 \text{ energy unit} \cdot \left(1 \text{ length unit}\right)^{-3} = \frac{170 \cdot 1.381 \cdot 10^{-23}\ \text{J}}{\left(3.31 \cdot 10^{-10}\right)^{3}} = 64.7\ \text{MPa}$$

The temperature always occurs multiplied by the boltzmann constant.

$$\left(k_\text{b} T\right)_\text{reduced units} = \frac{T_\text{SI} \cdot 1.381 \cdot 10^{-23}\ \text{J/K}}{170 \cdot 1.381 \cdot 10^{-23}\ \text{J}/\left(\text{energy unit}\right)} = \frac{T_\text{SI}}{170\ \text{K}} \text{energy units}$$

The heat capacity is always presented divided by $k_b$ and is independent of units

$$\left(\frac{C_\text{v}}{k_\text{b}}\right)_\text{reduced units} = \left(\frac{C_\text{v}}{k_\text{b}}\right)_\text{SI}$$

# REFERENCES

[1] http://www.hgur.se/activities/projects/financed_projects/k-s/nordholm_sture.htm

[2] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller, J. Chem. Phys., **21**, 1087 (1953)

[3] Understanding Molecular Simulation, D. Frenkel, B. Smit, Academic Press, 1996

[4] Numerical Recipies, Press, Teukolsky, Vetterling and Flannery, Cambridge University Press, 1992

# MonteLab CELLPAIR

# Chapter 2

## SIMPLE FLUIDS WITH CELL CONSTRAINTS WITH LJ OR MORSE POTENTIAL IN THE CANONICAL ENSEMBLE (NVT)

### THEORY

### 2.1. Phase segregation

Within the two-phase region, the fluid should segregate into either droplets in a gas or bubbles in a liquid. Such a fluid is difficult to simulate. Firstly, the number of active particles must be very large in order to form the inhomogenieties. And secondly, if the system is large enough, the timescale for the segregation is very long. Hence it is very difficult to get reliable results within the two-phase region using the normal MonteLab routine.

One way to deal with this problem is to confine the particles into subcells of the active cube. The active cube is symmetrically divided into uniform cubic subcells. The number of particles is the same in each cell. The particles are not allowed to leave the cell but PBC are still used in order to avoid surface effects (See Fig. 2.1). With this arrangement, the inhomogenieties in the system are maximized to the size of a cell. Thus, segregation of the system into droplets/bubbles in gas/liquid is prohibited and the system will converge to equilibrium much faster than a system without cells.

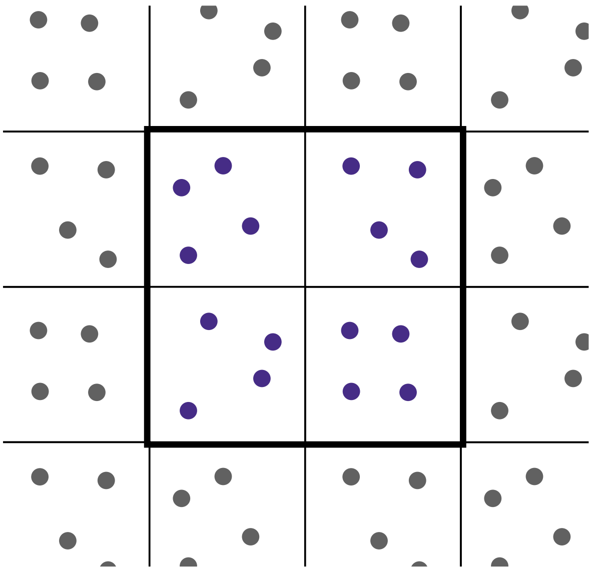


Fig. 2.1. This two-dimensional active cube is divided into four cells with four particles in each cell. The particles are not allowed to leave its cell. Periodic boundary constraints are used.

## 2.2. Equation of state

In the previous section we concluded that a fluid with cell constraints is easier to simulate. However, when comparing the results with experiments, we have not simulated the correct fluid. A diagram that is interesting for many applications is the P-v diagram that illustrates one of the equations of state. From simulations for different temperatures and densities such a diagram may be drawn. What could be the use of the equation of state for an artificial cell fluid? Our hopes have to be set on the generic translations of cell diagrams to proper diagrams that have been shown to exist in some cases. If such translations will be developed, the simulation of a cell fluid is an advantageous way to obtain the proper P-v diagrams.

The isotherms in a P-v diagram for a cell fluid will be quantitatively incorrect due to the prevention of phase segregation. Already in 1873, J.D. van der Waals suggested an equation of state for real gases that shows the principal shape of the isotherms below the critical temperature:

$$P = \frac{k_B T}{v-b} - \frac{a}{v^2},$$

where $v$ is the volume per particle and $a$ and $b$ are parameters. Such an isotherm is drawn in Fig. 2.2. The isotherm has two wiggles due to the prohibited segregation. The unphysical shape may however be taken care of by the so-called *Maxwell reconstruction*. A system that is allowed to segregate into two phases will leave the homogeneous isotherm at a volume $v_l$. At this volume, the pressure will remain constant until all the liquid has been transformed into gas at the volume $v_g$. For larger volumes, the isotherm is again correctly represented by the homogeneous one. Thus, the wiggles in the isotherm should be replaced by a straight line. The pressure of the horizontal line should be set so that the two shaded areas in Fig. 2.2 are equal. Note that wiggles may occur in simulations without cell constraints since larger inhomogenities are prohibited in all fluids with PBC.

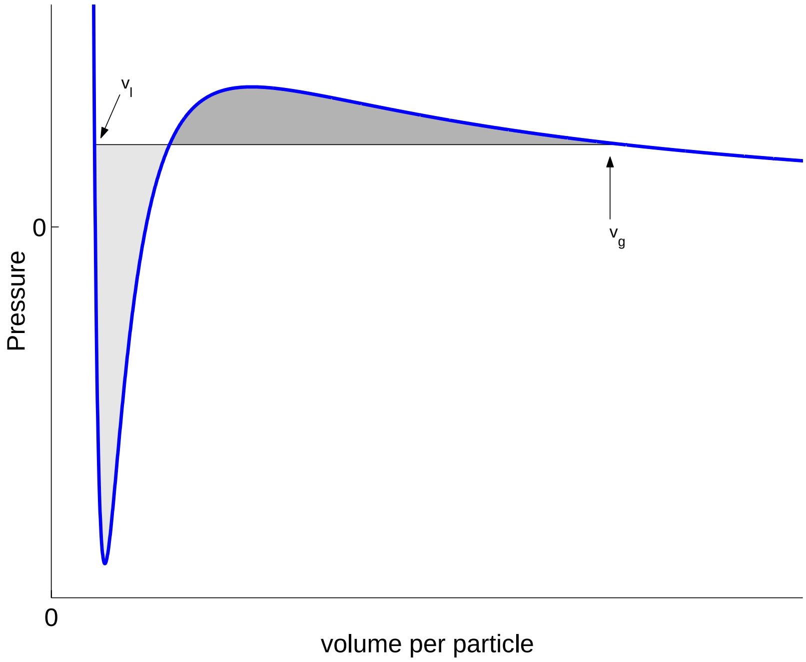


Fig. 2.2. Maxwell reconstruction of an isotherm of a non-segregating fluid.

Despite the use of Maxwell reconstruction, the equations of state for the cell fluid might be shifted and somewhat reshaped compared to the correct one. As mentioned above, future research will hopefully give us generic methods for a correct reshaping.

## 2.3. Calculations of physical quantities

All the physical quantities that were described in chapter one are calculated in the same way for a cell fluid.

## 2.4. Metropolis Monte Carlo with cell constraints

A trial move of an atom out of its original cell is rejected. The Monte Carlo algorithm is

THE MONTELAB CELL CONSTRAINED MONTE CARLO ALGORITHM

1. Start with a configuration $\mathbf{R}_k$ and pick randomly one of the particles in the active cube.
2. Generate a trial configuration by changing the coordinates of the chosen particle: $x_t = x_k + U_1[-\Delta, \Delta]$
   $y_t = y_k + U_2[-\Delta, \Delta]$
   $z_t = z_k + U_3[-\Delta, \Delta]$
3. Calculate $\gamma = \dfrac{\text{Probability}(\mathbf{R}_t)}{\text{Probability}(\mathbf{R}_k)} = e^{-(U(\mathbf{R}_t)-U(\mathbf{R}_k))/k_bT}$
4. If $\gamma > U_4[0,1]$ and the atom is not moved out of its cell, then accept the trial configuration as the new one $\mathbf{R}_{k+1}$, else let $\mathbf{R}_{k+1} = \mathbf{R}_k$.
5. Increase *k* by one a go back to 1.

## 2.5. Initial configuration and meltdown

The initial configuration of the fluid could either come from an input file or be generated by MonteLab. The initial configuration generated by MonteLab is a simple cubic lattice. The particles will be spread to form an infinite crystal with periodic boundary constraints.

When cells are to be used, the number of particles must be an even cubic number. The number of cells that the active cube may be divided into is given in the following table.

| **Number of particles** | **Number of cells** |
|---|---|
| 8 | 1 or 8 |
| 27 | 1 or 27 |
| 64 | 1, 8 or 64 |
| 125 | 1 or 125 |
| 216 | 1, 8, 27 or 216 |
| 343 | 1 or 343 |
| 512 | 1, 8, 64 or 512 |
| 729 | 1, 27 or 729 |
| 1000 | 1, 8, 125 or 1000 |
| 1331 | 1 or 1331 |
| 1728 | 1, 8, 27, 64, 216 or 1728 |
| 2197 | 1 or 2197 |
| 2744 | 1, 8, 343 or 2744 |
| 3375 | 1, 27, 125 or 3375 |
| 4096 | 1, 8, 64, 512 or 4096 |
| 4913 | 1 or 4913 |
| 5832 | 1, 8, 27, 216, 729 or 5832 |
| 6859 | 1 or 6859 |
| 8000 | 1, 8, 64, 125, 1000 or 8000 |

Note that letting the whole active cube be one cell is not the same thing as not using cell constraints at all. In the case of one cell, the particles cannot escape the active cube.

If the initial configuration is read from a file, the coordinates must obey $0 < x, y, z < L = \sqrt[3]{N/\rho}$, like in Sec. 1.9. In addition, the particles must be placed in the different cells correctly. Say that there are $N$ active particles and $M^3$ cells. The first

$n = N/M^3$ particles must be in the first cell, i.e. $0 < x, y, z < L/M$. Particles $n+1$ to $2n$ must satisfy $0 < x, y < L/M, L/M < z < 2L/M$ etc.

### 2.6. Steplength adjustment

Identical with the algorithm in Sec 1.10.

### 2.7. Random number generator

Identical with the generator in Sec 1.11.

### 2.8. Tail correction for the potential cut-off

Identical with the correction in Sec 1.12.

### 2.9. The $Q_g$ test

Identical with the test in Sec 1.13.

### 2.10. The Bubble search method

Identical with the method in Sec 1.14.

### 2.11. Restarting a simulation

As presented in Sec 1.15.

## USER INSTRUCTIONS TO MONTELAB - INPUT

By one *Monte Carlo step* we mean one trial move of a particle. When comparing simulation lengths for systems with different number of particles, it is rather the number of *Monte Carlo steps per particle* that should be used.

### 2.12. Online help

```
Do you want some help and information ( y / n ) ?
```

*A short description to MonteLab may be given. If zero is given to any input, a short help is given and once again MonteLab asks for input.*

### 2.13. Number of particles

```
Enter Number of Particles ( <8001 )
```

*The cubic numbers up to 8000 are 8, 27, 64, 125, 216, 343, 512, 729, 1000, 1331, 1728, 2197, 2744, 3375, 4096, 4913, 5832, 6859 and 8000.*

### 2.14. Particle density

```
Enter the Particle Density
```

*The particle density, i.e. particles per volume, should be given in reduced units. The critical density for the reduced Lennard-Jones is approximately 0.3.*

### 2.15. Temperature

```
Enter kb*T. (kb=The Boltzmann constant)
```

$k_bT$ *should be given in reduced energy units. The critical temperature for the reduced Lennard-Jones is approximately 1.3.*

### 2.16. Kind of potential

```
Choose potential function for the interaction between
the particles
```

*The options are Lennard-Jones or Morse potentials.*

```
Enter the Morse parameter alpha.
```

*The alpha parameter should be given in Morse reduced inverse length units.*

### 2.17. Cut-off distance

```
Enter Cut-off Distance for the potential
```

*The cut-off distance must be smaller than* $L/2$. *The reason is to avoid interaction between a particle and many images of another particle. The cut-off should be given in reduced length units. It is recommended to use a cut-off of at least 4.*

### 2.18. Initial configuration

```
Enter number of cells
```

*The possible number of cells for a given number of particles are listed in Sec. 2.5.*

```
Enter 'G' for 'generated' or 'F' for 'file'
```

*Decide whether the initial configuration should be generated as a simple cubic crystal by MonteLab or fetched from an input file.*

```
From which file should the configuration be fetched.
```

*A configuration input file should end with N lines of particle coordinates. The first column should be numbered 1 to N. The second to fourth columns should give the x, y and z coordinates of the particles, respectively. If anything goes wrong with the reading of the file, the simulation will be interrupted. Furthermore, all the particles must be in their cells according to Sec. 2.5. The output configuration file of a simulation is written so that it might be used in another simulation as an input file.*

### 2.19. Number of sampling intervals for the radial distribution function

```
Into how many g(r) Sampling Intervals do you want to divide
the r axis?
```

*The* $g(r)$ *function will be sampled in equally long intervals in* $r = 0$ *to* $r = L/2$. *The maximal number of intervals is 10000.*

### 2.20. Open output files

```
Enter a Name of this Simulation.
 (Maximum thirty letters and digits.)
```

*The name of the simulation will be given as suffix to all the output files.*

```
Enter Date (or just press enter).
(Maximum ten letters and digits.)
```

*Optional input of date that will be registered in all the output files.*

```
Do you want to perform bubble search (Y/N) ?
```

*Enter your choice.*

### 2.21. Use of meltdown

```
Do you want to use a 'Meltdown'? ( Y/N )
Data sampled during the meltdown period will not be
included in the final averages.
```

*Enter your choice.*

## 2.22. Use of steplength adjustment

```
Choose between fixed or adjusting Steplength.
Enter 'F' for 'fixed', 'A' for 'adjusting'.
```

*Enter your choice. This question will only be asked if meltdown was chosen in the previous question.*

```
Enter the rejection ratio you want in percentage.
(Within [5%,95%].)
```

*The steplength will be adjusted to give a rejection rate close to the one entered here. If you do not know of anything better, enter 50. The percentage should be given as an integer.*

## 2.23. Steplength

```
Enter (fixed/initial) Steplength for the Simulation.
```

*If the steplength is chosen to be adjusted, this steplength is the one that is used during the first tenth of the meltdown simulation. The steplength must be less than 0.9L.*

## 2.24. Seed random number

```
Enter a positive integer seed random number.
```

*The seed random number to the random number generator should be any positive integer.*

## 2.25. Setup of the meltdown simulation

```
How many Monte Carlo Steps do you want to perform between
every Evaluation of the Pressure, etc.?'
```

*Enter any number.*

```
Enter Number of Monte Carlo Steps during the meltdown.
```

*The meltdown is divided into ten parts and there must be at least one printing of the pressure in each part. Therefore the total number of Monte Carlo steps must be a multiple of ten times the number of steps between every pressure evaluation.*

```
How many Monte Carlo Steps do you want to perform between
every Storing of the g(r) function?
```

*Output of the g(r) function can be done at maximally every pressure printing. This number must therefore be a multiple of the number of steps between every pressure evaluation. If you do not want any g(r) output, enter a number greater than the total number of meltdown steps.*

```
How many Monte Carlo Steps do you want to perform between
every Writing of particle coordinates?
```

*Output of the particle coordinates can be done at maximally every pressure printing. This number must therefore be a multiple of the number of steps between every pressure evaluation. If you do not want any output of the coordinates, enter a number greater than the total number of meltdown steps.*

### 2.26. Setup of the real simulation

*The same questions about number of steps are asked for the real simulation.*

```
How many Monte Carlo Steps do you want to perform between
every Evaluation of the Pressure, etc.?
```

```
Enter   Number   of   Monte   Carlo   Steps   during   the   real
simulation.
```

```
How many Monte Carlo Steps do you want to perform between
every Storing of the g(r) function?
```

```
How many Monte Carlo Steps do you want to perform between
every Writing of particle coordinates?
```

## User instructions to MonteLab - Output

### 2.27. The Conf file

The same as in Sec. 1.31 except that the number of cells is written too.

### 2.28. The g.r file

The same as in Sec. 1.32.

### 2.29. The Data file

The same as in Sec. 1.33 except that the number of cells is written too.

### 2.30. Standard output

The same as in Sec. 1.34.

### 2.31. The Bubble file

The same as in Sec. 1.35 except that the number of cells is written too.

## OBSTACLES AND SUGGESTIONS

We have used 512 particles as standard and a cut-off of 4. For the meltdown we have used 1 million Monte Carlo steps and for the real simulation 10 million steps. By checking if the local average of e.g. the potential energy has converged during the meltdown one can check if the meltdown has been long enough.

## PUBLICATION

**On the role of density fluctuations in the equation of state of a simple fluid**

J. Westergren, S. Nordholm and R. Pendfold, Mol. Sim., **27**, 17 (2001)

MonteLab MBACLUSTER

Chapter 3

# CLUSTERS OF ATOMS INTERACTING WITH THE MBA (MANY-BODY ALLOY) POTENTIAL. THE CANONICAL ENSEMBLE (NVT)

## THEORY

### 3.1. Finite system

A metal clusters is an aggregate of a few to about a few hundreds or thousands of metal atoms. They are of scientific interest as they have different properties compared with both single atoms and bulk. They do also have properties which may be of technological interest but this kind of nanotechnology is still in its infancy.

In metal clusters with very few atoms, the bonds in the atoms might be directed in similarity with normal covalently bonded molecules. For larger clusters, the bond is of metallic character with dislocated free electrons, and the bonds are less directed. The bond strength is approximately as strong as in molecules. There is however a big difference between molecules and metal clusters. When two molecules encounter each other they normally bounce off unchanged. The metal clusters, however, generally reduce the energy per atom by combining and tend to form larger and larger aggregates. In a collision between two clusters, the excess energy might be too large for the clusters to combine, but the clusters that leave after a collisions are unlikely the same as the ones prior to the collision. The identity of a cluster is much weaker than for a molecule. Thus, experiments are often done on clusters in beams in vacuum or isolated clusters on substrates or cluster surrounded by insulating ligands.

Since the clusters tend to combine we cannot talk about "a liquid of $Pd_{13}$ cluster". Still, the term "melting of $Pd_{13}$ clusters" has a meaning. At low temperatures, a $Pd_{13}$ cluster has a stiff icosahedral structure but at higher temperatures the atoms in the clusters can change positions and the structure might be far from the icosahedral one. Such a cluster we call a molten one. Thus "solid" and "molten" refers to the state of a single cluster. The heat capacity of a cluster has a peak at a temperature between the "solid" and "molten" phases and this we identify as the melting temperature. In contrast to a bulk system, the transition occurs in a temperature interval instead of at a single melting point.

This MonteLab program aims for simulating an ideal gas of single-sized clusters at a specific temperature. Unfortunately such a system does not exist due to the tendency of the clusters to combine. A cluster is unlikely to leave a wall that it collides with. Furthermore, collisions between clusters will rapidly destroy the size uniformity. It can still be of interest to know the thermal properties of a specific size for instance in the short run of a cluster beam. Hence we can think of an ensemble of single-sized cluster that exist for a short while.

As collisions between clusters ruins the uniformity of size, we consider the clusters to form a gas of very low pressure, in which the cluster can be considered to exist independently of each other. Hence there is no need to simulate more than one cluster. Now the second fundamental difficulty arises. The probability for a cluster alone in the universe to fragment is unity for all temperatures above zero kelvin. This is demonstrated by a dimer alone in universe in Fig. 3.1. Say that the two atoms interact with a Lennard-Jones potential. The probability for having the bond distance is

$$\mathrm{Prob}\left(r\right) \propto e^{-U(r)/k_bT} \cdot 4\pi r^2 dr.$$

The probability has a local maximum near the zero-kelvin equilibrium but as the universe is so large, the small (vanishing) probability for a dimer configuration drowns in the whole universe of fragmented configurations.

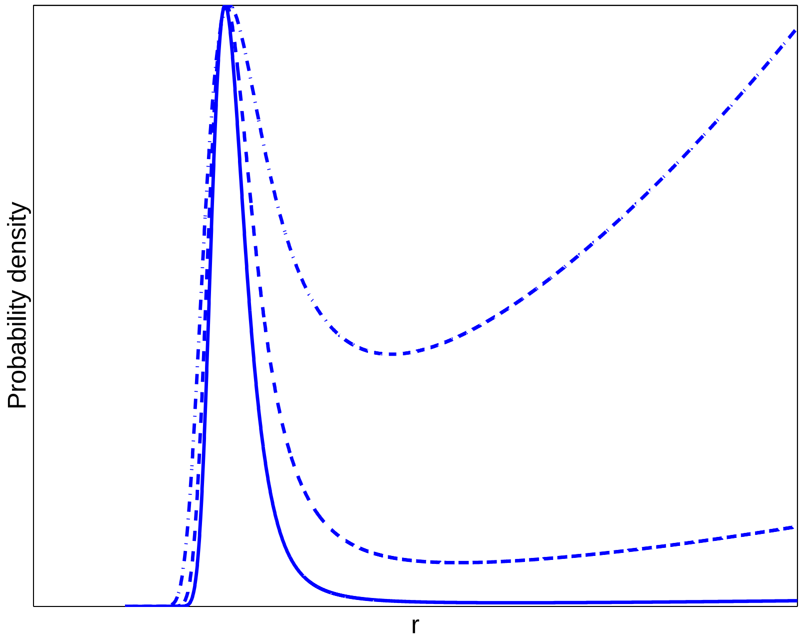


Fig. 3.1. The probability density for a LJ dimer at different temperatures. The temperature is low (solid), medium (dashed) and high (dashed-dotted).

The reason that not a normal gas of molecules fragment for all temperatures is that there is a certain pressure and thus a certain number of molecules per unit volume

The solution to this problem is to restrain the fragmentation in some way. One way is to confine the cluster in a sphere of a certain volume. Any trial move of an atom out of the sphere is regarded as a fragmentation attempt and is rejected. The choice of sphere radius must be done by the user and thus the outcome of the simulation is dependent on this choice. However, for moderate temperatures, the probability distribution in Fig. 3.1 has a distinct minimum. Setting a sphere diameter equal to that minimum would probably give properties that are close to the ones in a gas. Furthermore, if the simulation outcome does not change much for different choices of radii, the outcome is reliable. In MonteLab, however, another definition of a fragmented configuration has been used.

### 3.2. Fragmentation definition

The requirement that the atoms should stay within a fixed sphere may prohibit normal unfragmented non-spherical configurations and make the cluster more spherical than it should be. In order to avoid this, another definition of fragmented/non-fragmented configurations are used in MonteLab.

We say that two atoms are bonded simply if the distance between them are less than a *bond limit*, *B*. Then we say that *a cluster has fragmented if it is not possible to go from any atom to any other atom via bonds*. With this definition the chain of atoms in Fig. 3.2a is a cluster but the atoms in Fig. 3.2b is a fragmented cluster.

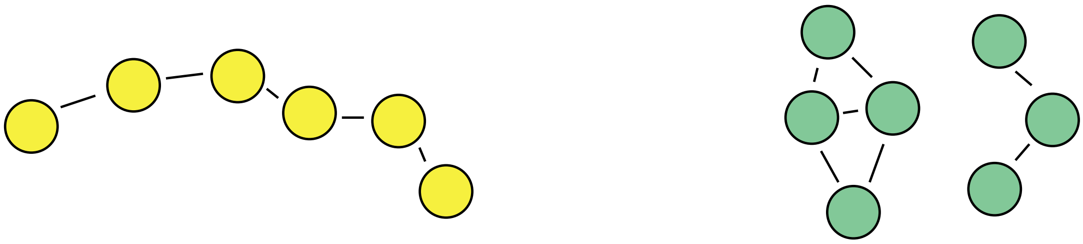


Fig. 3.2. Bonds between the atoms are shown by strokes. The atoms to the left form a cluster but the atoms to the right do not.

The method to test whether a collection of atoms is a cluster or not is taken from Graph Theory [7]. Let us define the *adjacent matrix* **A** by

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } i = j \\ 1 & \text{if atoms } i \text{ and } j \text{ are bonded} \\ 0 & \text{if atoms } i \text{ and } j \text{ are not bonded} \end{cases}$$

Then calculate the *connection matrix* $\mathbf{P} = (\mathbf{I} + \mathbf{A})^{N-1}$ where **I** is the identity matrix and $N$ is the number of atoms in the cluster. The matrix multiplication should be according to Boolean algebra, i.e. as usual algebra with the exception that $1+1=1$. If $\mathbf{P}_{ij} = 1$, it is possible to reach atom *j* from atom *i* via bonds but if $\mathbf{P}_{ij} = 0$, it is not possible. Thus the criterion that the cluster has not fragmented is that all entries of **P** are ones.

The user-defined parameter is the bond length limit *B*. If the outcome varies significantly with *B*, the simulations are useless. However, in simulations of $\mathrm{Pd}_{13}$, the results are almost identical for $4.0\ \text{Å} < B < 5.8\ \text{Å}$. Furthermore, at temperatures well above the melting temperature, we have seen that trial moves are very seldom rejected because of fragmentation. Hence we believe that the simulated configurations do represent a low-pressure gas of uniform palladium clusters.

### 3.3. The canonical ensemble and heat capacities

The probability for a configuration is zero if the cluster has fragmented or proportional to the Boltzmann factor $\exp\left(-U\left(\mathbf{R}\right)/k_bT\right)$. For a specific cluster it is not the temperature (kinetic energy) that is constant but rather the total energy. Thus the set of generated configurations does not represent the realistic configurations for a specific cluster but for an ensemble of clusters, exactly as we wish. In the canonical ensemble the volume should be constant, besides the number of particles and the temperature. Is the volume fixed with our definition of a non-fragmented cluster? Since the maximal length of a non-fragmented cluster is a chain of atoms with the length $\left(N-1\right)B$, the cluster will always fit in a cube with the side length $L = \left(N-1\right)B$. Thus the volume can be regarded as fixed and requirements for a canonical ensemble are fulfilled.

One of the main advantages with the canonical ensemble is that the heat capacity is obtainable from a simulation at one temperature. The heat capacity is the fluctuation in energy given by

$$C_V = \left\langle E^2 \right\rangle - \left\langle E \right\rangle^2 = \frac{3N}{2} k_B + \left\langle U^2 \right\rangle - \left\langle U \right\rangle^2 .$$

The heat capacity that we would like to obtain is however rather the one at constant pressure. At very low cluster pressure, the cluster gas can be regarded as ideal and there is a simple relation between the two types of heat capacity:

$$C_P = C_V + k_b.$$

Since $C_V$ is at least equal to $\left(3N-3\right)k_b$, the relative difference between $C_P$ and $C_V$ is very small already for rather small clusters.

### 3.4. Energy calculations

The interaction between the cluster atoms we approximate to be similar as in a bulk. Thus the binding of the atoms is of metal character. For the description of the potential energy we use the Many-Body Alloy potential [5] (which is identical to the Gupta potential in the expression):

$$U(\mathbf{R}) = \sum_{i=1}^{N} U_i$$

$$U_i = -\xi_0 \sqrt{\sum_{j\neq i}^{N} \exp\left(-2q\left(r_{ij}/r_0 - 1\right)\right)} + \varepsilon_0 \sum_{j\neq i}^{N} \exp\left(-p\left(r_{ij}/r_0 - 1\right)\right).$$

Note that the quantity $U_i$ is not identical to the removal energy of atom *i*. The attractive part of the potential is based on the second moment approximation of tight-binding calculations and is of many-body character. The attraction is due to hybridization of the d states into bands. The repulsion is modeled as being of Born-Mayer type. Parameters for different metals are found in Appendix 3.1

With this cluster, PBC are of course not used, and $r_{ij}$ is the true distance between atoms *i* and $j$. Note the difference between this potential and the pair potentials in chapters 1 and 2. For the pair potentials the sum is imagined to be over all bonds and thus a factor 1/2 is put in front of the sum over all particle distances. Here the potential is not in terms of bonds and the parameters are fitted to the expression without the factor 1/2.

### 3.5. Surface energy calculations

The surface energy we define as the sum of the potential energy for all initial surface atoms.

$$U_{surface}(\mathbf{R}) = \sum_{\substack{\text{surface}\\ \text{atoms}}} U_i .$$

A study of the surface energy can give information about a melting of the surface atoms at lower temperatures than the overall melting.

Note that MonteLab does not upgrade what atoms are at the surface as a simulation proceeds. It is instead given by the initiation of the simulation what atoms are to be considered as surface atoms. Thus, this quantity is not appropriate if any surface atom changes position with a core atom during the simulation. It is not necessary that the surface atoms should be initially at the surface. Thus the surface energy can be used more generally as the sum of potential energy of a desired subset of the atoms.

### 3.6. Geometric properties

The following properties giving information of the geometrical structure of the cluster is calculated: the mean distance to the centre of mass, the radii of gyration, the delta function and the $s(r)$ function.

### 3.7. The mean distance to the centre of mass

The mean distance to the centre of mass for a configuration is

$$M(\mathbf{R}) = \frac{1}{N}\sum_{i}^{N}\left|\mathbf{x}_i - \sum_{i}^{N}\mathbf{x}_i / N\right|,$$

where $\mathbf{x}_i$ is the position of atom *i*. The average is then taken over all the configurations

$$\langle M \rangle = \frac{1}{W}\sum_{k=1}^{W} M(\mathbf{R}_k) \qquad \text{as } W \to \infty.$$

This function is a good measure of the size of the cluster. By observing $\langle M \rangle$ as a function of the temperature, the melting can be identified as the increase of the slope of $\langle M \rangle$.

### 3.8. The radii of gyration

The radius of gyration along a certain axis for a body is defined as

$$K_A = \sqrt{\frac{I_A}{m}}$$

where $I_A$ is the corresponding moment of inertia and *m* is the total mass. For a solid ellipsoidal with the semi-axes *a*, *b* and *c*, the radii of gyration along the principal axes are

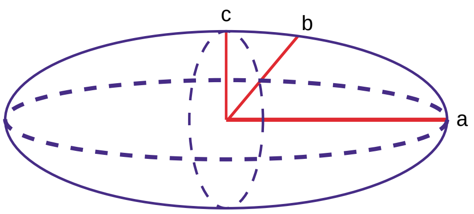
$$K_a = \sqrt{\left(b^2 + c^2\right)/5}$$
$$K_b = \sqrt{\left(a^2 + c^2\right)/5}$$
$$K_c = \sqrt{\left(a^2 + b^2\right)/5}.$$

Thus, the radii of gyration can tell about the shape of the ellipsoidal. For example a prolate (cigar-shaped) ellipsoidal (see Fig. 3.3a) with $a > b = c$ has $K_b = K_c > K_a$. Note that a prolate ellipsoidal has two semi-minor axes and one semi-major one but that two radii of gyration are larger than the third one.

In MonteLab, the radii of gyration are calculated for the cluster in order to give information about shape. The moment of inertia tensor for a particle system is defined as

$$\mathbf{I} = \begin{bmatrix} \sum_{i=1}^{N} m_i \left( y_i^2 + z_i^2 \right) & -\sum_{i=1}^{N} m_i \left( x_i y_i \right) & -\sum_{i=1}^{N} m_i \left( x_i z_i \right) \\ -\sum_{i=1}^{N} m_i \left( x_i y_i \right) & \sum_{i=1}^{N} m_i \left( x_i^2 + z_i^2 \right) & -\sum_{i=1}^{N} m_i \left( y_i z_i \right) \\ -\sum_{i=1}^{N} m_i \left( x_i z_i \right) & -\sum_{i=1}^{N} m_i \left( y_i z_i \right) & \sum_{i=1}^{N} m_i \left( x_i^2 + y_i^2 \right) \end{bmatrix}$$

$m_i$ is the mass of atom *i*. The coordinates are relative the centre of mass. The moments of inertia along the principal axes are the eigenvalues of **I**. As MonteLab presumes that all atoms are identical, the masses cancel when the radii of gyration are calculated. Thus, the radii of gyration of the cluster can give a rough indication of the shape. The clusters will of course not be exactly ellipsoidally shaped. In Fig. 3.3b, a snapshot of a molten $\mathrm{Pd}_{55}$ is shown. As discussed in Sec. 3.2., the shape is far from spherical above the melting temperature.



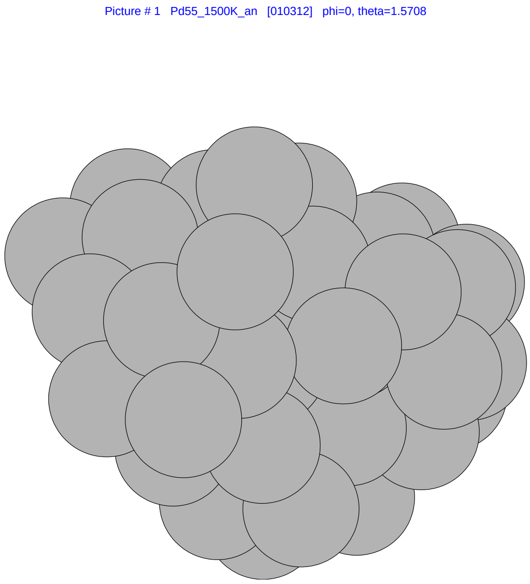


Fig. 3.3a. An ellipsoidal with the semi-axes *a*, *b* and *c*.
Fig. 3.3b. A molten $\mathrm{Pd}_{55}$ cluster.

MonteLab calculates the average radii of gyration as the largest, smallest and medium one.

## 3.9. The $\delta$ function

The $\delta$ function keeps track of the relative variation in interatomic distances. The function is defined as

$$\delta = \frac{2}{N(N-1)} \sum_{i,j>i} \frac{\sqrt{\langle r_{ij}^2 \rangle - \langle r_{ij} \rangle^2}}{\langle r_{ij} \rangle}.$$

This function can give information about how the bond distances change. A much more important role is however that $\delta$ increases significantly if two atoms change positions. Thus, $\delta$ is a good indicator for cluster melting.

### 3.10. The bond length spectrum, *s*(*r*)

The bond length spectrum $s(r)$ gives the spectrum of bond lengths, *r*, in the interval $0 < r < r_{max}$. The function is similar to the radial distribution function $g(r)$ and MonteLab uses the same routine for the calculation. The $s(r)$ function is normalized so that the integral from zero to infinity should be equal to the number of bonds, i.e.

$$\int_0^\infty s(r)\,dr = \frac{N(N-1)}{2}.$$

In Fig. 3.4, the spectrum for a solid $\mathrm{Pd}_{55}$ cluster and one at the melting temperature is shown. Note that for $s(r)$, a bond can be longer than the bond length limit, *B*.

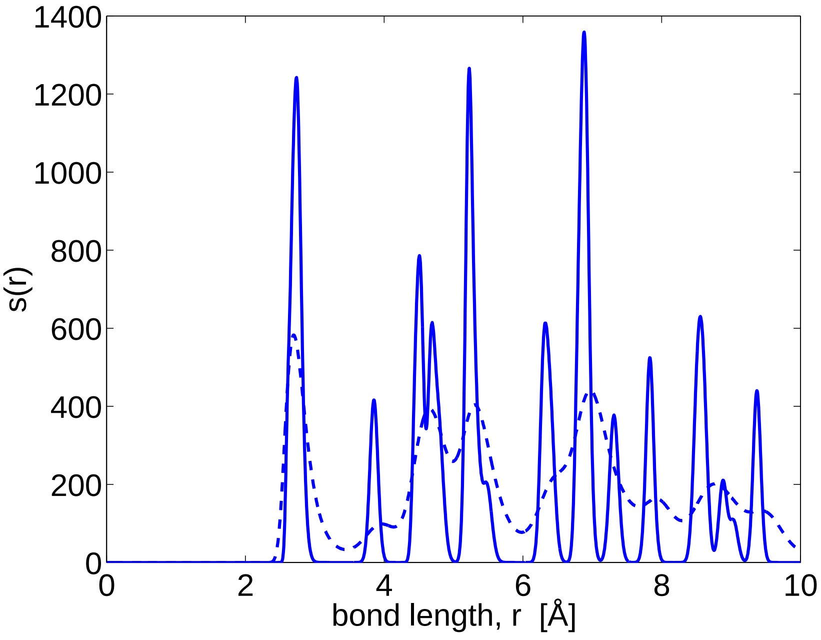


Fig. 3.4. The *s*(*r*) spectrum for a solid $\mathrm{Pd}_{55}$ cluster (solid) and one that is just about to melt (dashed).

## 3.10. Metropolis Monte Carlo with fragmentation prevention

The MonteLab definition of a fragmented cluster is that one or more entries in the connection matrix $\mathbf{P} = (\mathbf{I} + \mathbf{A})^{N-1}$ is zero. It is however not necessary to calculate this matrix after each trial move of an atom since we have advance information (except for the first configuration.) We now that the cluster was not fragmented before the move of the trial atom. Say that we the trial atom is number *k*. The "neighbours" are the atoms that are bonded to *k*. All the other atoms are in some way bonded to the neighbours.

THE MONTELAB CHECK FOR FRAGMENTED CLUSTER

1. Check the neighbours after the trial move.
   If all old neighbours are still neighbours, the cluster has not fragmented.
   If there are no trial neighbours, the cluster has fragmented.
   Else goto 2.

2. Some of the old neighbours are lost after the trial move.
   Check whether the lost neighbours are still bonded to the neighbours that are not lost.
   If they are, the cluster has not fragmented.
   Else goto 3.

3. We give up the shortcuts and calculate $\mathbf{P}_{trial} = (\mathbf{I} + \mathbf{A}_{trial})^{N-1}$.

The Monte Carlo algorithm is

THE MONTELAB CANONICAL CLUSTER MONTE CARLO ALGORITHM

1. Start with a configuration $\mathbf{R}_k$ and pick randomly one of the atoms.
2. Generate a trial configuration by changing the coordinates of the chosen particle: $x_t = x_k + U_1\left[-\Delta, \Delta\right]$

   $y_t = y_k + U_2\left[-\Delta, \Delta\right]$

   $z_t = z_k + U_3\left[-\Delta, \Delta\right]$
3. Calculate $\gamma = \dfrac{\text{Probability}\left(\mathbf{R}_t\right)}{\text{Probability}\left(\mathbf{R}_k\right)} = e^{-\left(U\left(\mathbf{R}_t\right) - U\left(\mathbf{R}_k\right)\right)/k_b T}$
4. If $\gamma < U_4\left[0,1\right]$ reject the trial configuration and let $\mathbf{R}_{k+1} = \mathbf{R}_k$. Else, test whether the trial configuration is a fragmented cluster. If so let $\mathbf{R}_{k+1} = \mathbf{R}_k$. If it is not fragmented, then accept the trial configuration as the new $\mathbf{R}_{k+1}$, else let $\mathbf{R}_{k+1} = \mathbf{R}_k$.
5. Increase *k* by one a go back to 1.

## FEATURES OF MONTELAB

### 3.11. Initial configuration and meltdown

The initial configuration of the cluster could either come from an input file or be generated by MonteLab. The initial configuration generated by MonteLab is an fcc lattice that is filled up as much as possible with the atoms.

Disregarded the initial configuration is read from a file or generated by MonteLab, the coordinates must form a non-fragmented cluster. If they do not, the simulation is interrupted.

### 3.12. Steplength adjustment

Identical with the algorithm in Sec 1.10.

### 3.13. Random number generator

Identical with the generator in Sec 1.11.

### 3.14. Tail correction for the potential cut-off

No correction is done for clusters. The potential cut-off should be large enough to get results that are accurate enough.

### 3.15. Restarting a simulation

As presented in Sec 1.15. Note that the delta function of two simulations is not simply the average of the two results from the two simulations.

## USER INSTRUCTIONS TO MONTELAB - INPUT

By one *Monte Carlo step* we mean one trial move of an atom. When comparing simulation lengths for systems with different number of atoms, it is rather the number of *Monte Carlo steps per atom* that should be used.

### 3.16. Online help

`Do you want some help and information ( y / n ) ?`

*A short description to MonteLab may be given. If zero is given to any input, a short help is given and once again MonteLab asks for input.*

### 3.17. Number of atoms

`Enter Number of Atoms ( <1001 )`

### 3.18. Surface energy

`Do you want to sample the surface energy? ( y / n )`

*The surface energy is defined in Sec. 3.5.*

`Enter the number of atoms that are surface atoms.`

*It is not necessary that the "surface atoms" are at the surface. They should just be a subset of all the atoms.*

`Enter the numbers of the surface atoms`

*The numbers of the atoms are the same numbers that are given before the coordinates in the Conf file. The numbers should be entered one per line.*

### 3.19. Bond limit

`Enter the bond limit [in Å].`

*Two atoms are considered to be bonded if the distance between them is less than the bond limit. The bonds are used for determination of fragmentation.*

### 3.20. s(r) max bond length

`Enter the s(r) max bond length [in Å].`

*The s(r) spectrum is sampled from 0 to the "s(r) max bond length".*

### 3.21. Temperature

`Enter T [in K]`

*The temperature should be given in kelvin.*

### 3.22. The MBA potential parameters

```
Enter the MBA parameter xi [in eV].
```

```
Enter the MBA parameter epsilon [in eV].
```

```
Enter the MBA parameter r0 [in Å].
```

```
Enter the MBA parameter p [no unit].
```

```
Enter the MBA parameter q [no unit].
```

### 3.23. Cut-off distance

```
Enter Cut-off Distance for the potential [in Å].
```

*Since there is no tail correction for the clusters, the cut-off distance should be rather large.*

### 3.24. Initial configuration

```
Enter 'G' for 'generated' or 'F' for 'file'
```

*Decide whether the initial configuration should be generated as an fcc crystal by MonteLab or fetched from an input file.*

```
Enter the lattice constant [in Å].
```

```
From which file should the configuration be fetched.
```

*A configuration input file should end with N lines of particle coordinates. The first column should be numbered 1 to N. The second to fourth columns should give the x, y and z coordinates of the particles, respectively. If anything goes wrong with the reading of the file, the simulation will be interrupted. Furthermore, the cluster must form a non-fragmented cluster according to the definition in Sec. 3.2. The output configuration file of a simulation is written so that it might be used in another simulation as an input file.*

### 3.25. Number of sampling intervals for the *s*(*r*) spectrum

```
Into how many s(r) Sampling Intervals do you want to divide
the r axis?
```

*The* $s(r)$ *function will be sampled in equally long intervals in* $r=0$ *to* $r=\text{"s(r) max bond length"}$. *The maximal number of intervals is 10000.*

### 3.26. Open output files

```
Enter a Name of this Simulation.
(Maximum thirty letters and digits.)
```

*The name of the simulation will be given as suffix to all the output files.*

```
Enter Date (or just press enter).
(Maximum ten letters and digits.)
```

*Optional input of date that will be registered in all the output files.*

### 3.27. Use of meltdown

```
Do you want to use a 'Meltdown'? ( Y/N )
Data sampled during the meltdown period will not be
included in the final averages.
```

*Enter your choice.*

### 3.28. Use of steplength adjustment

```
Choose between fixed or adjusting Steplength.
Enter 'F' for 'fixed', 'A' for 'adjusting'.
```

*Enter your choice. This question will only be asked if meltdown was chosen in the previous question.*

```
Enter the rejection ratio you want in percentage.
(Within [5%,95%].)
```

*The steplength will be adjusted to give a rejection rate close to the one entered here. If you do not know of anything better, enter 50. The percentage should be given as an integer.*

### 3.29. Steplength

```
Enter (fixed/initial) Steplength for the Simulation.
```

*If the steplength is chosen to be adjusted, this steplength is the one that is used during the first tenth of the meltdown simulation.*

### 3.30. Seed random number

```
Enter a positive integer seed random number.
```

*The seed random number to the random number generator should be any positive integer.*

### 3.31. Setup of the meltdown simulation

```
How many Monte Carlo Steps do you want to perform between
every writing of the energy
```

*Enter any number.*

```
Enter Number of Monte Carlo Steps during the meltdown.
```

*The meltdown is divided into ten parts and there must be at least one writing of the energy in each part. Therefore the total number of Monte Carlo steps must be a multiple of ten times the number of steps between every writing of the energy.*

```
How many Monte Carlo Steps do you want to perform between
every Storing of the s(r) function?
```

*Output of the s(r) function can be done at maximally every energy writing. This number must therefore be a multiple of the number of steps between every writing of the energy. If you do not want any s(r) output, enter a number greater than the total number of meltdown steps.*

```
How many Monte Carlo Steps do you want to perform between
every Writing of atomic coordinates?
```

*Output of the atomic coordinates can be done at maximally every energy writing. This number must therefore be a multiple of the number of steps between every writing of the energy. If you do not want any output of the coordinates, enter a number greater than the total number of meltdown steps.*

### 3.32. Setup of the real simulation

*The same questions about number of steps are asked for the real simulation.*

```
How many Monte Carlo Steps do you want to perform between
every writing of the energy
```

```
Enter Number of Monte Carlo Steps during the real
simulation.
```

```
How many Monte Carlo Steps do you want to perform between
every Storing of the s(r) function?
```

```
How many Monte Carlo Steps do you want to perform between
every Writing of atomic coordinates?
```

## USER INSTRUCTIONS TO MONTELAB - OUTPUT

### 3.33. The Conf file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date and then number of atoms are stated. Then for the desired Monte Carlo steps, the coordinates of the configurations are written. The first column gives the number of the particle, the second to the fourth give the *x*, *y* and *z* coordinates. If a meltdown simulation is used, the end of that one is indicated. The file is ended with the final configuration of the simulation. The configuration file can be used as an configuration input file to MonteLab.

### 3.34. The s.r file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date are stated. Then for the desired Monte Carlo steps, the $s(r)$ function is written. The $s(r)$ function is sampled in equal intervals from $r = 0$ to $r = r_{max} =$ "s(r) max bond length". In this file the first column is the middle point in each interval *I*, the second interval is the $s_I$ numbers, defined by

$$s_I \int_{\text{interval } I} dr = \int_{\text{interval } I} s(r)\, dr.$$

The third column is the number of the interval, *I*. If a meltdown simulation is used, the end of that one is indicated. All printings of $s(r)$ are averages from the start of the simulation (excluding the meltdown). The $s(r)$ function is always printed after the simulation is completed.

In versions 4 and 5, the $g(r)$ function was used instead. The relation between the functions is

$$s(r) = \text{constant} \cdot g(r) r^2.$$

### 3.35. The Data file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date and potential type are stated. By a local average we mean the average of a quantity since the last Monte Carlo step for which the quantity was written to the file. With the total average we mean the average since the start of the simulation. Note that the total average is reset after the meltdown.

The quantities are for the whole system and not per particle.
Local averages are then given for

1. The potential energy, the squared potential energy
2. The heat capacity
3. Radii of gyration. The first number is the average of the greatest radii of gyration in all configurations.
4. Mean distance to the centre of mass.
5. $\delta$.
6. If chosen, the surface energy and the squared surface energy.

The heat capacity divided by the Boltzmann constant is the total variance from the start of the simulation.

The last used random number is also given to enable restart of simulation (Sec. 1.15.).
Whenever the steplength is changed that is indicated in the file. If a meltdown simulation is used, the end of that one is indicated.

This file ends with a list of the input data and the final results. The steplengths and corresponding ratios of rejection are listed. The steplength that was used in the real simulation is displayed as the *final steplength*. The number of rejected trial moves in the real simulation is given so that a check of the rejection ratio can be calculated. Especially, the number of trial configurations that were rejected because they would lead to a fragmented cluster is written. Total averages and squared total averages are given so that error estimates can be done. The average interatomic distances $\left\langle r_{ij} \right\rangle$ between atoms $i$ and $j$ as well as $\left\langle r_{ij}^2 \right\rangle$ are given. Some quantities are given for the final configuration.

### 3.36. Standard output

In the standard output the total averages of the potential energy is given. The name of the simulation and the line "THE SIMULATION WAS PROPERLY FINISHED." is found at the end of the line when the simulation is over.

## OBSTACLES AND SUGGESTIONS

One obstacle is that there is a barrier between the solid and the molten phases. Thus, if the simulation is started with a solid structure and the temperature is high enough that the cluster should be molten a considerable fraction of the time, it still might take a very long time until the cluster switches to the molten phase.

Since there is no tail correction for the potential cut-off, the cut-off must be fairly long. For palladium we have used 12 Å. For the clusters we have used between 100 000 and 1 000 000 Monte Carlo steps per atom. The meltdown has been 10% of the real simulation. By checking if the local average of e.g. the potential energy has converged during the meltdown one can check if the meltdown has been long enough.

## PUBLICATION

**Statistical theory of cluster cooling in rare gas – II. The PEMET model**

Jan Westergren, Sture Nordholm and Hongrei Li, Phys. Chem. Chem. Phys., **4**,1815 (2002)

**Cooling efficiency in collisions between Pd13 and He, Ne, Ar and Kr**

Jan Westergren, Sture Nordholm and Arne Rosén, Eur. Phys. J. D, **22**, 81 (2003)

**Melting of palladium clusters –**
**Canonical and microcanonical Monte Carlo simulation**

Jan Westergren, Arne Rosén and Sture Nordholm
Phys. Chem. Chem. Phys., **5**, 136 (2003)

**Simulation of the Melting and Cooling of Palladium Clusters**

Jan Westergren
Doctoral thesis, Chalmers University of Technology and Göteborg University,
Göteborg, Sweden 2001

## APPENDIX 3.1

### Table of MBA parameters of bulk metals

| Element | $\xi_0$ $[\mathrm{eV}]$ | $\varepsilon_0$ $[\mathrm{eV}]$ | $r_0$ $[\text{Å}]$ | $p$ | $q$ | Reference |
|---|---|---|---|---|---|---|
| Pd | 1.2630 | 0.08376 | 2.758 | 14.8 | 3.40 | [6] |
| Ni | 1.18595 | 0.0465616 | 2.49 | 15.0541 | 1.38171 | [7] |
| Pd | 1.70769 | 0.173750 | 2.75 | 10.8874 | 3.75433 | [7] |
| Au | 1.88966 | 0.222356 | 2.88 | 9.95271 | 4.01820 | [7] |
| Ag | 1.16109 | 0.106026 | 2.89 | 10.64078 | 3.21858 | [7] |
| Ni | 1.070 | 0.0376 | 2.52 | 16.999 | 1.189 | [8] |
| Pd | 1.718 | 0.1746 | 2.75 | 10.867 | 3.742 | [8] |
| Pt | 2.695 | 0.2975 | 2.77 | 10.612 | 4.004 | [8] |

## REFERENCES

[5] D. Tománek, S. Mukherjee and K. H. Bennemann, Phys. Rev. B, **28**, 665 (1983); W. Zhong, Y. S. Li and D. Tománek, Phys. Rev. B, **44**, 13053 (1991)

[6] W. Zhong, Y. Cai and D. Tománek, Phys. Rev. B, **46**, 8099 (1992)

[7] C. Rey, L. Gallego, J. García-Rodeja, J. Alonso and M. Iñiguez, Phys. Rev. B, **48**, 8253 (1993)

[8] F. Cleri and V. Rossato, Phys. Rev. B, **48**, 22 (1993)

MonteLab MBAMICRO

# Chapter 4

# CLUSTERS OF ATOMS INTERACTING WITH THE MBA (MANY-BODY ALLOY) POTENTIAL. THE MICROCANONICAL ENSEMBLE (NVE)

## THEORY

### 4.1. The choice of proper ensemble

A cluster in a low-pressure gas seldom encounters any other cluster. The long periods between the encounters, the cluster is moving as an isolated particle in space. Such a particle rather has a constant total energy than a constant temperature. Perhaps, a more realistic simulation would be to simulate the cluster in the microcanonical ensemble. In that ensemble the number of atoms, $N$, the volume, $V$, and the total energy, $E$, are constant. A major advantage with the microcanonical ensemble is that if the total energy is not very high, the cluster cannot fragment. Hence, there is no need for a user-defined parameter as the bond length limit to prevent fragmentation. The disadvantage is that the sampled properties are for a certain total energy and not for a temperature that is the normal controlling parameter in experiments. Furthermore, the heat capacity of the cluster is naturally not available directly from a simulation. The results from a microcanonical ensemble might however be translated to thermal results with a Boltzmann weighting of results at different energies. From simulations of $Pd_{13}$, $Pd_{55}$ and $Pd_{147}$, we can conclude that the two kinds of ensembles give almost identical results.

### 4.2. The microcanonical ensemble

The number of atoms is clearly fixed and as mentioned in Sec. 3.3, the volume of the ensemble is not the very cluster volume, but is rather a fixed volume that the cluster can never escape. A Monte Carlo simulation in the microcanonical ensemble is in principle analogous to a simulation in the canonical ensemble. The atoms are moved randomly one by one in order to generate configurations according to the proper distribution. But what is the distribution?

In the canonical ensemble, the spatial coordinates **R** and the momenta **P** are independent and we only need to simulate the spatial configurations. In the microcanoncial ensemble, each configuration must satisfy the energy conservation

$$E_{kin}\left(\mathbf{P}\right)+U\left(\mathbf{R}\right)=E_{total}.$$

Since the quantities that we are interested in, as for instance the radii of gyration, are functions of the spatial coordinates we would like to simulate only the spatial configurations. What is then the probability for a configuration **R**?

The probability derivation is based on one of the fundamental postulates of Statistical thermodynamics:

> *"All states in an isolated (microcanonical) system are equally probable."*

Thus, all total configurations $\left[\mathbf{R},\mathbf{P}\right]$ are equally likely as long as they satisfy the energy conservation. Considering only spatial configurations, the probability for $\mathbf{R}_k$ must be proportional to the number of different momenta states that are available for the kinetic energy

$$E_{kin}\left(\mathbf{P}_k\right)=E_{total}-U\left(\mathbf{R}_k\right).$$

That probability is given by the *density of states* of the kinetic energy that is

$$\rho\left(E_{kin}\right)=constant\cdot E_{kin}^{3N/2-1}. \tag{4.1}$$

We end up at the following probability for a configuration $\mathbf{R}_k$:

$$\text{Probability}\left(\mathbf{R}_k\right)\propto\left(E_{total}-U\left(\mathbf{R}_k\right)\right)^{3N/2-1}d\mathbf{R}.$$

## 4.3. Calculations of quantities

Most of the quantities that are calculated in this version of MonteLab are identical to the ones in the canonical ensemble (chapter 3). A few ones are however different. The heat capacity is of course omitted here, as the temperature is not defined. Since the total energy is known, not only the average potential energy can be calculated but also the average kinetic energy

$$\left\langle E_{kin}\right\rangle=E_{total}-\left\langle U\right\rangle.$$

Using the density of states expression in Eq. 4.1, the average kinetic energy at a constant energy might be calculated using the boltzmann weighting

$$\left\langle E_{kin} \right\rangle = \frac{\int_0^\infty E_{kin} \cdot \rho\left(E_{kin}\right) \exp\left(-E_{kin}/k_b T\right) dE_{kin}}{\int_0^\infty \rho\left(E_{kin}\right) \exp\left(-E_{kin}/k_b T\right) dE_{kin}} = \frac{3N}{2} k_b T. \tag{4.2}$$

For systems with many atoms, the canonical and the microcanonical ensembles are equivalent, i.e. both the total energy and the temperature are constant. Hence, in a large microcanoncal system, Eq. 4.2 can be used to calculate the temperature. In smaller clusters, it is not appropriate to talk about temperatures. MonteLab still uses Eq. 4.2 to give a hint about how "warm" the cluster is.

If an average $\left\langle A \right\rangle_{E_{total}}$ of a quantity is known for all total energies $E_{total}$ from the ground energy $E_0$, the thermal average $\left\langle A \right\rangle_T$ at a specific temperature $T$ can be calculated by a weighted intergration over all energies

$$\left\langle A \right\rangle_T = \frac{\int_{E_0}^\infty \left\langle A \right\rangle_{E_{total}} \cdot \rho\left(E_{total}\right) \exp\left(-E_{total}/k_b T\right) dE_{total}}{\int_{E_0}^\infty \rho\left(E_{total}\right) \exp\left(-E_{total}/k_b T\right) dE_{total}}. \tag{4.3}$$

As simulation data from the microcanonical simulations is available only up to a maximal energy $E_{\max}$, the integrals in Eq. 4.3 must be truncated. If the temperature is not too high, the truncation error may however be negligible as the contribution is insignificant for large energies.

## 4.4. Metropolis Monte Carlo for constant total energy

THE MONTELAB MICROCANONICAL CLUSTER MONTE CARLO ALGORITHM

1. Start with a configuration $\mathbf{R}_k$ and pick randomly one of the atoms.
2. Generate a trial configuration by changing the coordinates of the chosen particle: $x_t = x_k + U_1[-\Delta, \Delta]$
   $y_t = y_k + U_2[-\Delta, \Delta]$
   $z_t = z_k + U_3[-\Delta, \Delta]$
3. If $U(\mathbf{R}_t) > E_{total}$ reject the trial configuration and let $\mathbf{R}_{k+1} = \mathbf{R}_k$.
   Else, calculate $\gamma = \dfrac{\text{Probability}(\mathbf{R}_t)}{\text{Probability}(\mathbf{R}_k)} = \left(\dfrac{E_{total} - U(\mathbf{R}_t)}{E_{total} - U(\mathbf{R}_k)}\right)^{3N/2-1}$
4. If $\gamma < U_4[0,1]$ reject the trial configuration and let $\mathbf{R}_{k+1} = \mathbf{R}_k$. Else, accept the trial configuration as the new $\mathbf{R}_{k+1}$.
5. Increase $k$ by one a go back to 1.

## FEATURES OF MONTELAB

### 4.5. Initial configuration and meltdown

The initial configuration of the cluster could either come from an input file or be generated by MonteLab. The initial configuration generated by MonteLab is an fcc lattice that is filled up as much as possible with the atoms.

Disregarded the initial configuration is read from a file or generated by MonteLab, the potential energy of the configuration must not exceed the total energy. If it does, the simulation is interrupted.

### 4.6. Steplength adjustment

Identical with the algorithm in Sec 1.10.

### 4.7. Random number generator

Identical with the generator in Sec 1.11.

### 4.8. Tail correction for the potential cut-off

No correction is done for clusters. The potential cut-off should be large enough to get results that are accurate enough.

### 4.9. Restarting a simulation

As presented in Sec 3.15.

## USER INSTRUCTIONS TO MONTELAB - INPUT

By one *Monte Carlo step* we mean one trial move of an atom. When comparing simulation lengths for systems with different number of atoms, it is rather the number of *Monte Carlo steps per atom* that should be used.

### 4.10. Online help

```
Do you want some help and information ( y / n ) ?
```

*A short description to MonteLab may be given. If zero is given to any input, a short help is given and once again MonteLab asks for input.*

### 4.11. Number of atoms

```
Enter Number of Atoms ( <1001 )
```

### 4.12. Surface energy

```
Do you want to sample the surface energy? ( y / n )
```

*The surface energy is defined in Sec. 3.5.*

```
Enter the number of atoms that are surface atoms.
```

*It is not necessary that the "surface atoms" are at the surface. They should just be a subset of all the atoms.*

```
Enter the numbers of the surface atoms
```

*The numbers of the atoms are the same numbers that are given before the coordinates in the Conf file. One numbers should be entered one on each line.*

### 4.13. s(r) max bond length

```
Enter the s(r) max bond length [in Å].
```

*The s(r) spectrum is sampled from 0 to the "s(r) max bond length".*

### 4.14. The total energy

```
Enter the total energy [in eV]
```

*Exactly zero cannot be entered.*

### 4.15. The MBA potential parameters

```
Enter the MBA parameter xi [in eV].
```

```
Enter the MBA parameter epsilon [in eV].
```

```
Enter the MBA parameter r0 [in Å].
```

```
Enter the MBA parameter p [no unit].
```

```
Enter the MBA parameter q [no unit].
```

### 4.16. Cut-off distance

```
Enter Cut-off Distance for the potential [in Å].
```

*Since there is no tail correction for the clusters, the cut-off distance should be rather large.*

### 4.17. Initial configuration

```
Enter 'G' for 'generated' or 'F' for 'file'
```

*Decide whether the initial configuration should be generated as an fcc crystal by MonteLab or fetched from an input file.*

```
From which file should the configuration be fetched.
```

*A configuration input file should end with N lines of particle coordinates. The first column should be numbered 1 to N. The second to fourth columns should give the x, y and z coordinates of the particles, respectively. If anything goes wrong with the reading of the file, the simulation will be interrupted. The potential energy of the initial configuration must be less than the total energy, otherwise the simulation is interrupted.*

### 4.18. Number of sampling intervals for the *s*(***r***) spectrum

```
Into how many s(r) Sampling Intervals do you want to divide
the r axis?
```

*The* $g(r)$ *function will be sampled in equally long intervals in* $r=0$ *to* $r=\text{"s(r) max bond length"}$. *The maximal number of intervals is 10000.*

### 4.19. Open output files

```
Enter a Name of this Simulation.
(Maximum thirty letters and digits.)
```

*The name of the simulation will be given as suffix to all the output files.*

```
Enter Date (or just press enter).
(Maximum ten letters and digits.)
```

*Optional input of date that will be registered in all the output files.*

### 4.20. Use of meltdown

```
Do you want to use a 'Meltdown'? ( Y/N )
Data   sampled   during   the   meltdown   period   will   not   be
included in the final averages.
```

*Enter your choice.*

### 4.21. Use of steplength adjustment

```
Choose between fixed or adjusting Steplength.
Enter 'F' for 'fixed', 'A' for 'adjusting'.
```

*Enter your choice. This question will only be asked if meltdown was chosen in the previous question.*

```
Enter the rejection ratio you want in percentage.
(Within [5%,95%].)
```

*The steplength will be adjusted to give a rejection rate close to the one entered here. If you do not know of anything better, enter 50. The percentage should be given as an integer.*

### 4.22. Steplength

```
Enter (fixed/initial) Steplength for the Simulation.
```

*If the steplength is chosen to be adjusted, this steplength is the one that is used during the first tenth of the meltdown simulation.*

### 4.23. Seed random number

```
Enter a positive integer seed random number.
```

*The seed random number to the random number generator should be any positive integer.*

### 4.24. Setup of the meltdown simulation

```
How many Monte Carlo Steps do you want to perform between
every writing of the energy
```

*Enter any number.*

```
Enter Number of Monte Carlo Steps during the meltdown.
```

*The meltdown is divided into ten parts and there must be at least one writing of the energy in each part. Therefore the total number of Monte Carlo steps must be a multiple of ten times the number of steps between every writing of the energy.*

```
How many Monte Carlo Steps do you want to perform between
every Storing of the s(r) function?
```

*Output of the s(r) function can be done at maximally every energy writing. This number must therefore be a multiple of the number of steps between every writing of the energy. If you do not want any s(r) output, enter a number greater than the total number of meltdown steps.*

```
How many Monte Carlo Steps do you want to perform between
every Writing of atomic coordinates?
```

*Output of the atomic coordinates can be done at maximally every energy writing. This number must therefore be a multiple of the number of steps between every writing of the energy. If you do not want any output of the coordinates, enter a number greater than the total number of meltdown steps.*

### 4.25. Setup of the real simulation

*The same questions about number of steps are asked for the real simulation.*

```
How many Monte Carlo Steps do you want to perform between
every writing of the energy

Enter   Number   of   Monte   Carlo   Steps   during   the   real
simulation.

How many Monte Carlo Steps do you want to perform between
every Storing of the s(r) function?

How many Monte Carlo Steps do you want to perform between
every Writing of atomic coordinates?
```

## USER INSTRUCTIONS TO MONTELAB - OUTPUT

### 4.26. The Conf file

The same as in Sec. 3.33.

## 4.27. The s.r file

The same as in Sec. 3.34.

## 4.28. The Data file

In the header, the MonteLab version number and kind of MonteLab program, simulation name and date and potential type are stated. By a local average we mean the average of a quantity since the last Monte Carlo step for which the quantity was written to the file. With the total average we mean the average since the start of the simulation. Note that the total average is reset after the meltdown.

The quantities are for the whole system and not per particle.
Local averages are then given for

1. The potential energy, the squared potential energy
2. Radii of gyration. The first number is the average of the greatest radii of gyration in all configurations.
3. Mean distance to the centre of mass.
4. $\delta$.
5. The average interatomic distances $\left\langle r_{ij} \right\rangle$ between atoms $i$ and $j$. Also $\left\langle r_{ij}^2 \right\rangle$ are given.
6. If chosen, the surface energy and the squared surface energy.

The heat capacity divided by the Boltzmann constant is a total variance from the start of the simulation. The "Average calculated temperature" is the temperature defined in Eq. 4.2. Recall that it is not appropriate to talk about temperatures for small clusters.

The last used random number is also given to enable restart of simulation (Sec. 1.15.).
Whenever the steplength is changed that is indicated in the file. If a meltdown simulation is used, the end of that one is indicated.

This file ends with a list of the input data and the final results. The steplengths and corresponding ratios of rejection are listed. The steplength that was used in the real simulation is displayed as the *final steplength*. The number of rejected trial moves in the real simulation is given so that a check of the rejection ratio can be calculated. Especially, the number of trial configurations that were rejected because they would lead to a fragmented cluster is written. Total averages and squared total averages are given so that error estimates can be done. Some quantities are given for the final configuration.

## 4.29. Standard output

The same as in Sec. 3.36. (The heat capacity is however not calculated.)

## OBSTACLES AND SUGGESTIONS

One obstacle is that there is a barrier between the solid and the molten phases. Thus, if the simulation is started with a solid structure and the total energy is high enough that the cluster should be molten a considerable fraction of the time, it still might take a very long time until the cluster switches to the molten phase.

Since there is no tail correction for the potential cut-off, the cut-off must be fairly long. For palladium we have used 12 Å. For the clusters we have used between 100 000 and 1 000 000 Monte Carlo steps per atom. The meltdown has been 10% of the real simulation. By checking if the local average of e.g. the potential energy has converged during the meltdown one can check if the meltdown has been long enough.

## PUBLICATION

**Melting of palladium clusters –**
**Canonical and microcanonical Monte Carlo simulation**

Jan Westergren, Arne Rosén and Sture Nordholm
Phys. Chem. Chem. Phys., **5**, 136 (2003)

**Simulation of the Melting and Cooling of Palladium Clusters**

Jan Westergren
Doctoral thesis, Chalmers University of Technology and Göteborg University,
Göteborg, Sweden 2001

# MonteLab MBAANNEAL Chapter 5

## Clusters of atoms interacting with the MBA (Many-Body Alloy) potential. The canonical ensemble (NVT), annealing

## Theory

### 5.1. Annealing speed

MonteLab MBAANNEAL is almost identical to MonteLab MBACLUSTER. The ensemble is the canonical ensemble with fixed number of atoms, volume and temperature. During the optional meltdown period the temperature is fixed but during the real simulation, the temperature is quenched or annealed. With the start temperature $T_1$, the end temperature $T_2$ and the number of Monte Carlo steps during the real simulation *W*, the temperature is changed by $\Delta T = (T_2 - T_1)/W$ after each trial move.

With a very slow annealing from a molten cluster, the cluster might crystallize and find the configuration with a minimum energy. This is the case for instance for $Pd_{55}$, although the freezing/melting show a small hysterisis. On the other hand, $Pd_{147}$ does not find its minimum energy configuration even at reasonably slow annealing.

## Features of MonteLab

### 5.2. Initial configuration and meltdown

As the temperature gradually decreases, it would desirable to simultaneously decrease the steplength. That is also implemented is MonteLab. If the steplength is chosen to be adjusted during meltdown, the steplength will continue to be adjusted throughout the real simulation. The real simulation is divided into ten parts and after each part a new steplength will be calculated according to

$$S_{i+1} = \frac{q_d}{q_i} S_i \qquad i = 2,...10$$

The desired rejection ratio is $q_d$. $S_i$ and $q_i$ are the steplength and the rejection ratio, respectively, of the simulation part $i$. Since the estimation of the steplength will be based on the rejection rate at a higher temperature, the rejection rate will generally be somewhat higher than the desired one.

## USER INSTRUCTIONS TO MONTELAB - INPUT

The input is the same as for MonteLab MBACLUSTER with the exception of a question of final the temperature.

### 5.3 Final temperature

`Enter final T [in K]`

*The temperature must be lower than the initial one.*

## USER INSTRUCTIONS TO MONTELAB - OUTPUT

The output is the same as for MonteLab MBACLUSTER with the exception of that the steplengths during the real simulation is registered in the Data file. Also, the temperature is followed during the simulation.

## PUBLICATION

**Melting of palladium clusters –**
**Canonical and microcanonical Monte Carlo simulation**

Jan Westergren, Arne Rosén and Sture Nordholm
Phys. Chem. Chem. Phys., **5**, 136 (2003)

**Simulation of the Melting and Cooling of Palladium Clusters**

Jan Westergren
Doctoral thesis, Chalmers University of Technology and Göteborg University,
Göteborg, Sweden 2001