

# Contents

<b>1. Deterministic Methods</b> .....	1
1.1 Molecular Dynamics .....	1
1.1.1 Microcanonical Ensemble Molecular Dynamics .....	13
1.1.2 Canonical Ensemble Molecular Dynamics .....	20
1.1.3 Isothermal-Isobaric Ensemble Molecular Dynamics ...	25
1.2 Density Functional Method .....	31
1.3 Problems .....	31
<b>References</b> .....	33
<b>2. Index</b> .....	39



# 1. Deterministic Methods

The kind of systems we are dealing with in this chapter are such that all degrees of freedom are explicitly taken into account. We do not allow stochastic elements representing, for example, an interaction of the system with a heat bath. The starting point is a Newtonian, Lagrangian or Hamiltonian formulation within the framework of classical mechanics. What we are interested in is to compute quantities for such systems, for example, thermodynamic variables, which appear as ensemble averages. Due to energy conservation the natural ensemble is the microcanonical one. However, sometimes it is desirable to compute a quantity in a different ensemble. To allow such calculations within the framework of a Newtonian, Lagrangian or Hamiltonian description, the formulation has to be modified. This results in a reformulation of the differential equations of motion. These equations will be discretized to generate a path in phase space, along which the properties are computed.

## 1.1 Molecular Dynamics

The mathematical background to the molecular dynamics method is presented. Some approximation schemes for the differential equations are discussed. Basic notions such as the computational cell, boundary conditions and the minimum image convention for the calculation of the force are introduced.

The starting point for the Molecular Dynamics (MD) method [27] [28], [29], [30], [31],[32], [33] is a well-defined microscopic description of a physical system. The system can be a few- or many-body system. The description may be a Hamiltonian, Lagrangian or expressed directly in Newton's equations of motion. In the first two cases the equations of motion must be derived by applying the well-known formalisms. The molecular dynamics method, as the name suggests, calculates properties using the equations of motion, and one obtains the static as well as the dynamic properties of a system. As we shall see, the Monte-Carlo method yields the configurational properties, although there is also a dynamic interpretation [34].

The approach taken by the MD method is to solve the equations of motion numerically on a computer. To do so, the equations are approximated by suitable schemes, ready for numerical evaluation on a computer. Clearly there

will be an error involved due to the transition from a description in terms of continuous variables with differential operators to a description with discrete variables and finite difference operators. The order of the entailed error depends on the specific approximation, i.e., the resulting algorithm. In principle, the error can be made as small as desired, restricted only by the speed and memory of the computer.

The molecular dynamics method computes phase space trajectories of a collection of molecules which individually obey classical laws of motion. Note that this includes not only point-particle systems but also collections of "particles with subunits" [97]. Indeed, an algorithm exists that allows systems to have internal constraints as, for example, a system of polymers [?],[38], [39],[40],[41], [42]. Also possible are constraints such as the motion in a specific geometry [43]. Early simulations were carried out for systems where the energy is a constant of motion [27] [28], [29], [30], [31],[32], [33]. Accordingly, properties were calculated in the microcanonical ensemble where the particle number  $N$ , the volume  $V$ , and energy  $E$  are constant. However, in most situations one is interested in the behaviour of a system at constant temperature  $T$ . This is partly due to the fact that the appropriate ensemble for certain quantities is not the micro-canonical but the canonical ensemble. Significant advances in recent years now allow computation within ensembles other than the microcanonical. We will see how the equations of motion are modified to allow such calculations without introducing stochastic forces. The general technique is not restricted to deterministic equations of motion. Rather, equations of motion involving stochastic forces can be simulated. Algorithms covering such problems will be discussed in later chapters; however, some of the material presented here also applies to non-deterministic dynamics. What we have to deal with are equations of the form

$$\frac{dz(t)}{dt} = K(z(t), t) \quad (1.1)$$

where  $z$  is the unknown variable, which might be, for example, a velocity, an angle or a position, and  $K$  is a known operator. The variable  $t$  is usually interpreted as the time. We shall not restrict ourselves to a deterministic interpretation of (1.1) but allow  $u(t)$  to be a random variable. For example, we might be interested in the motion of a Brownian particle and (1.1) takes on the form of the *Langevin equation*

$$\frac{dv(t)}{dt} = -\beta v(t) + \underline{R}(t) \quad (1.2)$$

Since the fluctuation force  $R(t)$  is a random variable, the solution  $v(t)$  to the Stochastic Differential Equation (SDE) will be a random function. We may distinguish four types of the Equation (1.1):

1.  $K$  does not involve stochastic elements and the initial conditions are precisely known;

2.  $K$  does not involve stochastic elements but the initial conditions are random;
3.  $K$  involves random force functions; or
4.  $K$  involves random coefficients.

We treat types 1 - 3 in this text. In the case of types 1 and 2 the task of solving (1.1) reduces to an integration. For type-3 problems, special precautions have to be taken, since the properties of the solution are developed through probabilistic arguments. For simplicity, we assume for the remainder of this chapter that we are dealing with monatomic systems so that the molecular interactions do not depend on the orientation of the molecules. Furthermore, we will always deal with pair wise additive central-force interactions. To stress once again the point made earlier, the technique is not restricted to such systems. The inclusion of orientation-dependent interactions and the constraints of connectivity would unnecessarily complicate the exposition. In general, the system will be described by the Hamiltonian

$$\mathcal{H} = \frac{1}{2} \sum_i \frac{p_i^2}{m_i} + \sum_{i < j} u(r_{ij}) \quad (1.3)$$

where  $r_{ij}$  is the distance between the particles  $i$  and  $j$ . For ease of reference, we abbreviate the configurational internal energy as

$$U(r) = \sum_{i < j} u(r_{ij}) \quad (1.4)$$

Let the system consist of  $N$  particles. Since we restrict ourselves to properties of the bulk at a specific density  $\rho$  we must introduce a volume, the *MD cell*, to retain a constant density. If the system is in thermal equilibrium, the shape of the volume is irrelevant [?]. This is true, of course, for gases and liquids in the limit where the volume is large enough. For systems in a crystalline state the shape does make a difference. For liquid or gaseous states we take a cubic volume for computational simplicity. Let  $L$  be the linear size of the MD cell with volume  $V = L^3$ . The introduction of the box creates six unwanted surfaces. Particles hitting these surfaces would be reflected back into the interior of the cell. Especially for systems with a small number of particles, important contributions to any property would come from the surfaces. To reduce the effect of the surfaces we impose periodic boundary conditions (pbc), i.e., the basic cell is identically repeated an infinite number of times. Mathematically this is stated as follows. For any observable  $A$  we have

$$A(\underline{x}) = A(\underline{x} + \underline{n}L), \quad \underline{n} = (n_1, n_2, n_3) \quad (1.5)$$

for all integers  $n_1, n_2, n_3$ . The computational implementation is that if a particle crosses a surface of the basic cell it re-enters through the opposite wall with unchanged velocity. With the periodic boundary conditions we have eliminated the surfaces and created a quasi-infinite volume to represent the

macroscopic system more closely. The assumption involved is that the small volume is embedded in an infinite block. Each component of a position vector is represented by a number between zero and  $L$ . If particle  $i$  is at  $\underline{r}_i$ , there is a set of image particles at positions  $\underline{r}_i + \underline{n}L$ ,  $\underline{n}$  being an integer vector. Due to the periodic boundary conditions the potential energy is affected because we have

$$U(\underline{r}_1, \underline{r}_2, \dots, \underline{r}_N) = \sum_{i < j} u(r_{ij}) + \sum_n \sum_{i < j} u(|\underline{r}_i - \underline{r}_j + \underline{n}L|) \quad (1.6)$$

In order to avoid the infinite summation in the second term on the right-hand side we introduce a convention about how the distances are computed [3.20, 21].

**Minimum Image Convention:**

The distance  $r_{ij}$  between particle  $i$  at  $\underline{r}_i$  and particle  $j$  at  $\underline{r}_j$  is

$$r_{ij} = \min \{ |\underline{r}_i - \underline{r}_j| + \underline{n}L \} \quad \text{over all } \underline{n}$$

A particle in the basic cell interacts only with each of the  $N - 1$  other particles in the basic cell or their nearest images. In effect, we have cut off the potential by the condition

$$r_c < \frac{L}{2} \quad (1.7)$$

The price to be paid is that we neglect the background. It would be more realistic to include the interaction of each particle with all the image particles. An elegant procedure for doing so has been worked out by Ewald [3.22, 23]. The question of how the properties under computation are influenced is not yet fully understood and remains to be investigated more closely. Better understood are the boundary conditions applied within the Monte-Carlo method. The value of  $L$  should be chosen so large that the forces that would occur for distances larger than  $L/2$  are negligibly small, to avoid finite-size effects.

A cubic volume is, of course, not the only possible geometry to confine the system and to conserve the density (Problem 3.1). Some applications, for example crystallization, require different choices [3.24-27]. In any case, there is a danger that the periodic boundary conditions impose a particular lattice structure [3.28,29].

**Integration Schemes**

From a numerical-mathematics point of view the MD method is an initial value problem. A host of algorithms have been developed [3.30,31] for this problem, which are, however, not all applicable in the context of physical problems. The reason is that many schemes require several evaluations of the right-hand side of (1.1, storage of previous evaluations and/or iterations. Specifically, assume that (1.1 was derived from the Hamiltonian (1.3), i.e., the equations of motion are

$$m \frac{dr_i}{dt} = p_i, \quad \frac{dp_i}{dt} = \sum_{i < j} F(r_{ij}) \quad (1.8)$$

Each evaluation of the right-hand sides for the  $N$  particles takes  $N(N - 1)/2$  quite time-consuming operations. To avoid this, simpler schemes are employed which suffice in accuracy for most applications. The conservation properties are also a problem, as we shall discuss below. To solve the equations of motion on a computer we construct a finite difference scheme for the differential equations to the highest possible order. From the difference equations we then derive recursion relations for the positions and/or velocities (momenta). These algorithms perform in a step-by-step way. At each step approximations for the positions and velocities are obtained, first at time  $t_1$  then at  $t_2 > t_1$ , etc. Hence, the integration proceeds in the time direction (time-integration algorithms). One also say that the simulation is time-driven as opposed to event-driven. The recursion relation must clearly allow efficient evaluation. In addition, the scheme must be numerically stable.

The most straightforward discretization of the differential equation stems from the Taylor expansion. The idea is to base the algorithm on a discrete version of the differential operator. With suitable assumptions we can expand the variable  $z$  in a Taylor series

$$z(t+h) = z(t) + \sum_{i=1}^{n-1} \frac{h^i}{i!} z^{(i)}(t) + R_n \quad (1.9)$$

where  $R_n$  gives the error involved in the approximation.

Equation (1.9) allows an immediate construction of a difference scheme (symmetric difference approximation) with a discretization error of the order  $h$ . Let  $n = 2$ , then

$$\begin{aligned} \frac{dz(t)}{dt} &= h^{-1} [z(t+h) - z(t)] + \mathcal{O}(h) \\ &= h^{-1} [z(t) - z(t-h)] + \mathcal{O}(h) \end{aligned} \quad (1.10)$$

These are the simplest schemes; the first we are familiar with from the example in Chap.1. The top equation of 1.10 is called the forward difference quotient and the lower equation the backward one. Using the forward difference we get the Euler algorithm [3.31] for the solution of the general problem 1.1) with the initial value  $z_t$ , at the starting time  $t$ , i.e.,

$$z(t) = z_t, \quad z(t+h) = z(t) + hK(z(t), t) \quad (1.11)$$

The Euler algorithm represents a typical example of a one-step method. Such methods use the previous value as the only input parameter to determine a new value. We shall now derive the error introduced by employing the

algorithm. We have to distinguish between the local and the global error. Let  $q(t)$  be the exact solution of

$$\frac{dq(t)}{dt} = K(q(t), t) \quad (1.12)$$

Define a function

$$\mu(z, t, h) = \begin{cases} \frac{q(t+h)-z}{h}, & h \neq 0, \\ K(z, t), & h = 0 \end{cases}$$

which is the difference quotient of the exact solution. The difference

$$\tau(z, t, h) = \mu(z, t, h) - K(z, t) \quad (1.13)$$

measures the *local discretization error*. If

$$\tau(z, t, h) = \mathcal{O}(h^p) \quad (1.14)$$

then the method is of order  $p$ . The Euler algorithm has  $p = 1$ . We can go even further and ask for the global discretization error. It can be shown [3.31] that the global error is equal to the local error for the one-step method. So far we have considered only one-step methods. A more sophisticated scheme yielding a two-step method is immediately derived by using (3.9) with  $n = 3$ :

$$\begin{aligned} z(t+h) &= z(t) + h \frac{dz(t)}{dt} + \frac{1}{2} h^2 \frac{d^2 z(t)}{dt^2} + R_3 \\ z(t-h) &= z(t) - h \frac{dz(t)}{dt} + \frac{1}{2} h^2 \frac{d^2 z(t)}{dt^2} + R_3^* \end{aligned} \quad (1.15)$$

Note that  $R_3 \neq R_3^*$ . Subtracting the second equation from the first, we get

$$z(t+h) = z(t-h) + 2h \frac{dz(t)}{dt} + R_3 + R_3^* \quad (1.16)$$

The error analysis shows that it is of order  $h^3$ . Hence

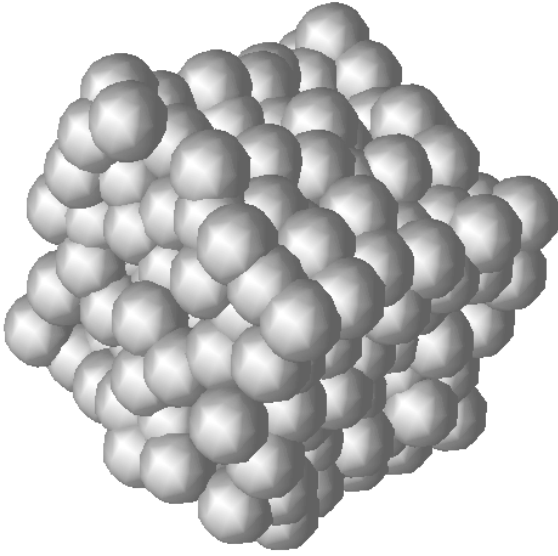
$$\frac{dz(t)}{dt} = \frac{1}{2h} [z(t+h) - z(t-h)] + \mathcal{O}(h^2) \quad (1.17)$$

Using the same idea we obtain for the second derivative

$$z^{(2)}(t) = h^{-2} [z(t+h) - 2z(t) + z(t-h)] + \mathcal{O}(h^2) \quad (1.18)$$

*Multi-step methods* allow the construction of algorithms of high order. Typical members of this class employed in simulational physics are those developed by Gear [3.33-35], Beeman [3.36], and Toxvaerd [3.37]. Such methods (including the one-step) have the general form





**Figure 1.1.** Configuration snapshot

$$z(t + rh) + \sum_{\nu=0}^{r-1} a_{\nu} z(t + \nu h) = hG(t; z(t + rh), \dots, z(t); h) \quad (1.19)$$

where  $G$  is some function of  $K$ , for example

$$G = \sum_{\nu=0}^r bK(z(t + \nu h), t + \nu h) \quad (1.20)$$

We distinguish between predictor and corrector schemes. In a predictor scheme  $G$  does not depend on  $z(t + rh)$  whereas it does in a corrector scheme. The Beeman algorithm, for example, is a third-order predictor-corrector scheme.

Now that we have derived some algorithms to solve the equations of motion numerically, the question arises as to the choice of the basic time step  $h$  (MD step). It determines the accuracy of the computed trajectory. Consequently,  $h$  affects the accuracy of the computed properties, in addition to the statistical error. But the choice of  $h$  is also important with regard to the simulated real time. For many problems it is desired to simulate a fairly long real time. The question is how large can the time step be? Consider, for example, an argon system of  $N$  particles (a snapshot is shown in Figure 1.1, which will be the standard example in this chapter. The interaction between the particles is assumed to be of the Lennard-Jones type. For the

argon system a time step  $h \propto 10^{-2}$  was found sufficient in most regions of the phase diagram [3.6,7]. Here  $h$  is a dimensionless quantity and the real time equivalent is roughly  $10^{-14}s$ . Hence, a simulation lasting 1000 steps yields a real time equivalent of  $10^{-11}s$ .

In connection with the number of MD steps carried out,  $h$  determines how much of the phase space is sampled. Naturally one would like to make  $h$  as large as possible to sample large portions. However,  $h$  determines the time scale, and we have to consider the time scale(s) on which changes in the system occur. Some systems have several different scales. A molecular system may have one time scale for intramolecular modes and another for intermolecular modes.

One reason for energy fluctuations is the potential cut-off to be described later. A second reason is the error entailed by the approximation. No matter how high the order of an algorithm, the system will eventually depart from the true trajectory, as long as  $h$  is finite. A drift in the energy  $\delta E$  is caused by the finite time step, though the drift might be small.

From a more general point of view we can ask for the conservation properties of the algorithms. The energy, and the linear and angular momenta should be conserved during the course of a molecular-dynamics simulation. One way to establish conservation is to constrain the system artificially [3.37]. There is, however, a rigorous way of enforcing conservation [3.38-40]: Instead of using forces to calculate the motion one should use the potentials. It can be shown [3.39,40] that with this approach the energy, and the linear and angular momenta remain constant if the algorithm is set up in a special form. Nevertheless, there is still the discretization error so that the computed trajectory is not the "true" one, even though the energy is conserved. The system will follow an alternative path on the constant-energy surface. It is also required that the potential is the true one. This is, however, not the case for a system enclosed by a finite box. In addition, we may ask for the time-reversal properties. Interestingly enough, only the one-step method is invariant under time reversal if we require that the equations define a canonical transformation [3.41,42].

We return to the reason for energy fluctuations. Such fluctuations may be produced due to the finite arithmetic of the computer as well as the finite step width. Though rounding errors usually play a less important role than the other phenomena, they nevertheless deserve consideration. Associated with each arithmetic operation is a round-off error [3.43]. The result due to an addition is obtained with finite precision so that the last digit is not the true one. Rather, it is the result of rounding. An error is also created on adding two quantities with quite different orders of magnitude (note that on a computer the associative property of addition does not hold!). This can occur in the calculation of the force acting on a particle. Imagine that at least one particle exerts a strongly repelling force, some particles are near the potential minimum giving only a negligible contribution, and the others

are far away. Adding the smaller contributions to the dominant repelling force will result in a loss of accuracy of some digits. However, if the summation is carried out by first sorting the contributions according to their magnitude and then summing, beginning with the smallest terms, significant digits are secured.

### Calculating Thermodynamic Quantities

In computer simulations of physical systems the ensemble average has to be replaced by the time average. In conventional MD simulations the number of particles,  $N$ , and the volume  $V$  are fixed. Strictly speaking the total linear momentum is another conserved quantity. The total linear momentum is set to zero to avoid motion of the system as a whole. From the equations of motion, given the initial positions  $\mathbf{r}^N(0)$  and momenta  $\mathbf{p}^N(0)$ , a MD algorithm generates the trajectory  $(\mathbf{r}(0), \mathbf{r}^N(0))$ . Assuming that the energy is conserved and that the trajectories spend equal time in all equal volumes with the same energy, the trajectory average, defined as

$$\bar{A} = \lim_{t' \rightarrow \infty} (t' - t_0)^{-1} \int_{t_0}^{t'} dt A(r^N(t), p^N(t); V(t)) \quad (1.21)$$

is equal to the microcanonical ensemble average

$$\bar{A} = \langle A \rangle_{NVE} \quad (1.22)$$

In the following we shall always denote ensemble averages by  $\langle . \rangle$  and trajectory averages by an overscore. For later application we include a non-constant volume in the definition. For now, the volume does not change in time and has a definite value determined by the number of particles and the density.

The total energy is a conserved quantity for an isolated system. Along any trajectory generated by a molecular-dynamics simulation the energy should remain constant, i.e.,  $E = \bar{E}$ . At this point we have to consider the range of interaction. In general, the range will be longer than the length  $L$  of the side of the MD cell and is cut off at  $r_c < L/2$ . This natural cut-off is, however, not the only one. For computational reasons the potential is usually truncated at a convenient range to reduce the time spent in computing the potential energy. Indeed, if no special precautions are taken 99% of the total execution time required for one MD step can go into the computation of the potentials, i.e., the forces required to propagate the particles.

The cut-off introduces a  $\delta$ -function singularity in the forces at the point of cut-off if the potential is not smoothly continued to zero. If the potential is given in a tabulated form, this is readily implemented. But the effects of the truncation on the properties of the system must be considered. In non-equilibrium situations, for example metastable states occurring at first-order phase transitions, the range is extremely important. It affects the relaxation of the non-equilibrium into the equilibrium state [3.44, 45].

The cut-off and the approximations made for the differential equations of motion, together with numerical round-off errors, introduce a drift in the energy. The trajectories are then not time reversible either.

The kinetic energy  $E_k$  and the potential energy  $U$  are not conserved quantities for an isolated system. Their values vary from point to point along the generated trajectory, and we have

$$\begin{aligned}\bar{E}_k &= \lim_{t' \rightarrow \infty} (t' - t_0)^{-1} \int_{t_0}^{t'} E_k(v(t)) dt \\ \bar{U} &= \lim_{t' \rightarrow \infty} (t' - t_0)^{-1} \int_{t_0}^{t'} U(r(t)) dt\end{aligned}\quad (1.23)$$

Let us first look at the kinetic energy. The path generated is not continuous and we have to take the average of the kinetic energy evaluated at the discrete points  $\nu$  in time

$$\bar{E}_k = \frac{1}{n - n_0} \sum_{\nu > n_0}^n E_k^\nu \quad (1.24)$$

where

$$E_k^\nu = \sum_{i=1}^N \sum_{l=1}^3 \frac{1}{2} m (v_{il}^2)^\nu \quad (1.25)$$

From the mean kinetic energy we can compute the temperature of the system. As will become apparent later, the temperature is an important quantity to monitor, especially during the initial stages of a simulation. Recall that we are interested in the computation of observables in the thermodynamic limit. In this limit all ensembles are equal, and we can apply the equipartition theorem:

If the Hamiltonian is given as in (1.1) we have

$$\frac{1}{2} m v_l^2 = \frac{1}{2} k_B T \quad (1.26)$$

Since the system has three degrees of freedom per particle (for the moment we ignore constraints such as zero total linear momentum), we obtain

$$\bar{E}_k = \frac{3}{2} N k_B T \quad (1.27)$$

Assume that the potential has been cut off at  $r_c$ . The average internal configurational energy is then given by

$$\bar{U} = \frac{1}{n - n_0} \sum_{\nu > n_0}^n U^\nu \quad (1.28)$$

**Figure 1.2.** Shown are pair correlation functions for two parameter sets as obtained from simulations. The left picture shows a high temperature state  $T' = 2.53$  with a density  $\rho = 0,063$ . The right shows the pair correlation function for  $T' = 0.722$  and  $\rho = 0.831$  (c.f. Example in the next section)

where

$$U^\nu = \sum_{i < j} u(r_{ij}^\nu) \quad (1.29)$$

Due to the cut-off the total energy and the potential energy entail an error. To estimate the necessary corrections we note that the potential energy is, in general, given by

$$\frac{U}{N} = \frac{1}{2} \rho \int_V u(r)g(\underline{r})d^3r = 2\pi\rho \int_0^\infty u(r)g(r)r^2dr \quad (1.30)$$

where  $g(\underline{r})$  is the pair correlation function and measures the time-independent correlations among the particles. To be precise,  $g(\underline{r})d^3r$  is the probability that a particle is found in the volume element  $d\underline{r}$  surrounding  $\underline{r}$  when there is a particle at the origin  $\underline{r} = \underline{0}$ . In a homogenous system  $g(\underline{r}) = g(r)$ . Let  $n(r)$  be the average number of particles situated at a distance between  $r$  and  $r + \Delta r$  from a given particle, then

$$g(r) = \frac{V}{N} \frac{n(r)}{4\pi r^2 \Delta r} \quad (1.31)$$

The pair correlation function is easily computed during a simulation. All the distances are available anyway from the calculation of the forces. Since  $g(r)$  is time independent one can perform a time average. Figure 1.2 shows  $g(r)$  for argon at two points of the phase diagram, as obtained by MD simulations. The pair correlation function is only meaningfully calculated for distances roughly less than half of the linear size of the MD cell. In (???) all the internal configurational energies are summed up to the cut-off distance. For the tail correction we can take

$$U_c = 2\pi\rho \int_{r_c}^\infty u(r)g(r)r^2dr \quad (1.32)$$

Instead of actually taking  $g(r)$  as computed during a simulation, one can also assume that the pair correlation function is identical to unity. The error made in such an approximation will be small if the potential cut-off was not chosen too small. For the results shown in Figure 1.2 the potential was a Lennard-Jonesian one with a cut-off to the right of the second peak, as indicated by the arrows. There, the pair correlation function is not too far off unity and will stay so for the range extending to infinity. A tail correction

is also necessary for other quantities. As an example we take the calculation of the pressure  $P$  for which the virial equation of state holds [3.19]:

$$P = \rho k_B T - \frac{\rho^2}{6} \int_0^\infty g(r) \frac{\partial u}{\partial r} 4\pi r^3 dr \quad (1.33)$$

As for the computation of the potential energy, we split the integral into a term due to the contributions within the interaction range and a term to correct for the truncation:

$$R = \rho k_B T - \left\langle \frac{\rho}{6N} \sum_{i<j} r_{ij} \frac{\partial u}{\partial r_{ij}} \right\rangle + P_c \quad (1.34)$$

The long-range correction is

$$P_c = \frac{\rho^2}{6} \int_{r_c}^\infty g(r) \frac{\partial u}{\partial r} 4\pi r^3 dr \quad (1.35)$$

In the example of the next section the significance of the corrections to the various quantities will be appreciated. They can amount to several percent.

### Organisation of a Simulation

The actual computer simulation of a molecular system can be broken up into three parts:

1. Initialisation
2. Equilibration
3. Production.

The first part of a simulation is the assignment of the initial conditions. Depending on the algorithm, different sets are required. An algorithm may need two sets of co-ordinates, one at time zero and one for the previous time step. For the moment assume that to start an algorithm we need the positions and the velocities. The problem one is faced with immediately is that, in general, the initial conditions are not known. Indeed, this is the starting point for a statistical-mechanics treatment! For the computer-simulation approach there are various possible assignments. For definiteness let the initial positions be on a lattice and the velocities drawn from a Boltzmann distribution. The precise choice of the initial conditions is irrelevant, since ultimately the system will lose all memory of the initial state.

A system set up, as outlined above, will not have the desired energy. Secondly, most probably the state does not correspond to an equilibrium state. To promote the system to equilibrium we need the equilibration phase. In this phase, energy is either added or removed until the energy has reached the required value. Energy may be removed or added by stepping the kinetic energy down or up. The system is now allowed to relax into equilibrium by integrating forward the equations of motion for a number of time steps.

Equilibrium is established if the system has settled to definite mean values of the kinetic and potential energies.

We can identify at least two potential problems arising in the first two steps. One problem concerns the relaxation time of the system. The basic time step  $h$  determines the real time of the simulation. If the intrinsic relaxation time is long, many steps are required in order for the system to reach equilibrium. For some systems the number of time steps may be prohibitively large for the present speed of computers. However, it is possible in some circumstances to circumvent the difficulty by an appropriate scaling of the variables. Examples of where this is possible are systems near second-order phase transitions.

In connection with the relaxation time one has to face the possibility that the system is trapped in a metastable state. Long-lived metastable states may not show an appreciable drift in the kinetic or potential energy. Especially for systems investigated near two-phase coexistence, say between liquid and gas, this danger arises.

The second potential problem is that the system might have been set up in an irrelevant part of the phase space. This problem can be handled by performing simulations with different initial conditions and different lengths.

The actual computation of the quantities is done in the third part of the simulation. In the production part all quantities of interest are computed along the trajectory of the system in phase space.

In the following we shall study particular algorithms. First we look at methods to deal with the constant energy, constant particle number and constant volume cases. Then we study ways of incorporating into the equations of motion constraints allowing a simulation of a constant temperature rather than of a constant energy. This will follow a discussion of how to compute properties in a constant pressure ensemble.

### 1.1.1 Microcanonical Ensemble Molecular Dynamics

The basic molecular dynamics algorithm with conserved energy is introduced.

We proceed by developing a computational method to propagate a system along a path of constant energy in the phase space. The starting point is the Hamiltonian describing the interaction of  $N$  particles. For simplicity, we assume, as before, a two-body potential with spherical symmetry

$$\mathcal{H} = \frac{1}{2} \sum_i \frac{1}{m} p_i^2 + \sum_{i < j} u(r_{ij}) \quad (1.36)$$

where  $r_{ij}$  denotes the distance between particle  $i$  and particle  $j$ . Time does not enter explicitly into the equations. We are considering a system where  $\mathcal{H} = \mathcal{E}$  is a constant of motion. In addition, we have a constant particle number  $N$ , and the fourth constraint of zero total linear momentum  $P$ .

In classical mechanics the Hamiltonian leads to various forms of the equations of motion. Depending on the choice, the algorithm to solve the equations will have certain features. Though the equations of motion are mathematically equivalent they are not numerically equivalent. Here we start with the Newtonian form

$$\frac{d^2 \underline{r}_i(t)}{dt^2} = \frac{1}{m} \sum_{i < j} \underline{F}_i(r_{ij}) \quad (1.37)$$

Analytically, the solution of the system of second-order differential equations is obtained by integrating twice from time zero to  $t$ , to obtain first the velocities and then the positions. Not only the initial positions are required but also the initial velocities. The initial positions fix the contribution of the potential energy to the total energy, and the velocities determine the kinetic energy contribution. With the specification of the initial conditions the system moves along a path of constant energy in phase space.

To solve the differential equations numerically we use the discretization (???) for the second-order differential operator on the left-hand side of (1.37) to get the explicit central difference method

$$\frac{d^2 \underline{r}_i}{dt^2} = \frac{1}{h^2} [\underline{r}_i(t+h) - 2\underline{r}_i(t) + \underline{r}_i(t-h)] = \frac{1}{m} \underline{F}_i(t) \quad (1.38)$$

This equation provides a prescription for obtaining the positions of the particles at time  $t+h$  from the positions at two immediately preceding time steps  $t$  and  $t-h$  and the forces acting at time  $t$ . Solving for the positions at time  $t+h$  we get

$$\underline{r}_i(t+h) = 2\underline{r}_i(t) - \underline{r}_i(t-h) + \underline{F}_i(t)h^2/m \quad (1.39)$$

Let

$$\begin{aligned} t_n &= nh \\ \underline{r}_i^n &= \underline{r}_i(t_n) \\ \underline{F}_i^n &= \underline{F}_i(t_n) \end{aligned}$$

Then (1.39) assumes a more algorithmic form

$$\underline{r}_i^{n+1} = 2\underline{r}_i^n - \underline{r}_i^{n-1} + \underline{F}_i^n h^2/m \quad (1.40)$$

Starting from positions  $\underline{r}_i^0$  and  $\underline{r}_i^1$  all subsequent positions are determined by the above recursion relation. In other words, the positions of the particles at time  $n+1$  are extrapolated or predicted from the two immediately preceding positions (two-step method).

In the above form the recursion relation produces only the positions. The velocities, however, are needed for the calculation of the kinetic energy and,



for example, the velocity auto-correlation function to study transport properties. Following the line of approach used so far, the velocities are computed as, see (??),

$$\underline{v}_i^n = (\underline{r}_i^{n+1} - \underline{r}_i^{n-1}) / 2h \quad (1.41)$$

Notice that at the  $(n+1)$ -th step the computed velocities are those of the previous time, i.e., the  $n$ -th step! Hence, the kinetic energy is one step behind the computed potential energy. Equations (??) together with the initial positions constitute the so-called Verlet algorithm [3.6,7].

**Algorithm A2. NVE Molecular Dynamics**

1. Specify positions  $\underline{r}_i^0$  and  $\underline{r}_i^1$ .
2. Compute the forces at time step  $n$ :  $\underline{F}_i^n$ .
3. Compute the positions at time step  $n+1$  as in (3.34):  $\underline{r}_i^{n+1}$ .
4. Compute the velocities at time step  $n$  as in (3.35):  $\underline{v}_i^n$ .

One advantage of the above algorithm is its time reversibility. Running the system backwards in time leads to the same equations. This is true only in principle. Due to inevitable round-off errors of the finite precision arithmetic, the trajectories depart from their original paths. At each time step there is an addition of the form  $O(1) + O(h^2)$ , introducing a round-off error. Further, the trajectory departs from the true one because of the finite step size.

In the form of the Verlet algorithm (Algorithm A2) the method is not self-starting. Not only the initial positions must be supplied but also one more set of positions. Sometimes this comes in handy if one sets up a lattice for the initial positions of the  $N$  particles and then perturbs it. If the positions and the velocities are initial conditions, the following procedure can be used to calculate the positions at  $\underline{r}_i^1$ :

$$\underline{r}_i^1 = \underline{r}_i^0 + h\underline{v}_i^0 + \frac{1}{2m}h^2\underline{F}_i^0 \quad (1.42)$$

From then on the algorithm proceeds as presented. The Verlet algorithm can be reformulated in such a way as to give a numerically more stable method [3.30,46]. Define

$$\underline{z}_i^n = \frac{\underline{r}_i^{n+1} - \underline{r}_i^n}{h} \quad (1.43)$$

The equations

$$\begin{aligned} \underline{r}_i^n &= \underline{r}_i^{n-1} + h\underline{z}_i^{n-1} \\ \underline{z}_i^n &= \underline{z}_i^{n-1} + m^{-1}h\underline{F}_i^n \end{aligned} \quad (1.44)$$

are (mathematically) equivalent to ( ) and are called the summed form. A further reformulation yields the velocity form of the Verlet algorithm.

**Algorithm A3. NVE MD Velocity Form**

1. Specify the initial positions  $r_i^1$ .
2. Specify the initial velocities  $v_i^1$ .
3. Compute the positions at time step  $n + 1$  as

$$r_i^{n+1} = 2r_i^n - r_i^{n-1} + \underline{F}_i^n h^2/m$$

4. Compute the velocities at time step  $n+1$  as

$$v_i^{n+1} = v_i^n + h(\underline{F}_i^{n+1} + \underline{F}_i^n)/m$$

The above algorithm is superior to the original one in many ways. Notably, we have succeeded in having the positions and the velocities for the same time step; secondly, the numerical stability is enhanced, which is extremely important for long runs. Yet another feature will show up when we discuss algorithms for the constant temperature ensemble.

In general, one does not know the precise initial conditions corresponding to a given energy. To adjust the system to a given energy, reasonable initial conditions are supplied and then energy is either drained or added. The procedure is carried out until the system reaches the desired state. For the equilibration phase in the Verlet algorithm, or its variant velocity forms, this is accomplished by an ad hoc scaling of the velocities [3.47]. Such a scaling can introduce large changes in the velocities. To eliminate possible effects the system must be given time to establish equilibrium again. Algorithmically the equilibration phase looks like

1. Integrate the equations of motion for some time steps.
2. Compute the kinetic and potential energies.
3. If the energy is not equal to that desired, then scale the velocities.
4. Repeat from step 1 until the system has reached equilibrium.

The success of the procedure depends on the initial positions and the distribution of the velocities. A common practice is to set up the system on a lattice and assign velocities according to a Boltzmann distribution. Sometimes, instead of the velocities being scaled, they are all set to zero. In any case, one has to check the velocity distribution after the equilibration phase has been reached to make sure that it has the equilibrium Maxwell-Boltzmann form.

### Example 3.1

We study a monatomic system of particles in which the total energy is fixed. In particular, we assume that the interaction between the particles is well represented by a two-body central force interaction of the Lennard-Jones type

$$u(r_{ij}) = 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right] \quad (1.45)$$

where  $-\epsilon$  is the minimum of the potential ( $\epsilon$  specifies the units of energy), which occurs when the distance  $r$  is equal to  $2^{1/6}\sigma$  ( $\sigma$  specifies the units of length). To be more specific we choose the values of  $\epsilon$  and  $\sigma$  appropriate for argon (Problems 3.5,6). We shall take  $N = 256$  particles in a box of volume  $V$ . Periodic boundary conditions are imposed to conserve the density. This implies that we have to use the minimum image convention (3.2). The volume and the number of particles together with the energy completely specify the point in the phase diagram we wish to study. To advance the particles inside the MD cell we need to know the forces acting on each particle. For the force in the  $x$ -direction on the  $i$ -th particle exerted by the  $j$ -th particle we obtain from (3.39)

$$F_x(r_{ij}) = 48 \left( \frac{\epsilon}{\sigma^2} \right) (x_i - x_j) \left[ \left( \frac{\sigma}{r_{ij}} \right)^{14} - \frac{1}{2} \left( \frac{\sigma}{r_{ij}} \right)^8 \right] \quad (1.46)$$

and similarly for the  $y$ - and  $z$ -components. This form of the potential and the force is, however, not suitable for a computer simulation. All quantities are conveniently expressed in a scaled form. Time and positions are scaled by

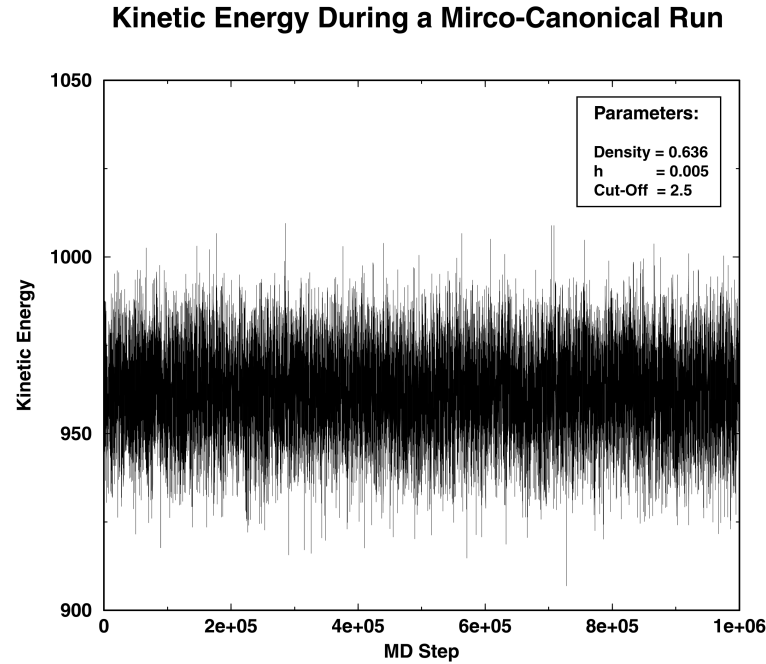
$$\left( \frac{m\sigma^2}{48\epsilon} \right)^{1/2}, \quad \sigma \quad (1.47)$$

respectively ( $m$  is the mass of the argon atom). This renders the equations dimensionless. Substituting the values for the argon atom into (3.41) the time unit is  $310^{-12}s$ . To ensure a reasonable numerical stability the basic time increment is taken to be  $h = 0.064$  or  $210^{-14}s$ . The actual real time will be fairly small since only a limited number of integration steps are possible.

We shall study the argon system at two points in the phase diagram:  $(T', p^*) = (2.53, 0.636)$  and  $(0.722, 0.83134)$ . For these values of the reduced densities the linear MD cell sizes are  $L = 7.38$  and  $L = 6.75$ , respectively. With these specifications the program can be set up. We use the summed form of the Verlet algorithm to advance the positions.

Initially we assign random positions of the atoms. To start the algorithm, velocities are drawn at random and scaled for the appropriate temperature. Since the MD cell should not move, we must assure a zero total linear momentum. This removes three degrees of freedom from the system and must be taken into account in the calculation of the temperature.

At this point, consideration must be given to the computation of the forces and the potential. To avoid the use of Ewald sums and to speed up the calculation we truncate the potential. To study the effect of the cut-off we use two values,  $r_c = 2.5$  and  $3.6$ . The impact of the truncation on the execution time is quite large. In going from  $r_c = 2.5$  to  $3.6$  the execution time doubles for the densities considered here! The cut-off can further be appreciated by noting that for  $r_c = 2.5$  roughly 80



**Figure 1.3.** Evolution of the kinetic energy during a molecular dynamics simulation. All quantities are given in reduced units

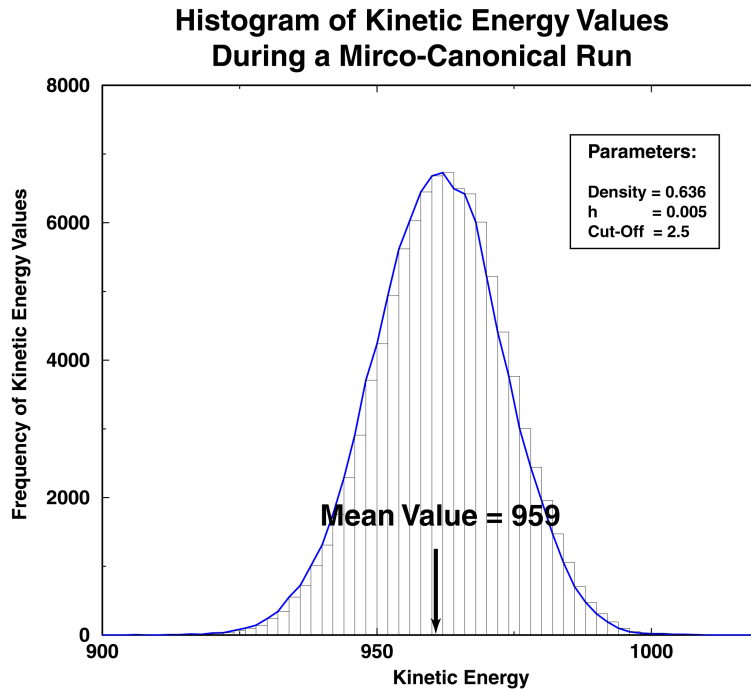
$$\beta = \left[ T^* (N - 1) / 16 \sum_i v_i^2 \right]^{1/2} \quad (1.48)$$

) The procedure is repeated until the desired energy, or equivalently an average temperature, is reached. In these particular simulations the equilibration was performed during the first 1000 MD steps. Figure 3.3 shows the evolution of the kinetic energy for the case  $r_c = 2.5$ ,  $T' = 2.53$ ,  $\rho = 0.636$

Looking at the potential energy we observe that the overall relaxation of the system is much slower in the case  $T' = 0.722$ .

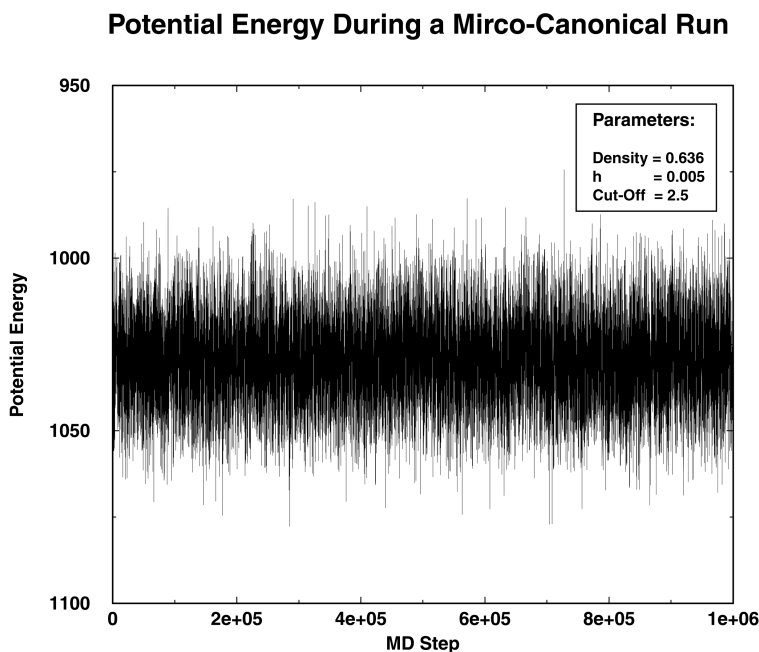
To check that the system does indeed behave as expected, we look at the distribution of the kinetic energy values encountered during the simulation.

The results of the simulation in the example represents one particular realization out of a multitude of possible ones. Starting from a different set of initial positions and velocities the system would have followed an alternative path on the constant energy surface. Contenting ourselves with one path, we rely on (3.21), i.e., that the trajectory average is equal to the ensemble average. In principle we should have followed the path for an infinitely long time to ensure that the system spends equal time in all equal volumes of the



**Figure 1.4.** Distribution of the kinetic energy

phase space. The limitation of the finite computer time does not permit this. We sampled some region in phase space, so it follows that there will be an error involved in the results. It might be that the path sampled an irrelevant part of the phase space. For example, the initial conditions might be such that the system is set up in an irrelevant part. If the duration of the simulation is too small, the system does not leave the irrelevant part, or only just enters the relevant part. Simulations of different lengths must be made in order to assess the error, i.e., to determine whether the asymptotic behaviour has set in. These remarks also apply to the other methods presented in this text. Apart from the accuracy and numerical-stability considerations, an important factor in molecular dynamics simulations is the calculation of the force acting on the particles. The integration steps require of order  $N$  operations. For two-body additive central forces one has to evaluate  $kN(N - 1)$  terms at each step. To reduce the computational complexity we can exploit the fact that most of the terms in the evaluation turn out to be zero if the potential has a cut-off. Only those terms where the particles are within the cut-off range  $r$ , give contributions. By choosing a suitable radius  $r$  we can ensure that only after  $n$  time steps does the number of particles inside this sphere change [3.3,7,48] (Problem 3.7). Hence, producing a list of nearest neighbours reduces the evaluation of the force term (???). Only those particles in the list



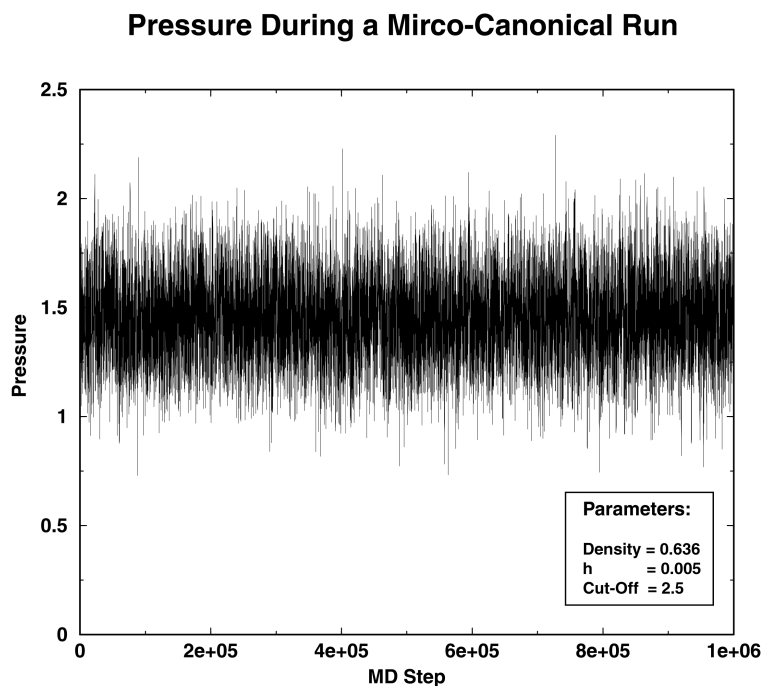
**Figure 1.5.** Evolution of the potential energy during a molecular dynamics simulation. All quantities are given in reduced units

of a given particle contribute. Every  $n$ th step the table must be updated. The trade-off is, of course, computer storage. The "Veriet table" has been used successfully on general purpose computers. For vector machines the technique has, however, drawbacks. A further discussion of time-saving techniques is deferred to the appendix.

### 1.1.2 Canonical Ensemble Molecular Dynamics

Various possibilities are discussed for a simulation with a constant temperature, instead of constant energy. Presented are the velocity scaling, iso-kinetic and damped force methods.

In the previous subsection we saw how the MD method solves the equations of motion numerically. The system under consideration was isolated, i.e., conservative, so the trajectory always stayed on a surface of constant energy in phase space. In many circumstances it is desirable to investigate a system along an isotherm rather than along a line of constant energy. Since the equations of motion allow propagation only on the constant-energy surface we have to modify the equations. The modification has to be such that the system will be conceptually coupled to a heat bath. The heat bath introduces the energy fluctuations which are necessary to keep a fixed temperature. Gen-

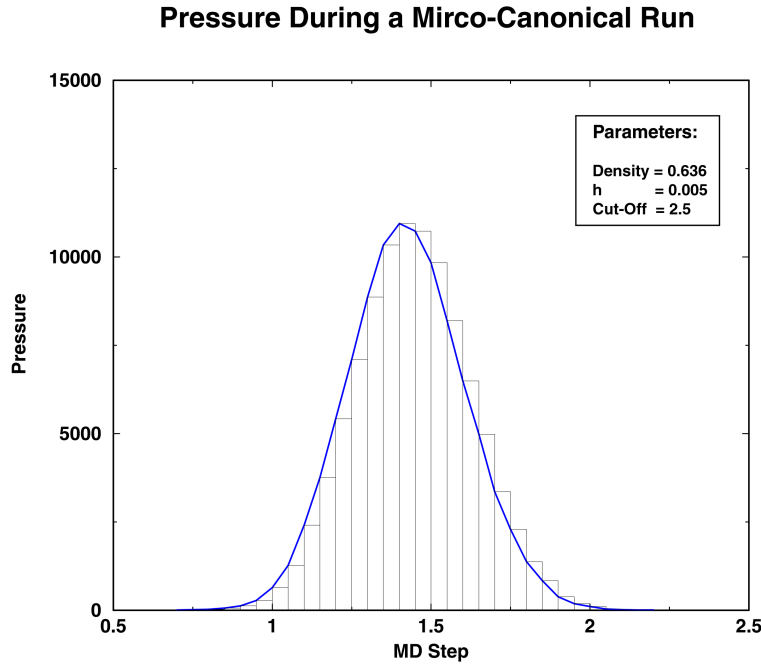


**Figure 1.6.** Evolution of the pressure during a molecular dynamics simulation. All quantities are given in reduced units

erally, observables appear as averages over an appropriate ensemble of similar systems. The appropriate ensemble here, representing equilibrium of a system in a heat bath, is the canonical ensemble, where the particle number  $N$ , the volume  $V$  and the temperature  $T$  are fixed, and there is zero total linear momentum  $P$  as well as a zero total angular momentum  $L$ . Since the total energy is not a conserved quantity for constant temperature, schemes have to introduce fluctuations in the total energy  $E$ . However, the average kinetic energy is a constant of motion due to its coupling with the temperature. Any scheme has to satisfy the requirement that the average properties computed along a trajectory must be equal to the ensemble average

$$\langle A \rangle_{NVT} = \lim_{t' \rightarrow \infty} \frac{1}{t' - t_0} \int_{t_0}^{t'} A(r^N(t), p^N(t); V(t)) dt \quad (1.49)$$

One way of achieving energy fluctuations for a constant temperature is to supplement the equations of motion with an equation of constraint. Alternatively one can add to the forces in the equations of motion a force of constraint (damped-force method) [3.50-55]. It can be shown [3.49] that the damped-force method is a special case of the constraint method. Another possibility is that of immersing the system in a heat bath by introducing



**Figure 1.7.** Evolution of the pressure during a molecular dynamics simulation. All quantities are given in reduced units

a stochastic force simulating collisions with virtual particles. Later we take up the idea of stochastic supplements to the equations of motion. A natural choice for the constraint is to fix the kinetic energy to a given value during the course of a simulation. Such a constraint may be the non-holonomic constraint [3.49]

$$A = \frac{1}{2} \sum_i m v_i^2 = \text{const} \quad (1.50)$$

(isokinetic MD) or one may take the total kinetic energy proportional to time with a vanishing proportionality constant if the system has reached a constant temperature (Gaussian isokinetic MD) [3.56]

$$\frac{1}{2} \sum_i m v_i^2 = \alpha t \quad (1.51)$$

Below, we shall adopt the iso-kinetic approach. Note that only the average temperature is fixed. We have already encountered a method to constrain the kinetic energy to a given value. To equilibrate the system, energy was drained or added by an ad hoc scaling of the velocities [3.47,57,58]. After reaching the desired energy or temperature the system was left to itself.



The functions  $g$  and  $gz$  denote the recursion relations. Note that  $gz$  can involve an additional dependence on the force at time  $n+1$ . In this case step 1 is introduced between 2 and 3. A bypass of step 5 is created after the equilibration phase. In the ad hoc velocity scaling method step 5 remains within the flow of the algorithm and scales the velocities at every time step. What is the appropriate scaling factor  $p$ ? The system has  $3N$  degrees of freedom. However, we require the system to have zero total linear momentum, so removing three degrees of freedom. The constraint of constant kinetic energy removes one more degree of freedom. Hence, the scaling factor

$$(3.46)$$

so that after the scaling step we have

Traditionally a weighting factor  $3N$  has been used instead of  $3N-4$ . The reason is that there are several problems associated with the procedure, making it non-exact, although, as we will see later, it is exact in its differential form. To reveal why it is not exact, let us study the ad hoc scaling within the leapfrog formulation of the Verlet algorithm (Problem 3.4)

$$(3.47)$$

Assume that the scaling factor was computed from the previous half-step velocity

$$(3.48)$$

cast into the following form, assuming a velocity form of the integration procedure (Algorithm A3 and Problem 3.4), i.e.,

do  $n = 1, \text{max}$  1. Compute the forces. 2. Compute  $r^{n+1} = g(r^n, v^n, F^n)$ . 3. Compute  $v^{n+1} = g(v^n, F^n, (F^{n+1}))$ . 4. Compute the kinetic energy. 5. Scale the velocities  $v^{n+1} = p v^{n+1}$ . time step loop end time step loop

The functions  $g$  and  $gz$  denote the recursion relations. Note that  $gz$  can involve an additional dependence on the force at time  $n+1$ . In this case step 1 is introduced between 2 and 3. A bypass of step 5 is created after the equilibration phase. In the ad hoc velocity scaling method step 5 remains within the flow of the algorithm and scales the velocities at every time step. What is the appropriate scaling factor  $p$ ? The system has  $3N$  degrees of freedom. However, we require the system to have zero total linear momentum, so removing three degrees of freedom. The constraint of constant kinetic energy removes one more degree of freedom. Hence, the scaling factor (3.46)

so that after the scaling step we have

Traditionally a weighting factor  $3N$  has been used instead of  $3N-4$ . The reason is that there are several problems associated with the procedure, making it non-exact, although, as we will see later, it is exact in its differential form. To reveal why it is not exact, let us study the ad hoc scaling within the leapfrog formulation of the Verlet algorithm (Problem 3.4)

$$(3.47)$$

Assume that the scaling factor was computed from the previous half-step velocity

$$(3.48)$$

(3.52)

yielding unconstrained equations of motion

(3.53)

with a new Lagrangian  $Z' = Z - V$ . To proceed we assume a simple form of  $V$  as a product of two functions

(3.54)

We imagine that  $p$  represents the mechanism of energy transfer between the system and the reservoir. The function  $\phi$  ensures the fulfilment of the constraint. The detailed mechanism has still to be specified. With (3.54) the equations of motion are calculated as

(3.55)

where  $p_i = \partial Z' / \partial \dot{r}_i$ . Let us now exploit the arbitrariness in the mechanism of energy transfer and assume that  $p$  is a function of the velocities only and that formally  $p$  is zero

(3.56)

The equations of motion reduce to

(3.57) An obvious choice for  $p$ , of course, is to take the constraint itself

(3.58)

so that the equations of motion become

(3.59)

We see that the introduction of energy fluctuations through a generalised potential with a specific choice of the detailed coupling leads to the velocity scaling mechanism. To constrain the kinetic energy a feedback loop is established. A problem arises when we discretize the differential equations. The discretization introduces a time delay in the feedback loop, leading to fluctuations in the average kinetic energy. Interestingly, the second possibility leads to a scaling mechanism, too. If one assumes a generalised force of the type

(3.60)

with the specific choice

(3.61)

one finds

(3.62)

For  $n = 1$  the equations of motion are those obtained by Hoover et al. [3.51-55]. In this case the equations of motion conform with the Gauss principle of least constraint [3.59,64]. Note that  $p$  does not reference the required temperature so that the initial conditions must be chosen in accordance with the constraint.

Example 3.2 In Example 3.1 (Sect.3.1.1) we studied a monatomic system consisting of 256 particles interacting with a Lennard-Jones potential. The simulation proceeded in such a way that energy was added or removed until a desired energy was achieved, corresponding to an average temperature. The energy remained constant during the rest of the simulation. To keep the temperature fixed and let the total internal energy fluctuate, the system must be brought in contact with a heat bath. One way of achieving this

is to constrain the kinetic energy to a fixed value during the course of the simulation.

Fig.3.7. Shown is the evolution of the (reduced) kinetic energy computed after the scaling of the minus half step velocities in the leap frog formulation of the Verlet algorithm. Time is given in molecular dynamics steps. In the text we discussed two possible implementations. The first was to rescale the "minus half-step velocities" of the leapfrog algorithm at every time step. Figure 3.7 shows the kinetic energy calculated after the scaling and after the "plus half-step velocities" were computed. Evidently the kinetic energy is not stationary. There are over- and under-shootings due to the time delay in the feedback loop. This is also the case if one applies the summed form of the Verlet algorithm. However, the fluctuations are somewhat less significant.

Table 3.2. Results from isokinetic molecular dynamic simulations for the reduced potential energy.  $r_c$  gives the cut-off of the Lennard-Jones potential

Fig.3.8. Reduced potential energy as a function of time (MD steps) observed in the isokinetic MD using the summed form algorithm

It is also instructive to observe the different behaviour in the relaxation of the potential energy (Fig.3.8). To compare the results for the constant energy with the iso-kinetic simulations, the system was prepared with exactly the same initial conditions for the positions and the velocities. A summary of the results is given in Table 3.2. A glance at this table shows that there is no observable difference for the high- and low-temperature states. The average potential energies, computed during the second half of altogether 2000 steps, agree within the statistical uncertainty. Hence such thermodynamic variables are unaffected by the scaling. It has been shown that the static properties computed along a trajectory indeed conform to the canonical ensemble averages [3.60-63]. Also the results from the above example bear out that static properties are invariant. The question neglected so far is how are the dynamic properties of the system influenced by any of the schemes? We may ask whether the transport properties are affected by the changes in the velocities imposed by the scaling. Clear-cut evidence from simulations does not exist yet. From an "experimental" point of view, possible effects due to the scaling are hard to disentangle from effects coming from the boundary condition imposed on the system. Furthermore, other finite-size effects are possible, not forgetting the potential cut-off. Also, analytically, no proof has been given that the algorithm becomes exact in the limit of infinite time.

### 1.1.3 Isothermal-Isobaric Ensemble Molecular Dynamics

Two algorithms are given for a simulation in which the temperature and the pressure are held constant.

A constant temperature and/or constant pressure molecular dynamics method is interesting not only from a theoretical point of view. The computation and the comparison of certain quantities with experimental observations sometimes require such an ensemble. One example is, of course, the specific

heat  $C$  at constant pressure. Let us begin with the constant pressure case. For the isolated  $N$ -particle system the energy  $E$  and the volume  $V$  are the independent variables. If we fix the pressure, then the volume, being the variable conjugate to the pressure, must be allowed to fluctuate. The system is not isolated anymore but in contact with the exterior. Assume that the transfer between the system and the exterior is adiabatic. In this situation, having a constant particle number  $N$  and constant pressure  $P$ , the total internal energy is not conserved. The conserved quantity is the enthalpy  $H$

$$H = E + P_E V \quad (1.52)$$

Here,  $P_E$  is the externally applied pressure. In mechanical equilibrium the external pressure and the internal pressure are equal. What we are dealing with is the isobaric-isoenthalpic ensemble  $(N, P, H)$ . As was the case for the canonical ensemble, we have to modify the equations of motion to allow for the constant pressure. Any modification has to be such that the average properties computed along the generated trajectory are those of the isobaric-isoenthalpic ensemble

$$\langle A \rangle_{NPH} = \lim_{t' \rightarrow \infty} \frac{1}{t' - t_0} \int_{t_0}^{t'} dt A(\underline{r}^N, \underline{v}^N; V(t)) \quad (1.53)$$

We shall keep a cubic volume and maintain periodic boundary conditions. In principle, shape fluctuations could be allowed [50, 51, 52, 53] which are important for crystal structures, but the formulae would become unduly complicated.

To make the volume fluctuations possible we introduce the volume  $V$  as a new dynamical variable. As such it is also assigned a mass  $M$ . To develop equations of motion for the particles and the volume we further take  $PV$  as the potential energy corresponding to the new dynamic variable [?] [3.59,65,66]. The Lagrangian now looks like

$$\mathcal{L}(\underline{r}, \dot{\underline{r}}, V, \dot{V}) = \frac{1}{2} \sum_i m \dot{\underline{r}}_i^2 - U(r) + \frac{1}{2} M \dot{V} + P_E V \quad (1.54)$$

Of course, the variables  $\underline{r}$  and  $V$  are not coupled. To proceed we appeal to intuition. If the system is subjected to pressure, the distances between the particles will change. Conversely, if the distances change, the pressure changes. The crucial step is the replacement of the coordinates  $\underline{r}_i$  of the particles by the scaled coordinates  $\underline{\rho}_i$  i.e.,

$$\underline{\rho}_i = \frac{\underline{r}_i}{V^{1/3}} = \frac{\underline{r}_i}{L} \quad (1.55)$$

Now all components of the position vectors of the particles are dimensionless numbers within the unit interval  $[0, 1]$ . With the transformation, the integrals of  $\underline{r}_i$  over the fluctuating volume  $V$  become integrals of  $\underline{\rho}_i$  over the unit cube. Having written down (1.55) we have made the implicit assumption

that each spatial point responds in the same way. Due to this, there is no consistent physical interpretation of the approach.

The equation (1.55) couples the dynamical variables  $\underline{r}$  to the volume. Taking the first time derivative we obtain

$$\dot{\underline{r}}_i = L\dot{\underline{\rho}}_i + \rho\dot{L} \quad (1.56)$$

In equilibrium, the changes in the volume can be regarded as slow. Therefore we may assume

$$\frac{p_i}{m} = L\dot{\underline{\rho}}_i \quad (1.57)$$

as the momentum conjugate to  $\rho_i$  and the Lagrangian becomes

$$\mathcal{L}(\underline{\rho}, \dot{\underline{\rho}}, V, \dot{V}) = \frac{1}{2}L^2 \sum_i m\dot{\underline{\rho}}_i^2 - U(L\rho) + \frac{1}{2}M\dot{V}^2 - P_E V \quad (1.58)$$

Recall that we anticipate possible effects on the intrinsic dynamics when we modify the equations. However, the static properties should not be affected. Concerning this point, note that the potential energy does not involve the new coordinates  $\underline{\rho}$  but the true  $\underline{r}$ . In a somewhat loose way the Hamiltonian of the system is formulated as [3.65,66]

$$\mathcal{H} = \frac{1}{2}L^2 \sum_i m\dot{\underline{\rho}}_i^2 + U(L\rho) + \frac{1}{2}M\dot{V}^2 + P_E V \quad (1.59)$$

Here  $M$  is still a free parameter, about which we will have more to say later. Having set up the Hamiltonian the next task is to derive the equations of motion for the particles and the volume. These equations will now be coupled. In the Newtonian formulation they are

$$\begin{aligned} \frac{d^2 \dot{\underline{\rho}}_i}{dt^2} &= \frac{F_i}{mL} - \frac{2}{3} \dot{\underline{\rho}}_i \left( \frac{\dot{V}}{V} \right), \\ \frac{d^2 V}{dt^2} &= \frac{P - P_E}{M} \end{aligned} \quad (1.60)$$

with the pressure  $P$  computed from the virial

$$P = \frac{1}{3L} \left[ \sum_i m\dot{\underline{\rho}}_i^2 + \sum_{i<j} r_{ij} F_{ij} \right] \quad (1.61)$$

These equations yield a constant average pressure. We have two frames to consider. The first is the frame with the original coordinates. The calculation of the forces, the energy and other structural quantities must be carried out

in this frame. The second frame with the scaled coordinates is needed for the evolution of the system.

For (1.60 and ??) we can immediately write down an algorithm. What is needed are only minor modifications of the summed form algorithm. There is, however, a problem due to the appearance of the first derivative of the position on the right-hand side of (1.60). Recall that the algorithm was developed for equations of the form

$$\frac{d^2 r}{dt^2} = f(r) \quad (1.62)$$

Assuming that the algorithm is still numerically stable with the inclusion of a first derivative, i.e. a velocity, on the right-hand side, we obtain for the positions and the volume at time  $n + 1$

$$\begin{aligned} \underline{\rho}^{n+1} &= \underline{\rho}^n + h\underline{\dot{\rho}}^n + \frac{1}{2} \frac{h^2 F^n}{mL^n} - \frac{1}{2} h^2 \underline{\rho}^n \frac{\dot{V}^n}{V^n} \\ V^{n+1} &= V^n + h\dot{V}^n + \frac{1}{2M} h^2 (P^n - P_E) \end{aligned} \quad (1.63)$$

To compute the velocities and the volume velocity we take first the partial velocities

$$\begin{aligned} h\underline{\dot{\rho}}^{n+1} &= h\underline{\dot{\rho}}^n + \frac{1}{2} \frac{h^2 F^n}{mL^n} - \frac{1}{2} h^2 \underline{\dot{\rho}}^n \frac{\dot{V}^n}{V^n} \\ h\dot{V}^{n+1} &= h\dot{V}^n + \frac{1}{2M} h^2 (P^n - P_E) \end{aligned} \quad (1.64)$$

The next step is to compute  $\frac{1}{2} h^2 F^{n+1}$  and  $\sum r_{ij} F(r_{ij})$ . At this stage another problem presents itself. To compute the pressure at the  $(n + 1)$ th step the velocities of the  $(n + 1)$ th step are required! To circumvent the computation of an extrapolation we simply take the partial velocities to estimate the kinetic energy. Using this approximation the velocities are

$$\begin{aligned} h\dot{V}^{n+1} &= h\dot{V}^{n+1} + \frac{1}{2M} h^2 (P^{n+1} - P_E) \\ h\underline{\dot{\rho}}^{n+1} &= h\underline{\dot{\rho}}^{n+1} + \frac{1}{2} \frac{h^2 F^{n+1}}{mL^{n+1}} - \frac{1}{2} h^2 \underline{\dot{\rho}}^{n+1} \frac{\dot{V}^{n+1}}{V^{n+1}} \end{aligned} \quad (1.65)$$

Note that there is no rigorous proof for the validity of the procedure. Let us formulate the algorithm developed as:

#### Algorithm A5. NPH Molecular Dynamics

1. Specify the initial positions and velocities.

2. Specify an initial volume  $V^0$  consistent with the required density.
3. Specify an initial velocity for the volume, for example  $\dot{V} = 0$ .
4. Compute  $\overline{\rho^{n+1}}$  and  $V^{n+1}$  according to (1.63).
5. Compute the partial velocities for the particles and the volume according to (1.64).
6. Compute the forces and the potential part of the virial.
7. Compute the pressure  $P^{n+1}$  using the partial velocities.
8. Compute the volume velocity.
9. Compute the particle velocities using the partial velocities.

We shall investigate the algorithm in the following example.

### Example 4.3

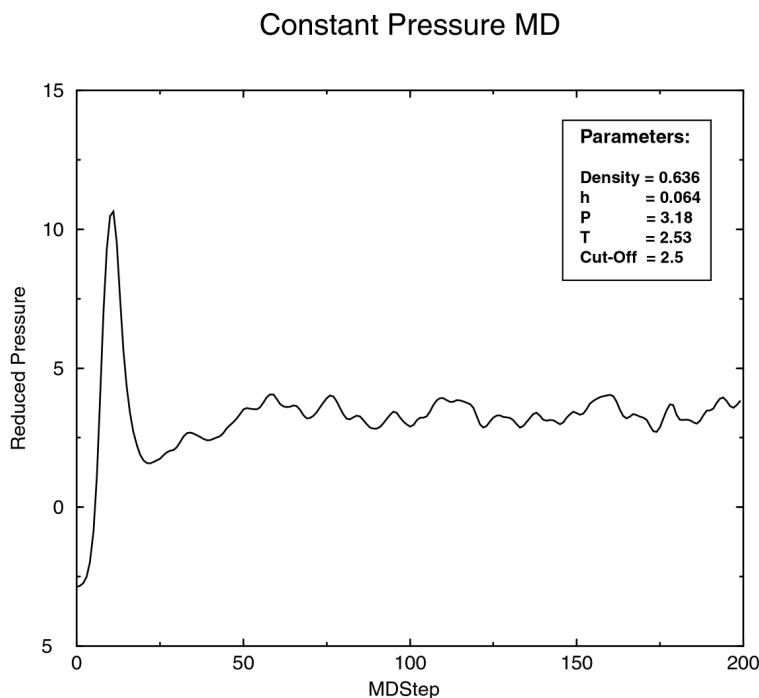
As a system to test the Algorithm A5 we choose again argon with  $N = 256$  particles and a potential cut-off at  $r_c = 2.5$ . The initial conditions for the positions and the velocities are identical to those in the previous examples. As the reference temperature we take  $T^* = 2.53$  and the initial density  $p^* = 0.636$ . To equilibrate the system energy, i.e., to arrive at the reference temperature, all velocities are rescaled every 50th step.

We now have to consider the choice of the mass  $M$ . Notice from Fig 1.8 that the initial pressure is negative. Hence the initial conditions are such that the system would like to contract. A negative pressure is not unphysical since the initial conditions, in general, do not correspond to equilibrium. At equilibrium the pressure has to be positive. On choosing a mass which is too small, the contraction results in a catastrophic overshooting of the volume. A similar observation was made by Smith [3.67]. In the particular examples depicted in Figs ?? the mass  $M^*$  is 0.01 (notice that the mass  $M$  is a reduced mass;  $M^* = M\rho^4/m$ ). On the other hand, if the mass is too large the system develops long-wavelength fluctuations in the volume [3.66]. In the case studied here the system also shows fluctuations extending over many MD steps. This is, indeed, expected for any finite  $M$  [3.65]. The value of  $M$  determines the time scale for the volume fluctuations.

The relaxation behaviour of the pressure is interesting (Fig. 1.8). The initial large fluctuations decay very rapidly. Looking at the average reduced pressure, i.e., the average as a function of time, we see that the settling to a constant pressure sets in very early. However, there are still fluctuations.

The magnitude of these depends on the chosen mass  $M^*$  [3.66].

In the NPH molecular-dynamics algorithm there is one free parameter  $M$ . In the example we saw that its magnitude influences the relaxation to equilibrium. Not only are the pressure and the volume affected but also the kinetic energy [3.66] (note that in equilibrium the pressure relaxes more rapidly than the temperature [3.19]). Unfortunately there is no criterion available for an appropriate choice. Indeed, it is difficult to develop such a criterion. As seen in the example,  $M$  must depend on the precise initial conditions. Furthermore, it is not yet established how far dynamical properties of the system, such as



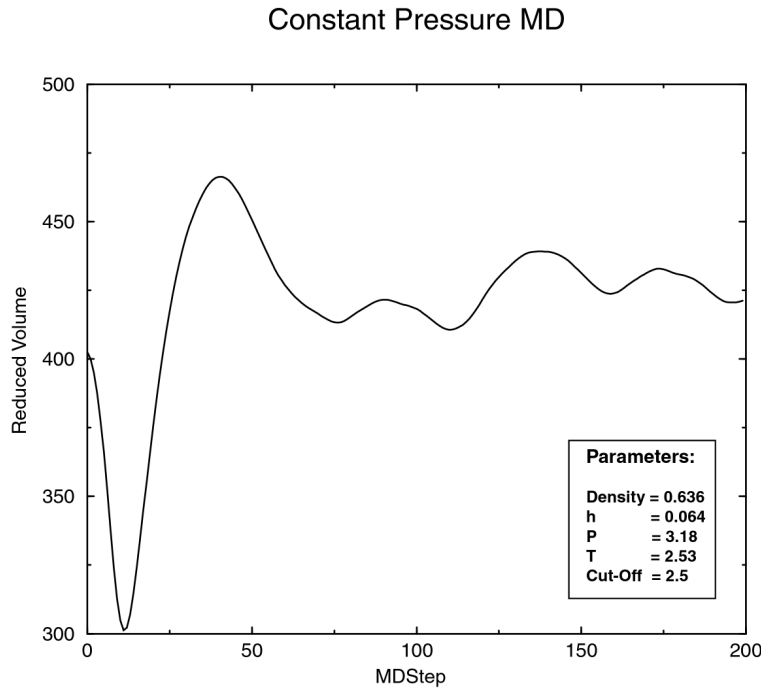
**Figure 1.8.** The right figure shows how the internal pressure relaxes towards the value given by the external pressure. In the left plot is shown the average of the pressure

transport coefficients, are affected by the magnitude of  $M$ . Static properties are independent of  $M$  [3.65,66]. To keep the impact as small as possible  $M$  has to be small. It is therefore desirable to have an algorithm which changes  $M$  from step to step. Initially  $M$  should be large, to compensate negative pressures, and gradually decrease as the system equilibrates. It follows that  $M$  should be coupled to the pressure difference. Up to now we have considered an ensemble where the particle number, the pressure and the enthalpy are the independent thermodynamic variables. Such an isobaric-isoenthalpic ensemble is rather unusual and instead of a constant enthalpy we introduce now a constant temperature.

We can achieve a constant temperature during the MD simulation by rescaling all the velocities at every time step. This constrains the kinetic energy to a fixed value and gives a desired temperature (Algorithm A4). The same idea can be used also for the  $(N, P, T)$ -ensemble algorithm. We combine the algorithm described above for constant pressure with velocity scaling.

Algorithm A6. NPT Molecular Dynamics





**Figure 1.9.** Shown is the evolution of the volume during the initial phase of a constant pressure MD simulation

1. As in Algorithm A5.
2. Rescale all velocities as in step 8 of Algorithm A5.

Indeed, Algorithm A6 requires only a trivial change, since the only modification one has to make is to proceed with the equilibration phase, with the scaling being performed at each step, until the end of the simulation.

## 1.2 Density Functional Method

### 1.3 Problems

- 3.1 The truncated octahedron boundary condition [3.68,69] is obtained by cutting off the corners of a cube of side  $2A$  until half its volume 253.83
- 3.11. Total internal energy  $E'$  as a function of time (MD-steps) reduced total energy -Z83.76 liD - Step ees
- In Sect.3.1.2 we achieved a constant temperature during the MD simulation by rescaling all the velocities at every time step. This constrains the kinetic energy to a fixed value and gives a desired temperature (Algorithm

A4). The same idea can be used also for the (N, P, T) ensemble algorithm. We combine the algorithm described above for constant pressure with velocity scaling. Algorithm A6. NPT Molecular Dynamics 1-9. As in Algorithm A5. 10. Rescale all velocities as in step 8 of Algorithm A5. Indeed, Algorithm A6 requires only a trivial change, since the only modification one has to make is to proceed with the equilibration phase, with the scaling being performed at each step, until the end of the simulation. Example 3.4 The conditions in this example are exactly the same as in the preceding one. Instead of performing the scaling at every 50th step, the scaling is done every step. In Fig.3.11 the total internal energy  $E' = E + U'$  is shown. As in the example of Sect.3.1.2 the energy relaxes very quickly. A quick relaxation is also seen in the pressure (Fig.3.12) and in the volume (Fig.3.13). O

- The truncated octahedron boundary condition [3.68,69] is obtained by cutting off the corners of a cube of side  $2A$  until half its volume 253.83

## References

1. D. Stauffer, *Introduction to Percolation Theory*, Taylor and Francics, 1985
2. J. Hoshen, R. Kopelman, *Phys. Rev. B* **14**, 3428 (1976)
3. M. Eden, in *Proc. 4th Berkeley Symp. Math. Statis. and Probability*, ed. by F. Neyman (Univ. California Press, Berekeley 1960) Vol. 4
4. D. Knuth, *The Art of Computer Programming*, Vol. 2, Wiley, New York, 1979
5. F. James, *Comput. Phys.* **60**, 329-344 (1990)
6. R.C. Tausworth, *Math Comput.* **19** 201 (1965)  
T.G Lewis and W.H. Payne *JACM*, **20**, 456-468 (1973)  
S. Kirkpatrick and E.P. Stoll, *J. Comput. Phys.* **40**, 517-526 (1981)
7. S.K. Park, K.W. Miller *Comm. ACM* **31**, 1192 (1988)  
T.G. Lewis, A.S. Goodman und J.M. Miller *IBM Sys. J.* **8**, 136 (1969)  
S.L. Anderson, *SIAM Review* **32**, 221 (1990)  
P. L'Ecuyer, *Comm. ACM* **33**, 86 (1990)
8. R.W. Hamming: *Numerical Methods for Scientists and Engineers*
9. *Collected Algorithms of the ACM*
10. J. von Neumann: *Various Techniques Used in Connection with Random Digits, Collected Works*, Vol. 5 (Pergamon, New York 1963)
11. G. Marsaglia, B. Narasimhan und A. Zaman, *Comp. Phys. Commun.* **60**, 345 (1990)
12. D.H. Lehmer, *Proc. 2nd Symposim on Large-Scale Digital Computing Machinery*, 142, Harvard University Press, Cambridge, 1951
13. R.C. Tausworth, *Math Comput.*, **19**, 201 (1965)
14. G.E. Forsythe: *Math. Comput.* **26**, 817 (1972)
15. G.E.P. Box, M.E. Muller, G. Marsaglia: *Ann. Math. Stat.* **28**, 610 (1958)
16. J. Lach: Unpublished, 1962
17. E. Friedberg, J.E. Cameron; *J. Chem. Phys.* **52**, 6049 (1970)
18. E.B. Smith, B.H. Wells: *Mol. Phys.* **52**, 709 (1984)
19. D. Finchman: Daresbury Lab. *Information Quaterly for Computer Simulation of Condensed Phases***17**, 43 (1985)
20. K. Binder, D.W. Heermann: *Monte Carlo*, in *Statistical Physics*, Springer Ser. Solid-State Sci., Vol .80 (Springer, Berlin, Heidelberg 1988)
21. W.G. Hoover. *Molecular Dynamics* (Springer, Berlin, Heidelberg 1986)
22. M.H. Kalos, P.A. Whitlock: *Monte Carlo Methods*, Vol.1 (Wiley, New York 1986)
23. D. Stauffer, F.W. Hehl, V. Winkelmann, J.G. Zabolitzky: *Computational Physics* (Springer, Berlin, Heidelberg 1988)
24. D. Rapaport *Comp. Phys. Rep.* 9, I (1988)
25. J.L. Lebowitz, J.K. Percus: *Phys. Rev.* **24**, 1673 (1961)
26. J.L. Lebowitz, J.K. Percus, L. Verlet *Phys. Rev* **124**, 1673 (1967)
27. B.J. Alder, T.E. Wainwright: *J. Chem. Phys.* **27**, 1208 (1957)
28. T.E. Wainwright, B.J. Alder. *Nuovo cimento* **9**, Suppl. Sec. 116 (1958)

29. B.J. Alder, T.E. Wainwright J. Chem. Phys. **31**, 456 (1959)
30. B.J. Alder, T.E. Wainwright J. Chem. Phys. **33**, 1439 (1960)
31. J.R. Beeler, Jr.: In *Physics of Many-Particle Systems*, ed. by C. Meeron (Gordon and Breach, New York 1964)
32. A. Rahman: Phys. Rev. **136**, A405 (1964)
33. L. Verlet: Phys. Rev. **159**, 98 (1967)
34. K. Binder, M.H. Kalos: In *Monte Carlo Methods in Statistical Physics*, 2nd edn., ed. by K. Binder, Topics Curr. Phys., Vol.7 (Springer, Berlin, Heidelberg 1986)
35. J.P. Ryckaert, G. Ciccotti, H.J.C. Berendsen: J. Comput. Phys. **23**, 327 (1977)
36. G. Ciccotti, M. Ferrario, J.P. Ryckaert Mol. Phys. **47**, 1253 (1982)
37. J.P. Ryckaert, G. Ciccotti: J. Chem. Phys. **78**, 7368 (1983)
38. W.F. van Gunsteren, H.J.C. Berendsen: Mod. Phys. **34**, 131 l (1977)
39. D.J. Tidlesley, P.A. Madden: Mol. Phys. 42, I 137 (1981)
40. R.M. Stratt, S.L. Holmgreen, D. Chandler. Mol. Phys. 46,1233 (1981)
41. M.K. Memon, R.W. Hockney, S.K. Mitrx J. Comput. Phys. 43, 345 (1982)
42. S. Nose, M.L. Klein: Mol. Phys. 50, 1055 (1983)
43. H.C. Anderson: J. Comput. Phys. **52** , 24 (1983)
44. L.D. Landau, E.M. Lifshitz: *Statistical Physics* Vol.5, 3rd edn. (Pergamon, Oxford 1980) p.42
45. D.W. Heermann and A.N. Burkitt, **Parallel Algorithms of Computational Science Problems** Springer Verlag, Heidelberg 1990
46. W.W. Wood, F.R. Parker. J. Chem. Phys. 27, 720 (1957)
47. J.N. Shaw: Monte Carlo calculations for a system of hard-sphere ions, PhD Thesis, Duke University (1963)
48. P.P. Ewald: Ann. Physik 64, 253 (1921)
49. S.G. Brush, H.L. Sahlín, E. Teller. J. Chem. Phys. 45, 2102 (1966)
50. M. Parinello, A. Rahman: Phys. Rev. Lett. 45, l 196 (1980)
51. M. Parinello, A. Rahman: J. Appl. Phys. 52, 7182 (1981)
52. M. Parinello, A. Rahman: J. Chem. Phys. 76, 2662 (1982)
53. M. Parinello, A. Rahman, P. Vashishtrix Phys. Rev. Lett. 50, 1073 (1983)
54. G. Dahlquist, A. Björck: *Numerical Methods* (Prentice Hall, Englewood Cliffs, NJ 1964)
55. J. Stoer, R. Bulirsch: *Einführung in die Numerische Mathematik II* (Springer, Berlin, Heidelberg 1973)
56. A. Nordsieck: Math. Comput. 16, 22 (1962)
57. C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice Hall, Englewood Cliffs, NJ 1971)
58. C.W. Gear. Report ANL 7126, Argonne National Lab. (1966)
59. C.W. Gear. Math. Comput. 21, 146 (1967)
60. D. Beeman: J. Comput. Phys. 20, 130 (1976)
61. S. Toxvaerd: J. Comput. Phys. 47, 444 (1982)
62. A. LaBudde, D. Greenspan: J. Comput. Phys. 15, 134 (1974)
63. D. Greenspan: J. Comput. Phys. 56, 28 (1984)
64. K. Kanatani: J. Comput. Phys. 53, 181 (1984)
65. K. Aizu: J. Comput. Phys. 55, 154 (1984)
66. K. Aizu: J. Comput. Phys. 58, 270 (1985)
67. J.H. Wilkinson: *Rounding Errors in Algebraic Processes* (HMSO, London 1963)
68. O. Penrose, J.L. Lebowitz: In *Studies in Statistical Mechanics*, Vol.7, ed. by J.L. Lebowitz, E.W. Montroll (North-Holland, Amsterdam 1979)
69. D.W. Heermann, W. Klein, D. Stauffer. Phys. Rev. Lett. 49, 1262 (1982)
70. W.C. Swope, H.C. Andersen, P.H. Berens, K.R. Wilson: J. Chem. Phys. 76, 637 (1982)

71. L.V. Woodcock: Chem. Phys. Lett. 10, 257 (1970)
72. B. Quentrec, C. Brot J. Comput Phys. 1, 430 (1973)
73. J.M. Haile, S. Gupta J. Chem. Phys. 79, 3067 (1983)
74. W.G. Hoover, D.J. Evans, R.B. Hickman, A.J.C. Ladd, W.T. Ashurst, B. Moran: Phys. Rev. A 22, 1690 (1980)
75. W.G. Hoover. Physica A 18, 111 (1983)
76. W.G. Hoover. In Nonlinear Fluid Behaviour, ed. by H.J. Hanley (North-Holland, Amsterdam 1983)
77. W.G. Hoover, A.J.C. Ladd, B. Moran: Phys. Rev. Lett. 48, 3297 (1983)
78. D.J. Evans, G.P. Morriss: Chem. Phys. 77, 63 (1983)
79. D.J. Evans: J. Chem. Phys. 78, 3297 (1983)
80. D.M. Heyes, D.J. Evans, G.P. Morriss: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 17, 25 (1985)
81. F. van Swol, L.V. Woodcock, J.N. Cape: J. Chem. Phys. 75, 913 (1980)
82. J.Q. Broughton, G.H. Gilmer, J.D. Weeks: J. Chem. Phys. 75, 5128 (1981)
83. D. Brown, J.H.R. Clarke: Mol. Phys. 51, 1243 (1984)
84. S. Nose: Mol. Phys. 52, 255 (1984)
85. S. Nose, M.L. Klein: Mol. Phys. 50, 1055 (1983)
86. S. Nose: J. Chem. Phys. 81, 511 (1984)
87. A. DiNola, J.R. Haak: J. Chem. Phys. 81, 3684 (1984)
88. J.R. Ray: Am. J. Phys. 40, 179 (1972)
89. H.C. Andersen: J. Chem. Phys. 72, 2384 (1980)
90. J.M. Haile, H.W. Graben: J. Chem. Phys. 73, 2412 (1980)
91. D. Adams: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 10, 30 (1983)
92. D. Fincham: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 12, 43 (1984)
93. W. Feller: An Introduction to Probability Theory and Its Applications (Wiley, New York 1968)
94. G.R. Grimmett, D.R. Stirzaker: Probability and Queuing Processes (Oxford Univ. Press, Oxford 1987)
95. J.G. Kemeny, J.L. Snell: Finite Markov Chains (Springer, Berlin, Heidelberg 1976)
96. K. Kremer, K. Binder: Comput. Phys. Rep. 7, 259 (1988)
97. P. Turq, F. Lantelme, H.L. Friedman: J. Chem. Phys. 66, 3039 (1977)
98. G.S. Grest, K. Kremer: Phys. Rev. Lett. 33, 3628 (1986)
99. S. Chandrasekhar: Rev. Mod. Phys. 15, I (1943)
100. M.C. Wang, G.E. Uhlenbeck: Rev. Mod. Phys. 17, 323 (1945)
101. A. Ludwig: Stochastic Differential Equations: Theory And Applications (Wiley, New York 1974)
102. S.O. Rice: Bell Syst. Tech. J. 23, (1946)
103. J.L. Doob: Ann. Math. 43, 351 (1942)
104. J.L. Doob: Ann. Am. Stat. 15, 229 (1944)
105. D.L. Ermak: J. Chem. Phys. 62, 4189 (1974)
106. D.L. Ermak: J. Chem. Phys. 62, 4197 (1974)
107. J.D. Doll, D.R. Dion: J. Chem. Phys. 65, 3762 (1976)
108. T. Schneider, E. Stoll: Phys. Rev. B 13, 1216 (1976)
109. T. Schneider, E. Stoll: Phys. Rev. B 17, 1302 (1978)
110. W.F. van Gunsteren, H.J.C. Berendsen: Mol. Phys. 45, 637 (1982)
111. M.P. Allen: Stochastic Dynamics (CECAM Workshop Report 1978)
112. M.P. Allen: Mol. Phys. 40, 1073 (1980)
113. M.P. Allen: Mol. Phys. 47, 599 (1982)

114. H.J.C. Berendsen, J.P.M. Postma, W.F. van Gunsteren, A. DiNola, J.R. Haak: J. Chem. Phys. 81,3684 (1984)
115. H.C. Andersen: J. Chem. Phys. 72,2384 (1980)
116. A. Greiner, W. Strittmatter, J. Honerkamp: J. Stat. Phys. 51,95 (1988)
117. B. Dnweg, W. Paul: To be published
118. J.H. Halton: SIAM Rev. 12,4 (1970)
119. J.M. Hammersley, D.C. Handscomb: Monte Carlo Methods (Chapman & Hall, New York 1964)
120. A. Papoulis: Probability, Randoom Variables And Stochastic Processes (McGraw-Hill, Tokyo 1965)
121. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller: J. Chem. Phys. 21,1087 (1953)
122. H. Mller-Krumbhaar, K. Binder: J. Stat. Phys. 8, I (1973)
123. K. Binder: Adv. Phys. 23,917 (1974)
124. K. Binder: In Phase Transitions and Critical Phenomena, ed. by C. Domb, M.S. Green (Academic, New York 1976)
125. L.D. Fosdick: Methods Comput. Phys. 1,245 (1963)
126. I.Z. Fisher: Sov. Phys.-Usp. 2,783 (1959)
127. K. Binder: In Monte Carlo Methods in Statistical Physics, 2nd ed., ed. by K. Binder, Topic Curr. Phys., Vol.7 (Springer, Berlin, Heidelberg 1986)
128. K. Kawasaki: In Phase Transitions and Critical Phenomena, Vol.2, ed. by C. Domb, M.S. Green (Academic, New York 1972)
129. W.W. Wood: In Physics Of Simple Liquids, ed. by H.N.V. Temperley, J.S. Rushbrook (North-Holland, Amsterdam 1968)
130. H.E. Stanley: Introduction to Phase Transitions and Critical Phenomena (Oxford Univ. Press, London 1971)
131. O.G. Mouritsen: Computer Studies Of Phase Transitions and Critical Phenomena (Springer, Berlin, Heidelberg 1984)
132. T.L. Hill: Thermodynamics of Small Systems (Benjamin, New York 1963)
133. G.S. Pawley, G.W. Thomas: Phys. Rev. Lett. 48,410 (1982)
134. R.G. Palmer: Adv. Phys. 31,669 (1982)
135. D. Stauffer: Private communication
136. A.E. Ferdinand, M.E. Fisher: Phys. Rev. 185,832 (1969)
137. P.C. Hohenberg, B.I. Halperin: Rev. Mod. Phys. 49,435 (1977)
138. K. Binder, D. Stauffer: Adv. Phys. 25,343 (1976)
139. K. Binder: J. Stat. Phys. 24,69 (1981)
140. E. Stoll, K. Binder, T. Schneider: Phys. Rev. B 8,3266 (1973)
141. N. Y. Choi, B.A. Huberman: Phys. Rev. B 29,2796 (1984)
142. K. Binder: J. Comput. Phys. 59, I (1985)
143. B.K. Chabrabarty, H.G. Baumgrtel, D. Stauffer: Z. Phys. B 44, 333 (1981)
144. C. Kalle: J. Phys. A 17, (1984)
145. M. Creutz: Phys. Rev. Lett. 50, 1411 (1983)
146. G. Bahnot, M. Creutz, H. Neuberger: Nucl. Phys. B 235 [FSII], 417 (1984)
147. D.W. Heermann, R.C. Desai: Comput. Phys. Commun. 50,297 (1988)
148. R.C. Tolman: The Principles Of Statistical Mechanics (Dover, New York 1979)
149. E. Ising: Z. Phys. 31,253 (1925)
150. K. Binder, H. Mller-Krumbhaar: Phys. Rev. B 7,3297 (1973)
151. J.W. Cahn: J. Chem. Phys. 66,3667 (1977)
152. K. Binder, D.P. Landau: J. Appl. Phys. 57,3306 (1985)
153. R.J. Glauber: J. Math. Phys. 4,294 (1963)
154. W. Paul, D.W. Heermann, K. Binder: J. Phys. A 22,325 (1989)
155. M.N. Barber: In Phase Transitions And Critical Phenomena, ed. by C. Domb. J.L. Lebowitz (Academic, New York 1983)

156. D.P. Landau: Phys. Rev. B 13,2997 (1967)
157. R.H. Swendsen, S. Krinsky: Phys. Rev. Lett. 43,177 (1979)
158. K. Binder, D.W. Heermann: The Monte Carlo Method in Statistical Physics, Springer Ser. Solid-State Sci., Vol.80 (Springer, Berlin, Heidelberg 1988)
159. A.B. Bortz, M.H. Kalos, J.L. Lebowitz: J. Comput. Phys. 17, 10 (1975)
160. A. Sadiq: J. Comput. Phys. 55, 387 (1984)
161. Z.W. Salsburg, J.D. Jacobsen, W. Fickett, W.W. Wood: J. Chem. Phys. 30, 65 (1959)
162. J.P. Hansen, L. Verlet Phys. Rev. 184,151 (1969)
163. K. Binder: J. Stat. Phys. 24,69 (1981)
164. K. Binder: Z. Phys. B 45,61 (1981)
165. T.L. Polgreen: Phys. Rev. B 29,1468 (1984)
166. S.K. Ma: J. Stat. Phys. 26,221 (1981)
167. Z. Alexandrowicz: J. Chem. Phys. 55,2765 (1971)
168. H. Meirovitch, Z. Alexandrowicz: J. Stat. Phys. 16,121 (1977)
169. H. Meirovitch: Chem. Phys. Lett. 45,389 (1977)
170. H. Meirovitch: J. Stat. Phys. 30681 (1983)
171. J.P. Valleau, D.N. Cord: J. Chem. Phys. 57,5457 (1972)
172. G. Torrie, J.P. Valleau: Chem. Phys. Lett. 28,578 (1974)
173. G. Torrie, J.P. Valleau: J. Comput. Phys. 23, 187 (1977)
174. K. Kawasaki, T. Imaeda, J.D. Gunton: In Perspectives in Statistical Physics, ed. by H.J. Raveche (North-Holland, Amsterdam 1981)
175. J.D. Gunton, M. San Miguel, P. Sahní: In Phase Transitions and Critical Phenomena, Vol.8, ed. by C. Domb, J.L. Lebowitz (Academic, New York 1985)
176. L.P. Kadanoff: Physics 2,263 (1966)
177. B. Freedman, P. Smolensky, D. Weingarten: Phys. Lett. 113B, 481 (1982)
178. A. Milchev, D.W. Heermann, K. Binder: J. Stat. Phys. 44,749 (1986)
179. R. Swendsen, Wang: Phys. Rev. Lett. 50,297 (1988)
180. U. Wolff: Phys. Rev. Lett. 50,297 (1988)
181. A.N. Burkitt, D.W. Heermann: to be published
182. C.M. Fortuin, P.W. Kastelyn: Physica 57,536 (1972)
183. M. DeMeo, D. W. Heermann, K. Binder: J. Stat. Phys. in press
184. J.D. Doll, R.D. Coalsen, D.L. Freeman: Phys. Rev. Lett. 55, I (1985)
185. S. Duane, A.D. Kennedy, B.J. Pendelton, D. Roweth: Phys. Rev. Lett. 2, 195 (1987)
186. G.E. Norman, V.S. Filinov: Hgh Temp. (USSR) 7,216 (1969)
187. J.P. Valleau, L.K. Cohen: J. Chem. Phys. 72,5935 (1980)
188. D.J. Adams: Mol. Phys. 28, 1241 (1974)
189. D.J. Adams: Mol. Phys. 29,307 (1975)
190. L.A. Rowley, D. Nicholson, N.G. Parsonage: J. Comput. Phys. 17,401 (1975)
191. L.A. Rowley, D. Nicholson, N.G. Parsonage: Mol. Phys. 31,365 (1976)
192. J. Yao: PhD Dissertation, Purdue University (1981)
193. J. Yao, R.A. Greenkorn, K.C. Chao: Mol. Phys. 46,587 (1982)
194. A.Z. Panagiatopoulos: Mol. Phys. 61,813 (1987)
195. A.Z. Panagiatopoulos, N. Quirke, M. Stapleton, D. Tildesley: Mol. Phys. 63, 527 (1988)
196. H. Herrmann: J. Stat. Phys. 45, 145 (1986)
197. G. Grinstein, C. Jayaprakash, Y. He: Phys. Rev. Lett. 55,2527 (1987)
198. Kauffman: Phys. Rev. Lett. (1987)
199. M. Rovere, D.W. Heermann, K. Binder: Europhys. Lett. 6,585 (1988)
200. R.P. Feynman and A.R. Hibbs, *Quantum Mechanics and Path Integrals*, McGraw-Hill, New York, 1965
201. R.P. Feynman, *Statistical Mechanics*, Reading, Mass., 1972

202. T. Morita, *J. Phys. Soc. Jpn* **35**, 98 (1973)
203. J.A. Barker, *J. Chem. Phys.* **70**, 2914 (1979)
204. M.F. Herman, E.J. Bruskin, and B.J. Berne, *J. Chem. Phys.* **76**, 5150 (1982)
205. D. Thirumalai, R.W. Hall, and B.J. Berne, *J. Chem. Phys.* **81**, 2523 (1984)
206. B.J. Berne, *J. Stat. Phys.* **43**, 911 (1986)
207. J.D. Doll, D.L. Freeman, and T.L. Beck, *Adv. Chem. Phys.* **LXXVIII**, 61 (1990)
208. S. Duane, A. D. Kennedy, B. J. Pendleton and D. Roweth, *Phys. Lett. B* **195**, 216 (1987)
209. M. Creutz, *Phys. Rev. D* **38**, 1228 (1988)
210. A. D. Kennedy, *Nucl. Phys. (Proc. Suppl.)* **4**, 576 (1988)
211. A. D. Kennedy, *Nucl. Phys. (Proc. Suppl.)* **9**, 457 (1989)
212. S. Gupta, A. Irbäck, F. Karsch and B. Petersson, *Phys. Lett. B* **242**, 437 (1990)
213. R. Gupta, G. W. Kilcup and S. R. Sharpe, *Phys. Rev. D* **38**, 1278 (1988)
214. A. D. Kennedy and B. J. Pendleton, Edinburgh preprint 90/498 (1990)
215. D.W. Heermann, P. Nielaba and M. Rovere, *Comput. Phys. Commun.* **60**, 311 (1990)
216. B. J. Pendleton (private communication)
217. D. W. Heermann, *Computer Simulation Methods in Theoretical Physics*, Springer-Verlag, Heidelberg 1990
218. K. Binder and D. W. Heermann, *Monte Carlo Simulation in Statistical Physics*, Springer-Verlag, Heidelberg 1988
219. K. Kremer, In: *Computersimulation in der Physik*, KFA Jülich, Jülich 1989
220. J. J. Nicolas, K. E. Gubbins, W. B. Streett and D. J. Tildesley, *Mol. Phys. A* **5**, (1978) 1429
221. M. Creutz and A. Goksch, *Phys. Rev. Lett.* **63**, 9 (1989)
222. M.P. Allen and D.J. Tildesley, *Computer Simulations of Liquids*, Clarendon Press, Oxford 1987
223. A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* **61**, 2635 (1988)
224. A. M. Ferrenberg and R. H. Swendsen, *Phys. Rev. Lett.* **63**, 1195 (1989)
225. M. Falcioni, E. Marinari, M. L. Paciello, G. Parisi and B. Taglienti, *Phys. Lett. B* **108**, 331 (1982)
226. P. Peczak, A. M. Ferrenberg and D. P. Landau, University of Georgia preprint (1991)
227. E. Brézin, J. C. Le Guillou and J. Zinn-Justin, In: *Phase Transitions and Critical Phenomena* **6**, Academic Press, London 1976
228. C. H. Bennett, *J. Comput. Phys.* **22**, 245 (1976)
229. D. Frenkel and B. Smit, *Understanding Molecular Simulation: From Algorithms to Application*, Academic Press, New York (1996)
230. G.M. Torrie and J.P. Valleau, *J. Comput. Phys.* **23**, 187 (1977)
231. A.M. Ferrenberg and R.H. Swendsen, *Phys. Rev. Lett.* **61**, 2635 (1988)
232. A.M. Ferrenberg in *Computer Simulation Studies in Condensed Matter Physics III*, eds. D. P. Landau, K. K. Mon and H.-B. Schüttler (Springer-Verlag, Heidelberg)(1991)
233. A.M. Ferrenberg and D. P. Landau, *Phys. Rev. B* **44**, 5081 (1991)
234. Ferrenberg, A. M. and Swendsen, R. H. (1989), *Phys. Rev. Lett.* **63**, 1195.
235. B. Berg, and T. Neuhaus, *Phys. Lett. B* **267**, 249 (1991)
236. B. Berg, and T. Neuhaus, *Phys. Rev. Lett.* **69**, 9 (1992)
237. B. A. Berg, *J. Stat. Phys.* **82**, 343 (1996)
238. B. A. Berg, U .Hansmann, and T. Neuhaus, *Phys. Rev. B* **47**, 497 (1993)



## 2. Index



**Index**

