

Biophysics
A Computational Approach
Concepts, Models, Methods and Algorithms
Lectures 1 and 2: Methods

Dieter W. Heermann

Heidelberg University

October 18, 2016

1. Introduction

- General Remarks
- Force Fields

2. Molecular Dynamics

- Basic Algorithm
- Boundary Conditions
- Force Calculation
- Verlet Algorithm
- Example
- Constant Temperature Molecular Dynamics
- Constant Pressure Molecular Dynamics

■ Event-Driven Molecular Dynamics

3. Langevin Dynamics

4. Monte Carlo Method

- Random Numbers
- Accept/Reject Method
- Gibbs-Sampler
- Markov Chain Monte Carlo
- Metropolis-Hastings Monte Carlo
- Error Analysis
- Hybrid (Hamiltonian) Monte Carlo
- Rejection-Free Monte Carlo

5. Exercises

6. Bibliography

7. Index

In these set of lectures you will find much more material than can be realistically taught in a course. Hence, each lecture will focus on particular aspects. The additional material may serve to further the understanding. We will cover a lot of ground starting off with basic computational methods. Clearly we will not be able to cover them all. Rather we focus on basic methods. To start off here is a very condensed account of the history of the development of computational methods that we will encounter:

1953 Monte Carlo method applied to hard spheres [1]

1956 Molecular dynamics of hard spheres [2]

1964 Molecular dynamics of liquid argon [3]

1971 Molecular dynamics of liquid water [4]

1976 Simulation of protein dynamics [5]

1977 Non-Boltzmann sampling [6]

1992 Mult-Canonical Monte Carlo [7]

1999 Generalized and extended ensemble methods [8]

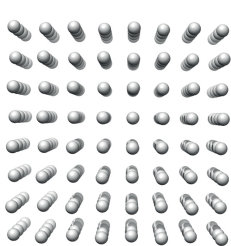
As always, some assumptions need to be made. Here it is assumed that there is

- a basic knowledge of statistical physics
- and some programming experience

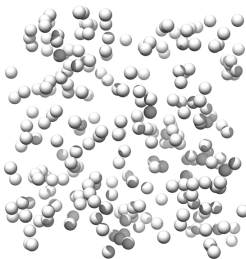
to understand the methods and algorithms.

General literature for the course can for example be found in [9–12].

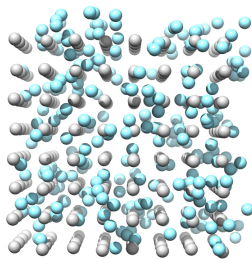
Shown below are 256 Lennard-Jones particles using the parameters for argon at the density of 0.636 and reference temperature $T = 2.53$.



crystal structure



liquid structure



Overlay of the liquid and
crystal state

Note that the radius with which the particles are depicted does not reflect the size of the particle. The image below is produced using the radius where the Lennard-Jones potential passes through zero.

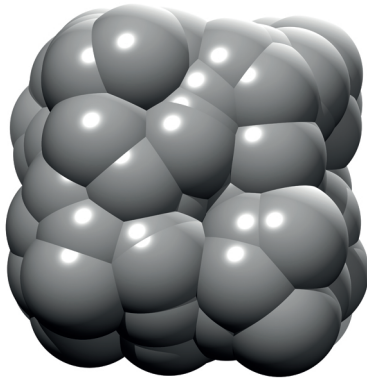


Figure: Example 1: Lennard-Jones liquid where is particles is depicted with the a radius where the Lennard-Jones potential goes to zero.

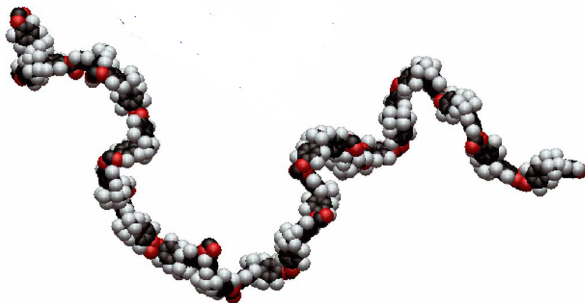


Figure: Example 2: Linear polymer

A molecular model requires typically

- definition of the degrees of freedom
- force fields
- boundary conditions

and a method to generate configuration or conformations as in the above two examples.

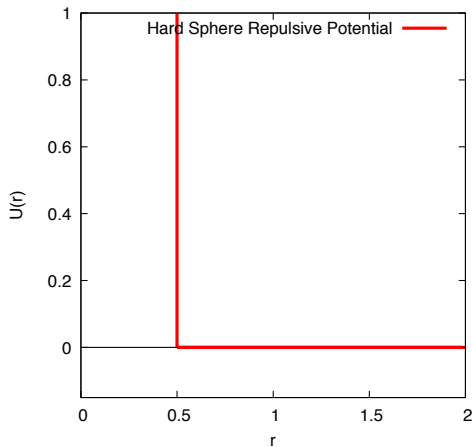
A central part of molecular dynamics, Monte Carlo, etc. is the computation of the forces that are exerted on a particle or unit. One way of categorizing these is to partition them in pair and many-body interaction.

- Bonded: These describes the covalent bonding between two atoms as: bond stretching and bending. Beyond the pair-potential in this class we also find the torsional potential.
- Non-Bonded: Here we find van-der Waals, electrostatic, etc.

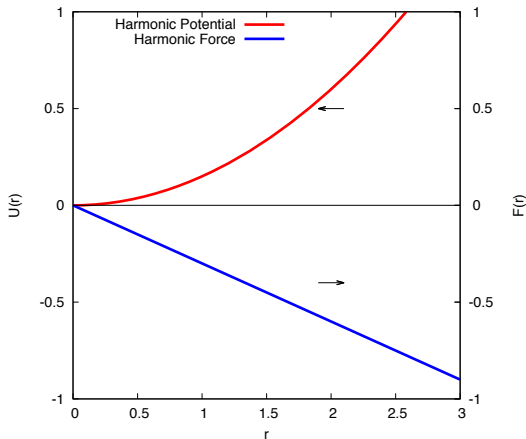
Here we are concerned with very basic potentials. More potentials will be discussed in the more dedicated lectures and sections. For now we consider:

- Excluded volume: Hard Sphere
- Harmonic
- Lennard-Jones
- Electrostatic long range $1/r$

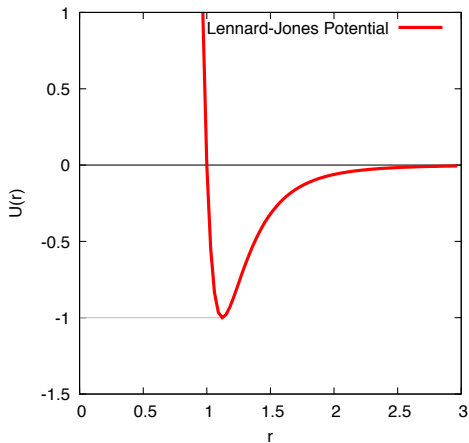
$$U_{\text{HC}}(r) = \begin{cases} \infty & r \leq \sigma \\ 0 & r > \sigma \end{cases} \quad (1)$$



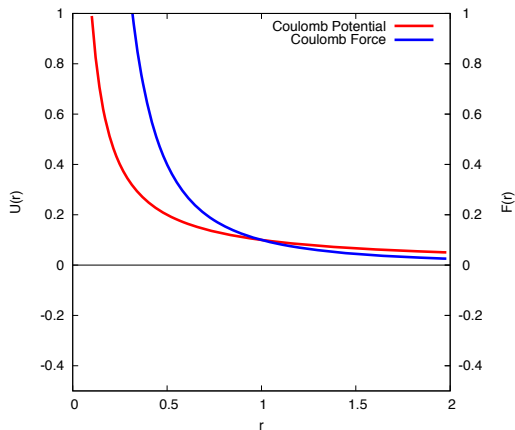
$$U_H(r) = \frac{1}{2}kr^2 \quad r \in [0, \infty] \quad (2)$$



$$U_{\text{LJ}}(r) = 4\epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right) \quad r \in [0, \infty] \quad (3)$$



$$U_C(r) = \frac{q_1 q_2}{4\pi\epsilon_0} \frac{1}{r} \quad r \in [0, \infty] \quad (4)$$



Empirical potentials provide a way to specify for example the interactions in proteins. Specifically for proteins, the base for deriving potentials are protein databases. On the one hand these provide indeed through NMR or other experimental methods data on interatomic relations. Many structures are deposited daily. On the other hand, only those that have more or less fixed structure enter the data base. Proteins like the intrinsically disordered proteins (IDP's) do not, thus biasing the results. Hence implicit is the assumption that the set we are looking at represents the full spectrum of energies.

We will see below, that most of the time pair potentials are derived. This certainly is an assumption that may not apply. From solid state theory we know that certain crystal structures can not be obtained from pair potentials and require many body potentials.

Assume two atoms or particles A and B separated by a distance r . Then the probability for finding them at distance r is given by

$$P_{AB}(r) = \frac{1}{Z} e^{-\beta U_{AB}(r)} . \quad (5)$$

Given a set of structures in a database like for example the protein database, a histogram of the distances that one finds in the database between A and B can be used to get an estimate of P_{AB} . Taking the logarithm of Eq 5 we obtain

$$\ln P_{AB} = -U_{AB}(r) - \ln Z . \quad (6)$$

Since the constant $\ln Z$ is irrelevant we find the specific interaction potential. One can normalize this with the probability P_a for any pair

$$U_{AB}(r) = -kT \ln P_{AB} / \ln P_a(r) . \quad (7)$$

Clearly we need to ask, whether in the calculation of U_{AB} we include every occurrence of pairs AB or whether specific relations along the backbone of the protein need to be considered.

Also, we need to discretize the distances to ensure a sufficient statistics.

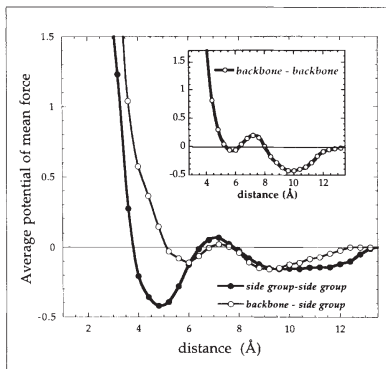


Figure: Empirical potential taken from *Structure-derived potentials and protein simulations* by Jernigan and Bahar [13]

Example

The H2AX protein (shown in Figure 4, Histone H2AX - P27661 (H2AX_MOUSE)) consists of 143 residues. It is a phosphorylated variant of histone H2A and associated with DNA damage. In this example the residues are mapped to a lattice model. To nevertheless capture specificity of each residue a residue-residue interaction and excluded volume constraints are implemented. Each residue interacts with the neighboring residues within a range of interaction using a generalized Lennard-Jones potential

$$U(r) = |\epsilon_{ij}| \left(\frac{\sigma}{r_{ij}} \right)^{12} + \epsilon_{ij} \left(\frac{\sigma}{r_{ij}} \right)^6 \quad \text{for } r_{ij} \leq r_c \quad (8)$$

where the ϵ_{ij} 's are parameterizations for Eq. 7 using X-ray crystallographic data [14] on a large number of protein structures from the protein data bank.

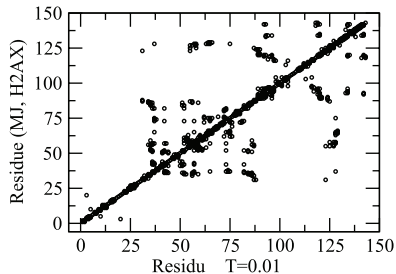
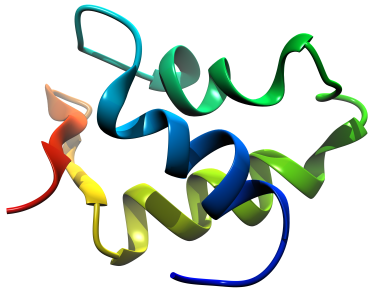


Figure: The left panel shows the ribbon images of H2AX. The right panel shows the contact probability between the residues at a specific temperature. Right image taken from Fritsche et. al. [15]

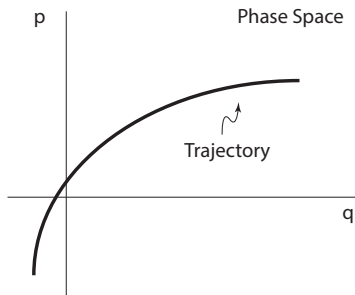
- Double sum
- We will later discuss ways to speed up the calculation
- Example: Lennard-Jones (see Eq. 3)

```
1  /**
   * Force calculation
   */
3  void forceCalculation(float x[], float y[], float z[],
5                        float fx[],float fy[],float fz[],
6                        float rcoffs,float *potential,
7                        float *virial, float side) {
   int i,j;
   float xi,yi,zi;
   float rd, rd2, rd3, rd4, rd6, rd7;
11  float r148;
   float kx,ky,kz;
```

```

1  for (i = 0; i < 1; i++) {
2      for (j = i+1; j < 3; j++) {
3          xi = x[i] - x[j];
4          yi = y[i] - y[j];
5          zi = z[i] - z[j];
6          minimumImage(&xi,&yi,&zi,side);
7          rd = xi*xi + yi*yi + zi*zi;
8          if (rd < rcoffs) {
9              rd2 = rd * rd;
10             rd3 = rd * rd2;
11             rd4 = rd2 * rd2;
12             rd6 = rd2 * rd4;
13             rd7 = rd3 * rd4;
14             *potential += ((1.0 / rd6) - (1.0 / rd3));
15             r148 = (1.0 / rd7) - (1.0 / rd4) * (float).5;
16             *virial -= rd * r148;
17
18             kx      = x[i] * r148;
19             fx[i]   += kx;
20             fx[j]   -= kx;
21
22             ky      = y[i] * r148;
23             fy[i]   += ky;
24             fy[j]   -= ky;
25
26             kz      = z[i] * r148;
27             fz[i]   += kz;
28             fz[j]   -= kz;
29         }
30     }
31 }
32 }
```

The starting point for the Molecular Dynamics (MD) simulation [2, 3, 16–20] is thus a well-defined force field. Using Hamiltonian, Lagrangian or Newton's equations of motion these are approximated by suitable schemes Ψ such that they can be solved numerically.



Let us be more general for the moment and use generalized coordinates p and q with the Hamiltonian instead of the usual coordinates

$$H(q, p) = \frac{1}{2} p^T M^{-1} p + V(q) \quad . \quad (9)$$

From this Hamiltonian we get the equations of motion

$$\frac{d}{dt} q = - \frac{\partial}{\partial p} H \quad (10)$$

$$\frac{d}{dt} p = \frac{\partial}{\partial q} H \quad . \quad (11)$$

An important consideration for any numerical integration scheme is that we want to conserve as many quantities during a numerical evaluation as possible that are conserved due to symmetries etc. This leads us to the concept of symplectic methods. Symplectic methods preserve certain abstract invariants of Hamiltonian systems [21–23] and are stable for linear systems for sufficiently small values of the step size.

We denote the trajectory that we want to generate by Γ

$$\Gamma(t) = \begin{pmatrix} q \\ p \end{pmatrix} . \quad (12)$$

Then this trajectory obeys the equation of motion

$$\frac{d}{dt}\Gamma = J\nabla H(\Gamma) \quad (13)$$

$$J = \begin{pmatrix} 0 & I \\ -I & 0 \end{pmatrix} , \quad (14)$$

where I denotes the unit matrix.

We compute an observable A along the trajectories and hence average along the states we find along the path

$$\langle A \rangle = \frac{1}{n_{\text{obs}}} \sum_{\nu=1}^{n_{\text{obs}}} A(\Gamma_{\nu}(t)) . \quad (15)$$

Here n_{obs} is the number observations we took, i.e., how many iterations we took in the numerical integration of the equations of motion.

Let $\rho_0(\Gamma)$ denote the probability density at time $t = 0$: $\Gamma(0) = \rho_0(\Gamma)$ and let $\rho(\Gamma, t)$ denote the probability density for $\Gamma(t)$. Then we have the Liouville theorem for the trajectories.

$$\rho(\Gamma, 0) = \rho_0 \quad (16)$$

$$\rho_t + \nabla \cdot (\rho J \nabla H) = 0 \quad (17)$$

This states that the flow in phase space is that of an incompressible fluid.

If $\rho_t = 0$, then

$$\nabla H \cdot J \cdot \nabla \rho = 0 \quad (18)$$

and with this

$$\rho(\Gamma) = \frac{e^{-H(\Gamma)/k_B T}}{\int e^{-H(\Gamma)/k_B T} d\Gamma} \quad (19)$$

Now let Ψ be a numerical integrator, i.e. a discretization of the equations of motion

$$\Gamma^{n+1} = \Psi(\Gamma^n) \quad , \quad (20)$$

then the phase space volume needs to be conserved

$$\det \partial_{\Gamma} \Psi(\Gamma) = 1 \quad . \quad (21)$$

The integrator is symplectic, if

$$(\partial_{\Gamma} \Psi(\Gamma))^T J (\partial_{\Gamma} \Psi(\Gamma)) = J \quad (22)$$

i.e. phase space volume and the energy is conserved.

The most straightforward discretization of the equations of motion that involve differentials comes 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 r in a Taylor series

$$r(t + \Delta t) = r(t) + \sum_{i=1}^{n-1} \frac{\Delta^i t}{i!} r^{(i)}(t) + R_n \quad (23)$$

where R_n gives the error involved in the approximation.

Using the forward $t + \Delta t$ and the backward difference $t - \Delta t$

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{F(t)}{2m}\Delta^2 t + \frac{d^3 r}{dt^3} \frac{\Delta^3 t}{3!} + R_4 \quad (24)$$

$$r(t - \Delta t) = r(t) - v(t)\Delta t + \frac{F(t)}{2m}\Delta^2 t - \frac{d^3 r}{dt^3} \frac{\Delta^3 t}{3!} + R_4 \quad (25)$$

$$(26)$$

If we add the two equations, we obtain

$$r(t + \Delta t) = 2r(t) - r(t - \Delta t) + \frac{F(t)}{2m}\Delta^2 t + R_4^* \quad (27)$$

If we subtract the two equations we obtain

$$r(t + \Delta t) + r(t - \Delta t) = 2v(t)\Delta t + R_3^* \quad (28)$$

and hence an estimator for the velocity

$$v(t) = \frac{r(t + \Delta t) + r(t - \Delta t)}{2\Delta t} + R_2^* \quad (29)$$

The estimator for the position and the velocity together comprise what is known as the **Verlet algorithm** [20]. The Verlet algorithm is a second-order method that is indeed symplectic.

So, much hinges on the simulation step-size, since this determines the time-scales, that we can cover. As we have seen above the choice of step size is dominated by stability demands and not by accuracy demands.

Algorithm 1 Molecular Dynamics Algorithm

```
1:  $n = 0$ 
2: Specify positions  $r_i^{-1}$  and  $r_i^0$ 
3: Set  $\Delta t$ 
4: while  $n \neq \text{maxSteps}$  do
5:   Compute the forces at time step  $n$ :  $F_i^n$ 
6:   Compute the positions at time step  $n + 1$   $r_i^{n+1}$ 
7:   Compute the velocities at time step  $n$ :  $v_i^n$ 
8:    $n = n + 1$ 
9: end while
```

Note that in this formulation of the molecular dynamics algorithm we need two starting positions, rather than position and velocity!

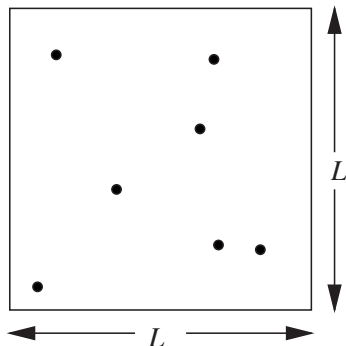
The computational volume is given by

$$\Omega = (a_1, b_1) \times \dots \times (a_d, b_d) \subset \mathbb{R}^d \quad (30)$$

where here for simplicity we assume

$$b_i - a_i = L, \quad a_i, b_i \in \mathbb{R} \quad (31)$$

Of course, more complicated situations may be considered.

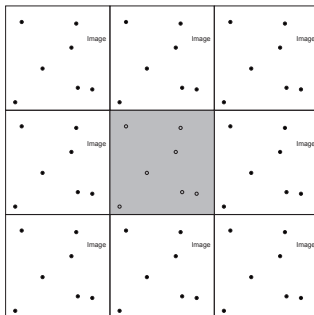


- macroscopic properties
- avoidance of boundaries such as walls
- linear momentum is conserved under periodic boundary conditions
- problem [24]

$$F : \overline{\Omega} \rightarrow \mathbb{R}$$

$$\forall r = (x_1, \dots, x_d) \in \Omega, i = 1, \dots, d :$$

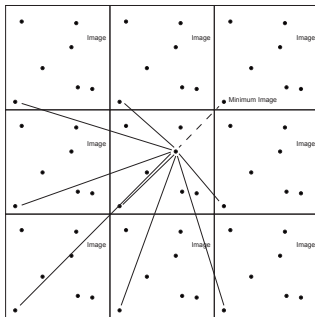
$$F(x_1, \dots, x_{i-1}, a_i, x_{i+1}, \dots, x_d) = F(x_1, \dots, x_{i-1}, b_i, x_{i+1}, \dots, x_d) \quad (32)$$



```
/**
2  * Periodic boundary conditions
3  */
4  void applyPeriodicBoundary(float x[], float y[], float z[],float side) {
5      for (int n = 0; n < N; n++) {
6          if ( x[n] < 0 )      x[n] += side;
7          if ( x[n] > side ) x[n] -= side;
8          if ( y[n] < 0 )      y[n] += side;
9          if ( y[n] > side ) y[n] -= side;
10         if ( z[n] < 0 )      z[n] += side;
11         if ( z[n] > side ) z[n] -= side;
12     }
13 }
```

Algorithm 2 Minimum Image Criterion (here for $d = 1$)

```
1:  $dx = x_j - x_i$   
2: if  $dx > L * 0.5$  then  
3:    $dx = dx - L$   
4: end if  
5: if  $dx \leq -L * 0.5$  then  
6:    $dx = dx + L$   
7: end if
```



```
1  /**
2   * Minimum image convention
3   */
4
5  void minimumImage(float* xi, float* yi, float* zi, float side){
6      float sideh;
7      sideh = side * 0.5;
8
9      if (*xi < -sideh) { *xi += side;}
10     if (*xi > sideh) { *xi -= side;}
11     if (*yi < -sideh) { *yi += side;}
12     if (*yi > sideh) { *yi -= side;}
13     if (*zi < -sideh) { *zi += side;}
14     if (*zi > sideh) { *zi -= side;}
15 }
```

- Verlet table

The Verlet algorithm can be reformulated in such a way as to give a numerically more stable method [25, 26]. Define

$$z_i^n = \frac{r_i^{n+1} - r_i^n}{h} \quad (33)$$

The equations

$$\begin{aligned} r_i^n &= r_i^{n-1} + h z_i^{n-1} \\ z_i^n &= z_i^{n-1} + h F_i^n / m \end{aligned} \quad (34)$$

are called the summed form. A further reformulation yields the velocity form of the Verlet algorithm.

Algorithm 3 Molecular Dynamics Algorithm: NVE Velocity Form

```
1:  $n = 1$ 
2: Set  $\Delta t$ 
3: Specify the initial positions  $r_i^0, r_i^1$ 
4: Specify the initial velocities  $v_i^0, v_i^1$ 
5: Compute the forces  $F_i^1$ 
6: while  $n \neq \text{maxSteps}$  do
7:    $r_i^{n+1} = 2r_i^n - r_i^{n-1} + F_i^n h^2 / m$ 
8:   Compute  $F_i^{n+1}$ 
9:    $v_i^{n+1} = v_i^n + h(F_i^{n+1} + F_i^n) / m$ 
10:   $n = n + 1$ 
11: end while
```

Excluded volume interactions are simulated by the WCA (Weeks-Chandler-Andersen) potential [27], which was designed to model excluded volume interactions by a short-range repulsive force. The WCA potential is basically a truncated and shifted Lennard-Jones potential with the following functional form,

$$U_{\text{WCA}}(r) = \begin{cases} 4\epsilon \left(\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 + c_{\text{shift}} \right) & r < r_{\text{cut}} \\ 0 & r \geq r_{\text{cut}} \end{cases} \quad (35)$$

Here $r_{\text{cut}} = 2^{1/6}$ and $c_{\text{shift}} = \frac{1}{4}$ are chosen such that the minimum of the potential is $U_{\text{WCA}}(r_{\text{min}}) = 0$, the attractive part of the Lennard-Jones interaction being cut off. The WCA potential has two parameters ϵ and σ . σ defines the radius of the monomers' hard core. ϵ controls the energy penalty of another monomer penetrating this hard core.

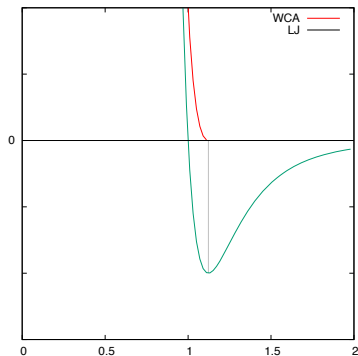


Figure: Weeks-Chandler-Andersen potential as well as the Lennard-Jones potential

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) [28–33]. It can be shown [34] 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 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 [34]

$$\Lambda = \frac{1}{2} \sum_i m v_i^2 = \text{const} \quad (36)$$

(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) [35]

$$\frac{1}{2} \sum_i m v_i^2 = \alpha t \quad (37)$$

$$\beta = \left[(3N - 4) k_B T_{\text{ref}} / \sum_i m v_i^2 \right]^{1/2} \quad (38)$$

so that after the scaling step we have

$$\frac{1}{2} \sum_i m v_i^2 = \frac{1}{2} (3N - 4) k_B T_{\text{ref}} \quad . \quad (39)$$

Algorithm 4 Molecular Dynamics Algorithm: NVT

```
1:  $n = 1$ 
2: Set  $\Delta t$ 
3: Specify the initial positions  $r_i^1$ 
4: Specify the initial velocities  $v_i^1$ 
5: Compute the forces  $F_i^1$ 
6: while  $n \neq \text{maxSteps}$  do
7:    $r_i^{n+1} = r_i^n + h v_i^n + h^2 F_i^n / 2m$ 
8:    $v_i^{n+1} = v_i^n + h(F_i^{n+1} + F_i^n) / 2m$ 
9:   Compute  $\sum_i (v_i^{n+1})^2$  and the scaling factor  $\beta$ .
10:  Scale all velocities  $v_i^{n+1} \rightarrow v_i^{n+1} \beta$ 
11:   $n = n + 1$ 
12: end while
```

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. For a constant particle number N and constant pressure P , the enthalpy H is conserved

$$H = E + PV \quad (40)$$

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 [36–38]. The Lagrangian now looks like

$$\mathcal{L}(\mathbf{r}, \dot{\mathbf{r}}, V, \dot{V}) = \frac{1}{2} \sum_i m \dot{\mathbf{r}}_i^2 - U(\mathbf{r}) + \frac{1}{2} M \dot{V}^2 + P_E V \quad (41)$$

Of course, the variables \mathbf{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 \mathbf{r}_i of the particles by the scaled coordinates ρ_i i.e.,

$$\rho_i = \frac{r_i}{V^{1/3}} = \frac{r_i}{L} \quad (42)$$

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 \mathbf{r}_i over the

fluctuating volume V become integrals of ρ_i over the unit cube. Having written down (42) 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 (42) couples the dynamical variables \mathbf{r} to the volume. Taking the first time derivative we obtain

$$\dot{r}_i = L\dot{\rho}_i + \rho\dot{L} \quad (43)$$

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

$$\frac{p_i}{m} = L\dot{\rho}_i \quad (44)$$

as the momentum conjugate to ρ_i and the Lagrangian becomes

$$\mathcal{L}(\rho, \dot{\rho}, V, \dot{V}) = \frac{1}{2}L^2 \sum_i m\dot{\rho}_i^2 - U(L\rho) + \frac{1}{2}M\dot{V}^2 - P_E V \quad (45)$$

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 ρ but the true r . In a somewhat loose way the Hamiltonian of the system is formulated as [38, 39]

$$\mathcal{H} = \frac{1}{2}L^2 \sum_i m \dot{\rho}_i^2 + U(L\rho) + \frac{1}{2}M\dot{V}^2 + P_E V \quad (46)$$

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

$$\begin{aligned} \frac{d^2 \dot{\rho}_i}{dt^2} &= \frac{F_i}{mL} - \frac{2}{3} \dot{\rho}_i \left(\frac{\dot{V}}{V} \right), \\ \frac{d^2 V}{dt^2} &= \frac{P - P_E}{M} \end{aligned} \quad (47)$$

with the pressure P computed from the virial

$$P = \frac{1}{3L} \left[\sum_i m \dot{\rho}_i^2 + \sum_{i < j} r_{ij} F_{ij} \right] \quad (48)$$

Algorithm 5 Molecular Dynamics Algorithm: NVT

```
1:  $n = 1$ 
2: Set  $\Delta t$ 
3: Specify the initial positions  $r_i^1$ 
4: Specify the initial velocities  $v_i^1$ 
5: Specify an initial volume  $V^0$  consistent with the required density
6: Specify an initial velocity for the volume
7: Compute the forces  $F_i^1$ 
8: while  $n \neq \text{maxSteps}$  do
9:   Compute  $\rho^{n+1}$  and  $V^{n+1}$ 
10:   $r_i^{n+1} = r_i^n + h v_i^n + h^2 F_i^n / 2m$ 
11:   $v_i^{n+1} = v_i^n + h(F_i^{n+1} + F_i^n) / 2m$ 
12:  Compute the pressure  $P^{n+1}$ 
13:   $n = n + 1$ 
14: end while
```

- Robustness
- error (propagation)
- implementation algorithms
- force calculation
- parallelization
- conservation of quantities (linear and angular momentum, energy,...)
- ensemble
- Alder Wainright Event-driven vs time-driven

- Event

Another approach to understand how structure arises is to model part of the system chemically as realistic as possible and take the interaction of atoms with the environment into account only through their stochastic influence. For example, the water molecules and all other possible solvent molecules are not explicitly taken into account.

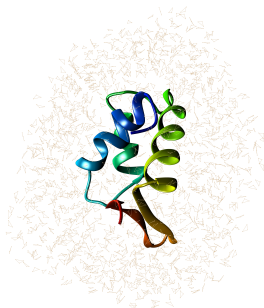


Figure: H2AX protein with solvents

Thus we start off with a Langevin equation [40]

$$M\ddot{r} + \eta\dot{r} + \nabla V = \xi(t) \quad , \quad (49)$$

where M is the mass matrix, η the damping matrix and ξ a normalized white noise resulting from a Wiener process.

Because of the independence of the coordinates, the Langevin equation (49) only depends on the covariance matrix of the noise $\xi(t) = DW(t)$ (here D is the diffusion matrix and W a Wiener process)

$$DD^T = \langle \xi \xi^T dt \rangle \quad . \quad (50)$$

The fluctuation-dissipation theorem relates the diffusion matrix to the damping matrix and the temperature

$$DD^T = 2k_B T \eta \quad . \quad (51)$$

Because of the lack of knowledge of the detailed damping a further reduction in complexity is usual taken by setting the damping matrix proportional to the mass matrix

$$\eta = \gamma M \quad (52)$$

with a damping constant γ . We then get

$$D = (2k_B T \gamma M)^{1/2} . \quad (53)$$

Let ΔE be the energy barrier which the molecule has to take to get to a new state. Then the activation energy can be related to the mean frequency of transition f by an **Arrhenius law**, i.e., the rate increases exponentially with the absolute temperature

$$f = \frac{k_B T}{h} \exp \left(-\frac{\Delta E}{k_B T} \right) \quad (54)$$

where h is the Plank constant.

We shall now assume that we are at low temperatures, where the motions of the involved atoms are small. The protein is further assumed in a local minimum x_{min} and we are interested in the high frequency modes. The highest frequencies are of the order of 10^{14} sec (roughly the C-H bond vibrations).

For simplicity we shall also assume that the eigenmodes are non-degenerate. Then we can expand the potential up to second order

$$V(r) \approx V(r_{min}) + \frac{1}{2}(r - r_{min})^T \bar{V}(r - r_{min}) \quad (55)$$

where $\bar{V} = \nabla^2 V(r_{min})$. Ignoring in the limit of low temperature and high frequency the damping and the random force term we obtain

$$M\ddot{r} + \bar{V}(r - r_{min}) = 0 \quad (56)$$

with the general solution

$$r = r_{min} + \sum_I u_I \exp(i\omega_I t) \quad (57)$$

Here ω_I are the frequencies and u_I are the normal modes.

What are Monte Carlo Methods?

- In the widest sense of the term, Monte Carlo (MC) simulations mean any simulation (not even necessarily a computer simulation) that utilizes random numbers in the simulation algorithm.
- Monte Carlo simulations are statistical and non-deterministic. Hence each simulation will give a different result, but the results will be related via some statistical error.
- The Monte Carlo algorithm was named the top algorithm of the 20th century by mathematicians and physicists.

Why do we need this?

- Multidimensional integrals
- Systems with a large number of degrees of freedom
 - Many atoms in a gas, liquid, solid
 - Many electrons in an atom
 - Gene expression
 - Networks
 - ...

Typical Parts of a Monte Carlo Method

- Probability Distribution Functions
- Random Number Generator
- Sampling Rule
- Evaluating
- Error Estimation

Bayes Theorem

- Let A and B be events

$$P(A|B) = \frac{P(B|A)P(A)}{P(B|A)P(A) + P(B|\bar{A})P(\bar{A})} \quad (58)$$

$$= \frac{P(B|A)P(A)}{P(B)} \quad (59)$$

Anyone who considers arithmetical methods of producing random digits is, of course, in a state of sin.

John Von Neumann, 1951

Intuitively, we can list a number of criteria that a sequence of numbers must fulfill to pass as a random number sequence:

- unpredictability,
- independence,
- without pattern.

These criteria appear to be the minimum request for an algorithm to produce random numbers. More precisely we can formulate:

- uniform distribution,
- uncorrelated,
- passes every test of randomness,
- large period before the sequence repeats (see later),
- sequence repeatable and possibility to vary starting values,
- fast algorithm.

The most common generators use very basic operations and apply them repeatedly on the numbers generated in previous steps. We formulate this as a recursion relation

$$x_{i+1} = G(x_i), \quad x_0 = \text{initial value} \quad , \quad (60)$$

where we have made explicit only the dependence on the immediate predecessor. The most important representatives of this class of generators are the

- linear congruential,
- lagged Fibonacci,
- shift-register or a
- combination of linear congruential.

Linear Congruential Generators

A very simple generator is constructed using the modulo function.

$$G(x) = (xa + b) \bmod M \quad (61)$$

This function produces a dilatation, translation and a folding back into the interval. Random number generators based on this function are called **linear congruential generators** or LCG(a,b,M) for short. If we assume integers as the set on which the modulo function is defined, then for example, the range of integer numbers for a 32-bit architecture is at most $M = 2^{31} - 1$. Here we assume that one bit is taken for the sign of the number. Then the numbers range at most from 0 to $M - 1$. Of course, we can map these onto the real interval between 0 and 1, recognizing that this is an approximation to the real-valued random numbers.

Inverse Congruential Generators

A very simple generator is constructed using the modulo function.

$$x_{n+1} = (x_n^{-1}a + b) \bmod M \quad , \quad (62)$$

where x_n^{-1} is the multiplicative inverse of x_n in the integers mod m with 0^{-1} defined as 0.

- The choice of the parameters a , b and M determine the statistical properties and how many different numbers we can expect before the sequence repeats itself.

- The period can be shown to be maximal, if M is chosen to be a prime number. Then the whole range of numbers occurs.
- Here we only consider modulo generators with $b = 0$.
- Such generators are called **multiplicative** and the short form MLCG(a, M) is used for such generators.
- These are the most commonly used, since one can show that **additive** generators, i.e. generators with b in general non zero have undesirable statistical properties.
- The choices for the parameter a are manifold. For example $a = 16807$, 630360016 or 397204094 are possible choices with $M = 2^{31} - 1$.

```
/*----- */
/*          Modulo Generator          */
/*----- */
int ModGenerator(modul, multi, inc, seed, max_sweeps, x)
{
    int      modul;
    int      multi;
    int      inc;
    int      seed;
    int      max_sweeps;
    float     *x;

    /*----- */
    /*          Declarations          */
    /*----- */
    int i;
    double r;
    double factor, increment, modulus;
    /*----- */
    /*          End of declares      */
    /*----- */

    r      = (double) seed;
    factor  = (double) multi;
    increment = (double) inc;
    modulus = (double) modul;

    for(i=0; i< max_sweeps; i++) {
        r=fmod(r*factor + increment, modulus);
        x[i] = (float) r / modulus;
    }
    return 0;
}
```

Figure: C source for a modulo random number generator

```
#include <stdlib.h>
int iseed, randInt;
float randFloat;

srand(iseed);
randInt = rand();
randFloat = (float) randInt / (float) RAND_MAX;
```

Figure: C source for the implicit modulo random number generator

Add-with-Carry/Subtract-with-Carry Generators

- Add-with-carry and subtract-with-carry generators rely on two numbers, the carry c and the modular base M .
- Add-with-carry generator;

$$x_{n+1} = (x_{n-s} + x_{n-r} + c) \bmod M \quad (63)$$

- Subtract-with-carry

$$x_{n+1} = (x_{n-s} - x_{n-r} - c) \bmod M \quad (64)$$

- Problems:
 - Require an initial seed of a sufficiently long sequence.
 - Pairs (or triplets) of terms fall on planes (see modulo generator).

Fibonacci Generators

The **lagged Fibonacci generator**, symbolically denoted by $LF(p,q,\otimes)$ with $p > q$, is based on a Fibonacci sequence of numbers with respect to an operation which we have given the generic symbol \otimes .

Let \mathcal{S} be the model set for the operation \otimes , for example the positive real numbers, the positive integers, or the set $\mathcal{S} = \{0, 1\}$. The binary operation \otimes computes a new number from previously generated numbers with a lag p

$$x_n = x_{n-p} \otimes x_{n-q} \quad , \quad p > q \quad . \quad (65)$$

To start the generator we need p numbers. These can be generated using for example a modulo generator. The advantage of the lagged Fibonacci generator, apart from removing some of the deficiencies that are build into the modulo type generators, is that one can operate on the level of numbers or on the level of bits.

```
for(i=0; i< max_sweeps; i++) {
    mf[p] = mf[p] + mf[q];
    if ( mf[p] > 1 ) mf[p] -= 1;
    x[i] = mf[p];
    if ( ++p == lagP-1 ) p = 0;
    if ( ++q == lagP-1 ) q = 0;
}
```

Figure: Fibonacci

In the following I have listed some lagged Fibonacci generators:

Recursion Relation	Period
$x_i = x_{i-17} - x_{i-5} \bmod (2^n)$	$(2^{17} - 1)2^{n-1}$
$x_i = x_{i-17} + x_{i-5} \bmod 2^n$	$(2^{17} - 1)2^{n-1}$
$x_i = x_{i-31} - x_{i-13} \bmod 2^n$	$(2^{31} - 1)2^{n-1}$
$x_i = x_{i-55} - x_{i-24}$	$2^{24}(2^{97} - 1)$ with 24 Bit Mantissa

For example, we can construct a **generalized shift-register generator** GFSR(p,q,⊗), where the operation is interpreted as the **exclusive or**, which acts on every of the 32 bits in a computer word. This generator is also known under the name of *R250*. (Follow this link to access the code for the R250.c.)

Figure: The shift bit register generator R250

- Let us turn now to the generation of non-uniform distributions. First we look at the normal or Gaussian distribution.

- Typically algorithms generating non-uniform variates do so by converting uniform variates.
- In its most straightforward form a normal deviate x with mean $\langle x \rangle$ and standard deviation σ is produced as follows:
- Let n be an integer, determined by the needed accuracy. Then
 - sum n uniform random numbers r_i from the interval $(-1, 1)$:

$$s_n = \sum_{i=1}^n r_i$$

- and let $x = \langle x \rangle + \sigma s_n \sqrt{3.0/n}$.

Let $G(x)$ be a function on the interval $[a, b]$ with $0 < G(x) < 1$ and $f(x)$ the probability distribution $f(x) = a \exp[-G(x)]$, where a is a constant.

Algorithm 6 Algorithm

- 1: Generate r from a uniform distribution on $(0, 1)$
 - 2: Set $x = a + (b - a)r$
 - 3: Calculate $t = G(x)$
 - 4: Generate r_1, r_2, \dots, r_k from a uniform distribution on $(0, 1)$ (k is determined from the condition $t > r_1 > r_2 > \dots > r_{k-1} < r_k$)
 - 5: **if** $t < r_1$ **then**
 - 6: $k = 1$
 - 7: **end if**
 - 8: **if** k is even **then**
 - 9: reject x and go to 1
 - 10: **else**
 - 11: x is a sample
 - 12: **end if**
-

An interesting method for generating normal variates is the **polar method**. It has the advantage that two independent, normally distributed variates are produced with practically no additional cost in computer time.

Algorithm 7 Polar Algorithm

- 1: Generate two independent random variables, U_1, U_2 from the interval $(0, 1)$.
 - 2: Set $V_1 = 2U_1 - 1, V_2 = 2U_2 - 1$
 - 3: Compute $S = V_1^2 + V_2^2$
 - 4: **if** $S \geq 1$ **then**
 - 5: return to step 1
 - 6: **else**
 - 7: $x_1 = V_1 \sqrt{-2 \ln S / S}$
 - 8: $x_2 = V_2 \sqrt{-2 \ln S / S}$
 - 9: **end if**
-

- Another idea of converting one distribution into another is to accept or reject drawn number for an initial distribution such that the accepted numbers have the desired distribution.
- Assume that we are given a uniform random number generator $U \sim (0, 1)$ and $X \sim g$.
- We want to generate $Y \sim f$.
- Assume that there exists a constant c such that $f(x) < cg(x)$ for all x .

Algorithm 8 Accept/Reject Algorithm

```
1: Generate  $X \sim g$ 
2: Generate  $U \sim (0, 1)$ 
3: if  $U \leq f(X)/cg(X)$  then
4:    $Y = X$ 
5: else
6:   Goto 1
7: end if
```

- To prove that this is correct we show that

$$P(X < y | U \leq f(X)/cg(X)) = P(Y \leq y) \quad .$$

Note that

$$\frac{P(X < y | U \leq f(X)/cg(X)) = P(Y \leq y)}{P(U \leq f(X)/cg(X))} = \frac{\int_{-\infty}^y \int_0^{f(x)/cg(x)} g(x) du dx}{\int_{-\infty}^{\infty} \int_0^{f(x)/cg(x)} g(x) du dx}$$

which simplifies to

$$\int_{-\infty}^y f(x) dx \quad .$$

Assume $x = (x^1, x^2)$ with target $\pi(x)$

Gibbs-Sampler Algorithm:

Algorithm 9 Gibbs Sampler Algorithm

```
1: initialize  $x_0 = (x_0^1, x_0^2)$ 
2: while  $i \leq \text{mcsmax}$  do
3:   sample  $x_i^1 \sim \pi(x^1 | x_{i-1}^2)$ 
4:   sample  $x_i^2 \sim \pi(x^2 | x_i^1)$ 
5: end while
```

$\{x^1, x^2\}$ is a Markov chain

Generalization: $x = (x^1, \dots, x^p), p > 2$

Algorithm 10 Generalized Gibbs Sampler Algorithm

```
1: initialize  $x_0 = (x^1, \dots, x^p)$ 
2: while  $i \leq \text{mcsmax}$  do
3:   for  $k = 1 \rightarrow p$  do
4:     sample  $x_i^k \sim \pi(x^k | x_i^{-k})$ 
5:   end for
6: end while
```

where $x_i^{-k} = (x_i^1, \dots, x_i^{k-1}, x_{i-1}^{k+1}, \dots, x_{i-1}^p)$

A **Markov chain** is a sequence of random variables $\{x_n; n \in \mathbb{N}\}$ which satisfies the property

$$P(x_n | x_0, \dots, x_{n-1}) = P(x_n | x_{n-1})$$

Goal: Given a distribution π , construct transition probabilities P such that asymptotically as $n \rightarrow \infty$

$$\frac{1}{n} \sum_{i=1}^n \phi(x_i) \rightarrow \int \phi(x) \pi(x) dx$$

and

$$X_i \sim \pi$$

Examples:

- $\pi(x) = Z^{-1} \exp\{-\beta H(x)\}$
- Autoregression for $|\alpha| < 1$

$$X_n = \alpha X_{n-1} + V_n, V_n \sim N(0, \sigma^2)$$

Here $\pi(x) = \mathcal{N}(x; 0, \frac{\sigma^2}{1-\alpha^2})$

- Random Walk on a circle
 - The walker at time state i is in one of the four points
 - At state $i + 1$ the walker jumps to one of the two neighbours with probability p and stays at the same point with probability $1 - 2p$.
- Let $P(x_{i+1}|x_i) =: P_{x_i, x_{i+1}}$ then

$$P = \begin{pmatrix} 1-2p & p & 0 & p \\ p & 1-2p & p & 0 \\ 0 & p & 1-2p & p \\ p & 0 & p & 1-2p \end{pmatrix}, p \leq 1/2$$

- $P_{x,y} \geq 0, \sum_y P_{xy} = 1$
- Calculate P^n

$$\lim_{n \rightarrow \infty} P^n = \begin{pmatrix} 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \\ 1/4 & 1/4 & 1/4 & 1/4 \end{pmatrix} = \pi$$

a uniform distribution, as expected!

We want:

- The desired distribution π to be a fixed point of the algorithm

$$\int P(x, y) \pi dx = \pi(y)$$

- The successive distributions of the Markov chain converges towards π
- The estimator $\frac{1}{n} \sum_i^n \phi(x_i)$ converges and asymptotically $X_n \sim \pi$
- For every π there exists infinitely many P that have π as there invariant distribution.
- How to choose a good one? *Criterion*: Rate of convergence
- Convergence ensured if the chain is irreducible, aperiodic and every state can be reached.

We require:

- Irreducibility: From any state we can go to any state

$$\forall x, y \in S \quad \exists n \in \mathbb{N}, n > 0 : (P^n)_{xy} > 0$$

- Aperiodicity

then, if P is irreducible and aperiodic we have

- $\lim_{n \rightarrow \infty} P^n = \pi$

- if π_y is not identically zero, then

$$\sum_{y \in S} \pi_y = 1, \quad \sum_{y \in S} \pi_y P_{yx} = \pi_x$$

- π_x is a unique non-negative solution of the above equation and a probability distribution on S .

If we know π_x construct P such that π_x is its equilibrium distribution. For this it is enough to choose P such that

- 1 $\sum_{y \in S} \pi_y P_{yx} = \pi_x$
- 2 P is irreducible and aperiodic

Replace (1) by the **detailed balance condition**

$$\pi_y P_{yx} = \pi_x P_{xy}$$

which implies (1).

We will now construct an algorithm to that realizes a constant temperature ensemble, i.e.

$$\pi(x) = Z^{-1} \exp\{-\beta H(x)\}$$

The state space can for example be configurations of spins (see later Ising model), positions of atoms in a liquid, polymers conformation etc.

- Let $P^0 = \{p_{xy}^{(0)}\}$ be a irreducible transition matrix.
- We will use P^0 to propose transitions from x to y .
- These transitions will then either be accepted with a probability a_{xy} and rejected $1 - a_{xy}$.

- The complete transition matrix P is then constructed as:

$$p_{xy} := p_{xy}^{(0)} a_{xy} \quad \text{if } x \neq y ,$$

$$p_{xx} := p_{xx}^{(0)} + \sum_{x \neq y} p_{xy}^{(0)} (1 - a_{xy}) \quad [\text{zero transition}] ,$$

$$\text{where } \forall x, y \in S : 0 \leq a_{xy} \leq 1 .$$

With this we have

$$\frac{a_{xy}}{a_{yx}} = \frac{\pi_y p_{yx}^{(0)}}{\pi_x p_{xy}^{(0)}} \quad \forall x, y \in S : x \neq y .$$

- If we use

$$a_{xy} := F\left(\frac{\pi_y \rho_{yx}^{(0)}}{\pi_x \rho_{xy}^{(0)}}\right) \quad \forall x, y \in \mathcal{S} : x \neq y \quad (66)$$

then

$$\frac{a_{xy}}{a_{yx}} = \frac{F\left(\frac{\pi_y \rho_{yx}^{(0)}}{\pi_x \rho_{xy}^{(0)}}\right)}{F\left(\frac{\pi_x \rho_{xy}^{(0)}}{\pi_y \rho_{yx}^{(0)}}\right)} = \frac{F(z)}{F(1/z)} \stackrel{!}{=} z \quad (67)$$

with

$$z := \frac{\pi_x \rho_{xy}^{(0)}}{\pi_y \rho_{yx}^{(0)}} \quad (68)$$

- The condition of detailed balance is satisfied if

$$\forall z : \frac{F(z)}{F(1/z)} \stackrel{!}{=} z \quad (69)$$

- Often used choices are

$$F(z) = \min(z, 1) \quad (70)$$

and

$$F(z) = \frac{z}{1+z} \quad (71)$$

- Note that the proposals of states need not to be symmetric. As a further point we note that it was proven[41] that the choice $F(z) = \min(z, 1)$ is optimal in that suitable candidates are rejected least often and hence statistical efficiency is optimized

[1, 42]

Algorithm 11 Metropolis-Hastings Monte Carlo Algorithm

```
1: for  $i=0$ ;  $i < \text{mcsmax}$  do
2:   sample a new state  $x$ ;
3:   set  $y = x_i$ ;
4:   sample a uniform random number  $r$  from  $(0, 1)$ ;
5:   if  $r \leq \min\{\frac{p_{xy}^0 \pi_x}{p_x^0 y \pi_y}, 1\}$  then
6:      $x_{i+1} = x$ ;
7:   else
8:      $x_{i+1} = y$ ;
9:   end if
10: end for
```

- What does this mean if we calculate the time average of an observable A , which by necessity can cover only a finite observation time?
- Let us consider the statistical error for n successive observations $A_i, i = 1, \dots, n$:

$$\langle (\delta A)^2 \rangle = \left\langle \left[n^{-1} \sum_{i=1}^n (A_i - \langle A \rangle)^2 \right] \right\rangle . \quad (72)$$

- In terms of the autocorrelation function for the observable A

$$\phi_A(t) = \frac{\langle A(0)A(t) \rangle - \langle A \rangle^2}{\langle A^2 \rangle - \langle A \rangle^2} \quad (73)$$

We define two characteristic correlation times.

■ Exponential autocorrelation time

- Typically we expect that (asymptotically, for large t) one gets an exponential behavior

$$\Phi_A(t) \propto \exp \left(- \frac{t}{\tau_{A,exp}} \right) \quad (74)$$

- We do expect, though, that the complete expression involves a sum over several such terms; here we consider only the asymptotically most leading term with largest autocorrelation time.

- **Integrated autocorrelation time**

$$\tau_A^{int} = \int_0^\infty \phi_A(t) dt \quad . \quad (75)$$

- We can rewrite the statistical error as

$$\langle (\delta A)^2 \rangle \cong \frac{2\tau_A}{n\delta t} \left[\langle A^2 \rangle - \langle A \rangle^2 \right] \quad , \quad (76)$$

where δt is the time between observations, i.e., $n\delta t$ is the total observation time τ_{obs} .

- We notice that the error does not depend on the spacing between the observations but on the total observation time.
- Also the error is not the one which one would find if all observations were independent.
- The error is enhanced by the characteristic (integral) correlation time between configurations.
- Only an increase in the sample size and/or a reduction in the characteristic correlation time τ_A can reduce the error.

- In conventional Monte-Carlo (MC) calculations of condensed matter systems, such as an N -particle system with a Hamiltonian $\mathcal{H} = \mathcal{U}$, only local moves (displacement of a single particle) are made.
- Updating more than one particle typically results in a prohibitively low average acceptance probability $\langle P_A \rangle$.
- This implies large relaxation times and high autocorrelations especially for macromolecular systems.
- In a Molecular Dynamics (MD) simulation, with $\mathcal{H} = \mathcal{T} + \mathcal{U}$, on the other hand, global moves are made.
- The MD scheme, however, is prone to errors and instabilities due to the finite step size in time.
- In order to introduce temperature in the microcanonical context, isokinetic MD schemes are often used.
- However, they do not yield the canonical probability distribution, unlike Monte-Carlo calculations.

- The Hybrid Monte-Carlo (HMC) method combines the advantages of Molecular Dynamics and Monte-Carlo methods:
 - it allows for global moves (which essentially consist in integrating the system through **phase space**);
 - HMC is an exact method, i.e., the ensemble averages do not depend on the step size chosen;
 - algorithms derived from the method do not suffer from numerical instabilities due to finite step size as MD algorithms do;
 - and temperature is incorporated in the correct statistical mechanical sense.
- In the HMC scheme global moves can be made while keeping the average acceptance probability $\langle P_A \rangle$ high.

- One global move in **configuration** space consists in integrating the system through **phase** space for a fixed time t using some discretization scheme (δt denotes the step size)

$$\begin{aligned} g^{\delta t} : \quad R^{6N} &\longrightarrow R^{6N} \\ (x, p) &\longrightarrow g^{\delta t}(x, p) =: (x', p') \end{aligned}$$

of Hamilton's equations

$$\begin{aligned} \frac{dx}{dt} &= \frac{\partial \mathcal{H}}{\partial p} \\ \frac{dp}{dt} &= -\frac{\partial \mathcal{H}}{\partial x}. \end{aligned} \tag{77}$$

- Since the system is moved deterministically through phase space, the conditional probability of suggesting configuration x' starting at x is given by

$$p_C(x \rightarrow x') dx' = p_C(p) dp. \tag{78}$$

- The initial momenta are drawn from a Gaussian distribution at inverse temperature β :

$$p_C(p) \propto e^{-\beta \sum_{j=1}^N \frac{p_j^2}{2m}}. \quad (79)$$

- Thus

$$P_A((x, p) \rightarrow g^{\delta t}(x, p)) = \min\{1, e^{-\beta \delta \mathcal{H}}\}, \quad (80)$$

where

$$\delta \mathcal{H} = \mathcal{H}(g^{\delta t}(x, p)) - \mathcal{H}(x, p)$$

is the discretization error associated with $g^{\delta t}$. Using the algebraic identity

$$e^{-\mathcal{H}(x, p)} \min\{1, e^{-\delta \mathcal{H}}\} = e^{-\mathcal{H}(g^{\delta t}(x, p))} \min\{e^{\delta \mathcal{H}}, 1\} \quad (81)$$

- it can be shown that for a discretization scheme which is **time-reversible**

$$g^{-\delta t} \circ g^{\delta t} = id \quad (82)$$

- and **area-preserving**

$$\det \frac{\partial g^{\delta t}(x, p)}{\partial (x, p)} = 1, \quad (83)$$

detailed balance is satisfied:

$$\begin{aligned} p(x)p_M(x \rightarrow x')dx dp &= p(x)p_C(p)P_A((x, p) \rightarrow g^{\delta t}(x, p))dx dp \\ &= p(x')p_C(p')P_A(g^{\delta t}(x, p) \rightarrow (x, p))dx dp \\ &= p(x')p_C(p')P_A((x', p') \rightarrow g^{-\delta t}(x', p'))dx dp \\ &= p(x')p_C(p')P_A((x', p') \rightarrow g^{-\delta t}(x', p'))dx' dp' \\ &= p(x')p_M(x' \rightarrow x)dx' dp'. \end{aligned}$$

- Thus, provided the discretization scheme used is **time-reversible** and **area-preserving**, the HMC algorithm generates a Markov chain with the stationary probability distribution $p(x)$.
- The probability distribution is entirely determined by the detailed balance condition.
- Therefore neither $p(x)$ nor any ensemble averages depend on the step size δt chosen.

- However, the average acceptance probability $\langle P_A \rangle$, because of (80), depends on the average discretization error $\langle \delta \mathcal{H} \rangle$ and hence does depend on δt .
- It can be shown that for $(\varrho, T) \neq (\varrho_c, T_c)$

$$\langle P_A \rangle = \operatorname{erfc}\left(\frac{1}{2} \sqrt{\beta \langle \delta \mathcal{H} \rangle}\right)$$

is a good approximation for sufficiently large systems ($N \rightarrow \infty$) and small step sizes ($\delta t \rightarrow 0$).

- From normalization and the area-preserving property one has

$$\langle e^{-\beta \delta \mathcal{H}} \rangle = 1. \tag{84}$$

- Equation (84) can be expanded into cumulants

$$\langle \delta \mathcal{H} \rangle = \frac{\beta}{2} \langle (\delta \mathcal{H} - \langle \delta \mathcal{H} \rangle)^2 \rangle + \dots$$

- In order to obtain a nonzero average acceptance probability $\langle P_A \rangle$ in the limit $N \rightarrow \infty$ one has to let $\delta t \rightarrow 0$, keeping $\langle (\delta \mathcal{H} - \langle \delta \mathcal{H} \rangle)^2 \rangle$ fixed.

- In this limit higher-order cumulants will vanish. The resulting distribution of the discretization error will thus be gaussian with mean and width related through

$$\langle \delta \mathcal{H} \rangle = \frac{\beta}{2} \langle (\delta \mathcal{H} - \langle \delta \mathcal{H} \rangle)^2 \rangle. \quad (85)$$

- From (80) and (85) one has in this case

$$\begin{aligned} \langle P_A \rangle &= \frac{1}{\sqrt{2\pi \langle (\delta \mathcal{H} - \langle \delta \mathcal{H} \rangle)^2 \rangle}} \int_{-\infty}^{\infty} dt \min\{1, e^{-\beta t}\} e^{-\frac{(t - \langle \delta \mathcal{H} \rangle)^2}{2 \langle (\delta \mathcal{H} - \langle \delta \mathcal{H} \rangle)^2 \rangle}} \\ &= \operatorname{erfc}\left(\frac{1}{2} \sqrt{\beta \langle \delta \mathcal{H} \rangle}\right). \end{aligned} \quad (86)$$

- The square root in (86) is always well defined since (84) implies

$$\langle \delta \mathcal{H} \rangle \geq 0.$$

- Equality holds in the limit $\delta t \rightarrow 0$, where energy is conserved exactly and $\langle P_A \rangle = 1$.
- Increasing the step size will result in a lower average acceptance probability $\langle P_A \rangle$. Varying δt , the average acceptance probability $\langle P_A \rangle$ can thus be adjusted to minimize autocorrelations.

- The momenta do not necessarily have to be drawn from the Gaussian distribution.
- A particularly simple and computationally efficient alternative would be a uniform momentum distribution.
- This choice, however, did not prove successful, since a cut-off has to be introduced for computational reasons. This cut-off must be taken into account in P_A , leading to a very low average acceptance probability $\langle P_A \rangle$.
- It is clear that instead of choosing a discretization scheme of Hamilton's equations (77) any time-reversible and area-preserving discrete mapping can be used to propagate the system through phase space.

Exercise 1: **Test** abc

- [1] M. N. Rosenbluth A. H. Teller N. Metropolis, A. W. Rosenbluth and E. Teller. *J. Chem. Phys.*, 21:1087–1091, 1953.
- [2] T. E. Wainwright B. J. Alder. *J. Chem. Phys.*, 27(1208), 1957.
- [3] A. Rahman. *Phys. Rev. A*, 136:405, 1964.
- [4] A. Rahman and F. Stillinger. Molecular dynamics study of liquid water. *J. Chem. Phys.*, 55(5), 1971.
- [5] M. Karplus P.G. Wolynes J.A. McCammon, B.R. Gelin. *Nature*, 262:325–26, 1976.
- [6] J. P. Valleau G. Torrie. *J. Comput. Phys*, 23:187, 1977.
- [7] Bernd A. Berg and Thomas Neuhaus. Multicanonical ensemble: A new approach to simulate first-order phase transitions. *Physical Review Letters*, 68(1):9–12, 01 1992.
- [8] Fugao Wang and D. P. Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical Review Letters*, 86(10):2050–2053, 03 2001.
- [9] Berend Smit and Daan Frenkel. *Understanding Molecular Simulation*. Academic Press, 2001.
- [10] Andrew R. Leach. *Molecular Modelling: Principles and Applications*. Prentice-Hall, 20012.

- [11] K. Binder and D.W. Heermann. *Monte Carlo Simulation in Statistical Physics*. Springer-Verlag (first edition 1988), 2017.
- [12] P. Allen and D.J. Tildesley. *Computer Simulations of Liquids*. Clarendon Press, Oxford, 1987. , ,.
- [13] Robert L Jernigan and Ivet Bahar. Structure-derived potentials and protein simulations. *Current Opinion in Structural Biology*, 6(2):195–209, 1996.
- [14] S. Miyazawa and R.L. Jernigan. *Macromolecules*, 18:534–552, 1985.
- [15] Miriam Fritsche, Ras B. Pandey, Barry L. Farmer, and Dieter W. Heermann. Variation in structure of a protein (h2ax) with knowledge-based interactions. *PLoS ONE*, 8(5):e64507–, 05 2013.
- [16] B. J. Alder T. E. Wainwright. *Nuovo cimento, Suppl. Sec.*, 9:116, 1958.
- [17] T. E. Wainwright B. J. Alder. *J. Chem. Phys.*, 31:456, 1959.
- [18] T. E. Wainwright B. J. Alder. *J. Chem. Phys.*, 33:1439, 1960.
- [19] 1439 B. J. Alder, T. E. Wainwright **33**. *J. Chem. Phys.*, 33:1439, 1960. R. Beeler, Jr.: In *Physics of Many-Particle Systems*, ed. by C. Meeron (Gordon and Breach, New York 1964).
- [20] L. Verlet. *Phys. Rev.*, 159:98, 1967.
- [21] Y. B. Suris. *Comput. Math. Phys.*, 27:149–156, 1987.

- [22] M. Duncan B. Gladman and J. Candy. Symplectic integrators for long-term integrations in celestial mechanics. *Celestial Mech. Dynam. Astronom.*, 52:221–240, 1991.
- [23] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems*. Chapman and Hall, London, 1994.
- [24] Randall B. Shirts, Scott R. Burt, and Aaron M. Johnson. Periodic boundary condition induced breakdown of the equipartition principle and other kinetic effects of finite sample size in classical hard-sphere molecular dynamics simulation. *The Journal of Chemical Physics*, 125(16), 2006.
- [25] M. L. Klein S. Nose. *J. Chem. Phys.*, 78:6928, 1983. Dahlquist, A. Bjorck: Numerical Methods (Prentice Hall, Englewood Cliffs, NJ 1964).
- [26] P. H. Berens K. R. Wilson: W. C. Swope, H. C. Andersen. *J. Chem. Phys.*, 76:637, 1982.
- [27] D. Chandler J. D. Weeks and H. C. Andersen. *J. Chem. Phys.*, 54:5237, 1971.
- [28] R. B. Hickman A. J.C. Ladd W. T. Ashurst B. Moran: Phys. Rev. A 22 1690 W. G. Hoover, D. J. Evans. 1980. 1980.
- [29] I I I W. G. Hoover. Physica A 18. 1983. 1983.
- [30] I I I W. G. Hoover. Physica A 18. W. 1983. G. Hoover. In Nonlinear Fluid Behaviour, ed. by H.J. Hanley (North-Holland, Amsterdam).
- [31] B. Moran: Phys. Rev. Lett. 48 3297 W. G. Hoover, A. J.C. Ladd. 1983. 1983.

- [32] 63 D. J. Evans, G. P. Morriss: Chem. Phys. 77. 1983. 1983.
- [33] 63 D. J. Evans, G. P. Morriss: Chem. Phys. 77. D. 1983. J. Evans: J. Chem. Phys. 78, 3297 (1983).
- [34] 3067 J. M. Haile, S. Gupta J. Chem. Phys. 79. 1983. 1983.
- [35] G. P. Morriss: Daresbury Lab. Information Quarterly for Computer Simulation of Condensed Phases 17 25 D. M. Heyes, D. J. Evans. 1985. 1985.
- [36] 1243 D. Brown, J. H.R. Clarke: Mol. Phys. Sl. 1984. 1984.
- [37] 179 J. R. Ray: Am. J. Phys. 40. 1972. 1972.
- [38] 2384 H. C. Andersen: J. Chem. Phys. 72. 1980. 1980.
- [39] 2412 J. M. Haile, H. W. Graben: J. Chem. Phys. 73. 1980. 1980.
- [40] G. van Kampen. *Stochastic Processes in Physics and Chemistry*. North Holland, Amsterdam, 1981.
- [41] 607-612 P. H. Peskun, Biometrika **60**. 1973.
- [42] 97-109 W. K. Hastings Biometrika **57**. 1970. 1970.

Accept/Reject Method, 63
additive generators, 54
Arrhenius law, 46
autocorrelation function, 76
Constant Pressure Molecular Dynamics, 38
Constant Temperature Molecular Dynamics, 35
damped-force method, 35
diffusion matrix, 45
enthalpy, 38
Error Analysis, 76
Event-Driven Molecular Dynamics, 43
Force Fields, 9
Gaussian isokinetic MD, 35
generalized shift-register generator, 59
Gibbs-Sampler, 65
Hamiltonian Monte Carlo, 78
Hybrid Monte Carlo, 78
lagged Fibonacci generator, 57
Langevin Dynamics, 44
Langevin equation, 45
linear congruential generators, 53
Liouville theorem, 20
Markov Chain Monte Carlo, 67
Metropolis-Hastings Monte Carlo, 71
Molecular Dynamics, 17
Molecular Dynamics (MD), 17
Monte Carlo Method, 49
multiplicative random number generator, 54
period, 54
Random Numbers, 51
rejection-free Monte Carlo, 79
symplectic, 18
symplectic, 21
Taylor expansion, 22
Verlet algorithm, 23
Wiener process, 45