

Critical Exponents of the Ising Model

In the vicinity of the critical temperature for $T < T_c$, the spontaneous magnetization scales as

$$M_S(T) \propto (T_c - T)^\beta. \quad (16)$$

For $T = T_c$ and $H \rightarrow 0$, we find the following scaling

$$M(T = T_c, H) \propto H^{1/\delta}. \quad (17)$$

The exponents β and δ are so-called *critical exponents* and characterize together with other exponents the underlying phase transition.

- 2D: $\beta = 1/8$ and $\delta = 15$
- 3D: $\beta = 0.326$ and $\delta = 4.790$

Similarly to the power-law scaling of the spontaneous magnetization defined in Eq. (16), we find for the magnetic susceptibility in the vicinity of T_c

$$\chi(T) \propto |T_c - T|^{-\gamma} \quad (23)$$

$$C(T) \propto |T_c - T|^{-\alpha}, \quad (24)$$

- 2D: $\gamma = 7/4$ and $\alpha = 0^1$
- 3D: $\gamma \approx 1.24$ and $\alpha \approx 0.11$

¹An exponent of $\alpha = 0$ corresponds to a logarithmic decay since $\lim_{s \rightarrow 0} \frac{|x|^{-s}-1}{s} = -\ln|x|$.

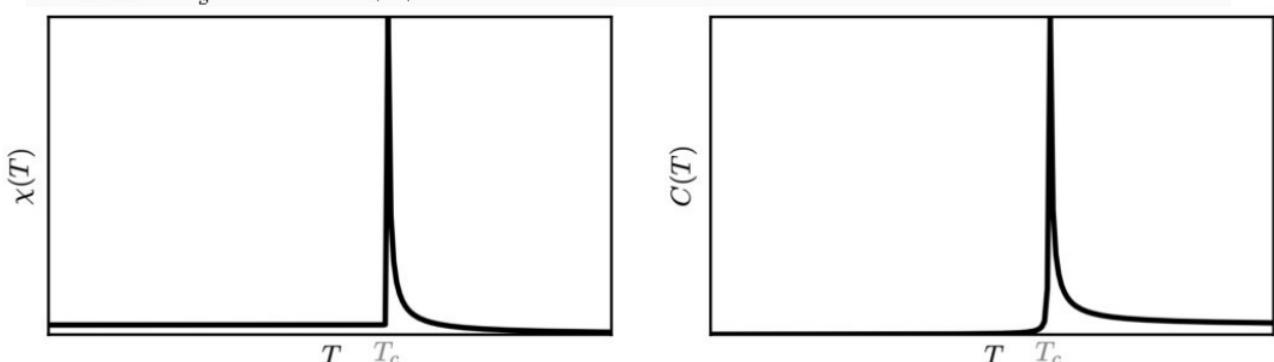


Figure 5: Susceptibility and specific heat as a function of temperature for the three dimensional Ising model. Both quantities diverge at the critical temperature T_c in the thermodynamic limit. [Böttcher,

Critical exponents and universality

The aforementioned six critical exponents are connected by four scaling laws

$$\alpha + 2\beta + \gamma = 2 \quad (\text{Rushbrooke}), \quad (29)$$

$$\gamma = \beta(\delta - 1) \quad (\text{Widom}), \quad (30)$$

$$\gamma = (2 - \eta)\nu \quad (\text{Fisher}), \quad (31)$$

$$2 - \alpha = d\nu \quad (\text{Josephson}), \quad (32)$$

which have been derived in the context of the phenomenological scaling theory for ferromagnetic systems. Due to these relations, the number of independent exponents reduces to two.

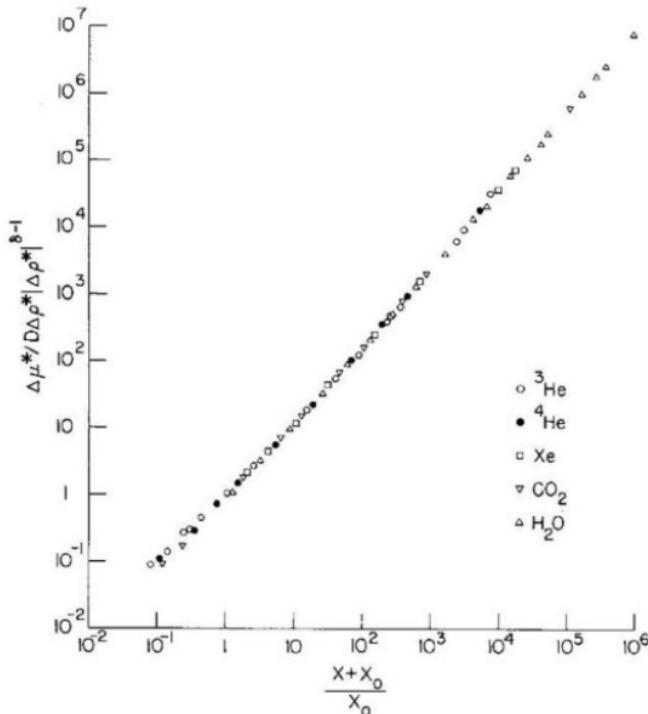


Figure 6: Universal scaling for five different gases. The scaling variable is defined as $x = \Delta T |\Delta \rho|^{-1/\beta}$ and x_0 depends on the amplitude B of the power-law for the coexistence curve $\Delta \rho = B \Delta T^\beta$

Table 1: The critical exponents of the Ising model in two and three dimensions [Pelissetto, Vicari, Phys. Rep. 368, 549–727 (2002)]

Exponent	$d = 2$	$d = 3$
α	0	0.110(1)
β	1/8	0.3265(3)
γ	7/4	1.2372(5)
δ	15	4.789(2)
η	1/4	0.0364(5)
ν	1	0.6301(4)

In terms of a Markov chain, the transition probability from one state to another is given by the probability of a new state to be proposed (T) and the probability of this state to be accepted (A). Namely, $T(X \rightarrow Y)$ is the probability that a new configuration Y is proposed, starting from configuration X . The transition probability fulfills three conditions:

1. *Ergodicity*: any configuration in the phase space must be reachable within a finite number of steps,
2. *Normalization*: $\sum_Y T(X \rightarrow Y) = 1$,
3. *Reversibility*: $T(X \rightarrow Y) = T(Y \rightarrow X)$.

Once a configuration is proposed, we can accept the new configuration with probability $A(X \rightarrow Y)$ or reject it with probability $1 - A(X \rightarrow Y)$. The *probability of the Markov chain* is then given by

$$W(X \rightarrow Y) = T(X \rightarrow Y) \cdot A(X \rightarrow Y). \quad (1)$$

We denote the probability to find the system in a certain configuration X at virtual time τ by $p(X, \tau)$. The *master equation* describes the time evolution of $p(X, \tau)$ and is given by

$$\frac{dp(X, \tau)}{d\tau} = \sum_Y p(Y)W(Y \rightarrow X) - \sum_Y p(X)W(X \rightarrow Y). \quad (2)$$

A stationary state p_{st} is reached if $\frac{dp(X, \tau)}{d\tau} = 0$. The probability of the Markov chain fulfills the following properties:

1. *Ergodicity*: any configuration must be reachable: $\forall X, Y : W(X \rightarrow Y) \geq 0$,
2. *Normalization*: $\sum_Y W(X \rightarrow Y) = 1$,
3. *Homogeneity*: $\sum_Y p_{\text{st}}(Y)W(Y \rightarrow X) = p_{\text{st}}(X)$.

Markov Chains stationary states

A stationary state p_{st} is reached if $\frac{dp(X,\tau)}{d\tau} = 0$. The probability of the Markov chain fulfills the following properties:

1. *Ergodicity*: any configuration must be reachable: $\forall X, Y : W(X \rightarrow Y) \geq 0$,
2. *Normalization*: $\sum_Y W(X \rightarrow Y) = 1$,
3. *Homogeneity*: $\sum_Y p_{\text{st}}(Y)W(Y \rightarrow X) = p_{\text{st}}(X)$.

It then follows from the stationary state condition ($\frac{dp(X,\tau)}{d\tau} = 0$) that

$$\sum_Y p_{\text{eq}}(Y)W(Y \rightarrow X) = \sum_Y p_{\text{eq}}(X)W(X \rightarrow Y).$$

A sufficient condition for this to be true is

$$p_{\text{eq}}(Y)W(Y \rightarrow X) = p_{\text{eq}}(X)W(X \rightarrow Y), \quad (4)$$

which is referred to as a *detailed balance condition*.

As an example, in a canonical ensemble at fixed Temperature T , the equilibrium distribution is given by the Boltzmann factor

$$p_{\text{eq}}(X) = \frac{1}{Z_T} \exp \left[-\frac{E(X)}{k_B T} \right] \quad (5)$$

with the partition function $Z_T = \sum_X \exp \left[-\frac{E(X)}{k_B T} \right]$.

$M(RT)^2$ algorithm

One possible choice of the acceptance probability fulfilling the detailed balance condition is given by

$$A(X \rightarrow Y) = \min \left[1, \frac{p_{\text{eq}}(Y)}{p_{\text{eq}}(X)} \right]. \quad (6)$$

which can be obtained by rewriting Eq. (4).

In the case of the canonical ensemble with

$p_{\text{eq}}(X) = \frac{1}{Z_T} \exp \left[-\frac{E(X)}{k_B T} \right]$, the acceptance probability becomes

$$A(X \rightarrow Y) = \min \left[1, \exp \left(-\frac{\Delta E}{k_B T} \right) \right], \quad (7)$$

where $\Delta E = E(Y) - E(X)$. The last equation implies that the step is always accepted if the energy decreases, and if the energy increases, it is accepted with probability $\exp \left(-\frac{\Delta E}{k_B T} \right)$.

In summary, the steps of the $M(RT)^2$ algorithm applied to the Ising model are

M(RT)² algorithm

- Randomly choose a lattice site i ,
- Compute $\Delta E = E(Y) - E(X) = 2J\sigma_i h_i$,
- Flip the spin if $\Delta E \leq 0$, otherwise accept it with probability $\exp \left(-\frac{\Delta E}{k_B T} \right)$,

with $h_i = \sum_{\langle i,j \rangle} \sigma_j$ and $E = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j$.

Glauber dynamics general

The Metropolis algorithm is not the only possible choice to fulfill the detailed balance condition. Another acceptance probability given by

$$A_G(X \rightarrow Y) = \frac{\exp\left(-\frac{\Delta E}{k_B T}\right)}{1 + \exp\left(-\frac{\Delta E}{k_B T}\right)} \quad (8)$$

has been suggested by Glauber in 1963.

In contrast to the M(RT)² acceptance probability, updates with $\Delta E = 0$ are not always accepted but with probability 1/2.

To prove that Eq. (8) satisfies the condition of detailed balance, we have to show that

$$p_{\text{eq}}(Y)A_G(Y \rightarrow X) = p_{\text{eq}}(X)A_G(X \rightarrow Y) \quad (9)$$

since $T(Y \rightarrow X) = T(X \rightarrow Y)$.

The previous equation is equivalent to

$$\frac{p_{\text{eq}}(Y)}{p_{\text{eq}}(X)} = \frac{A_G(X \rightarrow Y)}{A_G(Y \rightarrow X)} \quad (10)$$

which is fulfilled since

$$\frac{p_{\text{eq}}(Y)}{p_{\text{eq}}(X)} = \exp\left(-\frac{\Delta E}{k_B T}\right) \quad (11)$$

and

$$\frac{A_G(X \rightarrow Y)}{A_G(Y \rightarrow X)} = \frac{\exp\left(-\frac{\Delta E}{k_B T}\right)}{1 + \exp\left(-\frac{\Delta E}{k_B T}\right)} \left[\frac{\exp\left(\frac{\Delta E}{k_B T}\right)}{1 + \exp\left(\frac{\Delta E}{k_B T}\right)} \right]^{-1} = \exp\left(-\frac{\Delta E}{k_B T}\right) \quad (12)$$

Glauber dynamics for the Ising model

As in the M(RT)² algorithm, only the local configuration around the lattice site is relevant for the update procedure.

Furthermore, with $J = 1$, the probability to flip spin σ_i is

$$A_G(X \rightarrow Y) = \frac{\exp\left(\frac{-2\sigma_i h_i}{k_B T}\right)}{1 + \exp\left(\frac{-2\sigma_i h_i}{k_B T}\right)} \quad (13)$$

with $h_i = \sum_{\langle i,j \rangle} \sigma_j$ being the local field and $X = \{\dots, \sigma_{i-1}, \sigma_i, \sigma_{i+1}, \dots\}$ and $Y = \{\dots, \sigma_{i-1}, -\sigma_i, \sigma_{i+1}, \dots\}$ the initial and final configuration, respectively.

We abbreviate the probability defined by Eq. (13) as p_i . The spin flip and no flip probabilities can then be expressed as

$$p_{\text{flip}} = \begin{cases} p_i & \text{for } \sigma_i = -1 \\ 1 - p_i & \text{for } \sigma_i = +1 \end{cases} \quad \text{and} \quad p_{\text{no-flip}} = \begin{cases} 1 - p_i & \text{for } \sigma_i = -1 \\ p_i & \text{for } \sigma_i = +1 \end{cases} \quad (14)$$

A possible implementation is

$$\sigma_i(\tau + 1) = -\sigma_i(\tau) \cdot \text{sign}(p_i - z), \quad (15)$$

with $z \in (0, 1)$ being a uniformly distributed random number, or

$$\sigma_i(\tau+1) = \begin{cases} +1 & \text{with probability } p_i \\ -1 & \text{with probability } 1 - p_i \end{cases} \quad \text{and} \quad p_i = \frac{\exp(2\beta h_i)}{1 + \exp(2\beta h_i)}. \quad (16)$$

This method does not depend on the spin value at time t and is called *heat-bath Monte Carlo*.

Creutz Algorithm

Figure 3: An algorithm to perform microcanonical Monte Carlo simulations, i.e., system at constant energy.

The movement in phase space is in fact not strictly constrained to a subspace of constant energy but there is a certain additional volume in which we can freely move. The condition often constant energy is softened by introducing a so-called *demon* which corresponds to a small reservoir of energy E_D that can store a certain maximum energy E_{\max} .

Creutz algorithm

- Choose a site,
- Compute ΔE for the spin flip,
- Accept the change if $E_{\max} \geq E_D - \Delta E \geq 0$.

Pro: Besides the fact that we can randomly choose a site, this method involves no random numbers and is thus said to be completely deterministic and therefore reversible.

Con: The temperature of the system is not known.

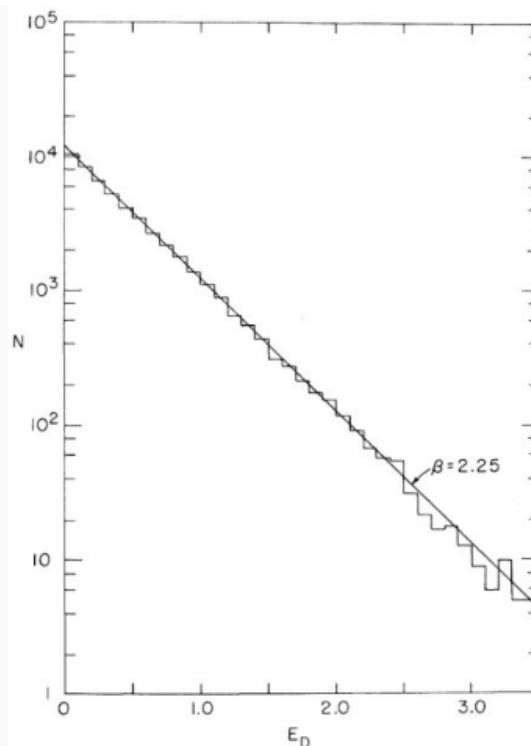


Figure 4: The distribution of the demon energy E_D is exponentially distributed. Based on the Boltzmann factor, it is possible to extract the inverse temperature $\beta = (k_B T)^{-1} = 2.25$. The figure is taken from Ref. shown in Figure 3.

Temporal Correlations

According to the definition of a Markov chain, the dependence of a quantity A on virtual time τ is given by

$$\langle A(\tau) \rangle = \sum_X p(X, \tau) A(X) = \sum_X p(X, \tau_0) A(X(\tau)). \quad (17)$$

In the second step of the latter equation, we used the fact that the average is taken over an ensemble of initial configurations $X(\tau_0)$ which evolve according to Eq. (2).

For some $\tau_0 < \tau$, the *non-linear correlation function*

$$\Phi_A^{\text{nl}}(\tau) = \frac{\langle A(\tau) \rangle - \langle A(\infty) \rangle}{\langle A(\tau_0) \rangle - \langle A(\infty) \rangle} \quad (18)$$

is a measure to quantify the deviation of $A(\tau)$ from $A(\infty)$ relative to the deviation of $A(\tau_0)$ from $A(\infty)$.

The linear correlation function of two values A, B is defined as

$$\Phi_{AB}(\tau) = \frac{\langle A(\tau_0)B(\tau) \rangle - \langle A \rangle \langle B \rangle}{\langle AB \rangle - \langle A \rangle \langle B \rangle} \quad (22)$$

with

$$\langle A(\tau_0)B(\tau) \rangle = \sum_X p(X, \tau_0) A(X(\tau_0)) B(X(\tau)).$$

As τ goes to infinity, $\Phi_{AB}(\tau)$ decreases from unity to zero.

If $A = B$, we call Eq. (22) the *autocorrelation function*. For the spin-spin correlation in the Ising model we obtain

$$\Phi_\sigma(\tau) = \frac{\langle \sigma(\tau_0)\sigma(\tau) \rangle - \langle \sigma(\tau_0) \rangle^2}{\langle \sigma^2(\tau_0) \rangle - \langle \sigma(\tau_0) \rangle^2}$$

Non-Linear Correlation

The *non-linear* correlation time τ_A^{nl} describes the relaxation towards equilibrium and is defined as¹

$$\tau_A^{\text{nl}} = \int_0^\infty \Phi_A^{\text{nl}}(\tau) d\tau. \quad (20)$$

¹If we consider an exponential decay of $\Phi_A^{\text{nl}}(\tau)$, we find that this definition is meaningful since

$$\int_0^\infty \exp\left(-\tau/\tau_A^{\text{nl}}\right) d\tau = \tau_A^{\text{nl}}. \quad (19)$$

For some $\tau_0 < \tau$, the *non-linear correlation function*

$$\Phi_A^{\text{nl}}(\tau) = \frac{\langle A(\tau) \rangle - \langle A(\infty) \rangle}{\langle A(\tau_0) \rangle - \langle A(\infty) \rangle} \quad (18)$$

is a measure to quantify the deviation of $A(\tau)$ from $A(\infty)$ relative to the deviation of $A(\tau_0)$ from $A(\infty)$.

In the vicinity of the critical temperature T_c , we observe the so-called *critical slowing down* of our dynamics, i.e., the non-linear correlation time is described by power law

$$\tau_A^{\text{nl}} \sim |T - T_c|^{-z_A^{\text{nl}}} \quad (21)$$

with z_A^{nl} being the non-linear dynamical critical exponent. This implies that the time needed to reach equilibrium diverges at T_c !

Linear Correlation

The linear correlation function of two values A, B is defined as

$$\Phi_{AB}(\tau) = \frac{\langle A(\tau_0)B(\tau) \rangle - \langle A \rangle \langle B \rangle}{\langle AB \rangle - \langle A \rangle \langle B \rangle} \quad (22)$$

with

$$\langle A(\tau_0)B(\tau) \rangle = \sum_X p(X, \tau_0) A(X(\tau_0)) B(X(\tau)).$$

As τ goes to infinity, $\Phi_{AB}(\tau)$ decreases from unity to zero. The *linear* correlation time τ_A^{nl} describes the relaxation towards equilibrium

$$\tau_{AB} = \int_0^\infty \Phi_{AB}(\tau) d\tau. \quad (23)$$

As in the case of the non-linear correlation time, in the vicinity of T_c , we observe a *critical slowing down*, i.e.,

$$\tau_{AB} \sim |T - T_c|^{-z_A}. \quad (24)$$

with z_A being the *linear* dynamical critical exponent.

The dynamical exponents for spin correlations turn out to be

$$z_\sigma = 2.16 \text{ (2D)},$$

$$z_\sigma = 2.09 \text{ (3D)}.$$

There is a conjectured relation between the Ising critical exponents and the critical dynamical exponents for spin σ and energy correlations E . The relations

$$z_\sigma - z_\sigma^{\text{nl}} = \beta, \quad (25)$$

$$z_E - z_E^{\text{nl}} = 1 - \alpha, \quad (26)$$

$$(27)$$

are numerically well-established, however, not yet analytically proven.

Decorrelated configurations

Connecting this behavior with the one observed for the correlation time described by Eq. (23) yields

$$\tau_{AB} \sim |T - T_c|^{-z_{AB}} \sim L^{\frac{z_{AB}}{\nu}} \quad (28)$$

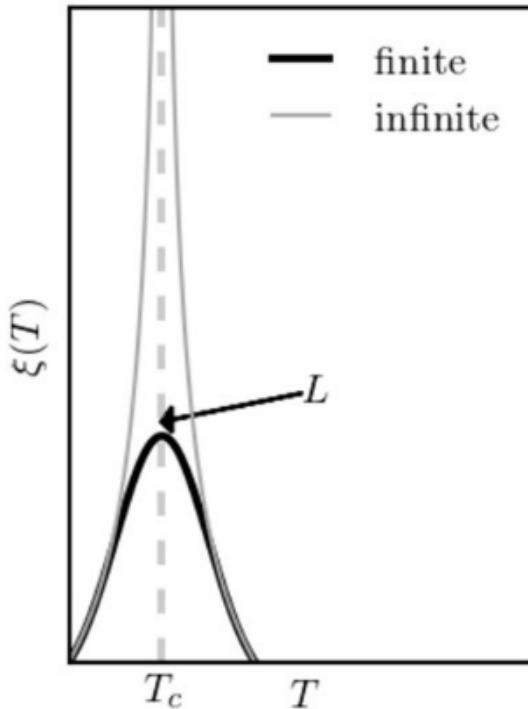


Figure 5: The correlation length diverges in an infinite system at T_c according to the definition of the correlation length from last week's lecture. In a finite system, however, we observe a round off and the correlation length approaches the system size L at T_c .

To ensure not to sample correlated configurations one should

- first reach equilibrium (discard $n_0 = c\tau^{\text{nl}}(T)$ configurations),
- only sample every $n_e^{th} = c\tau(T)$ configuration,
- and at T_c use $n_0 = cL^{\frac{z_{\text{nl}}}{\nu}}$ and $n_e = cL^{\frac{z}{\nu}}$

where $c \approx 3$ is a "safety factor" to make sure to discard enough samples.

Finite size methods

Divergent behavior at T_c as described by

$$\chi(T) \sim |T_c - T|^{-\gamma} \quad (1)$$

$$C(T) \sim |T_c - T|^{-\alpha}, \quad (2)$$

$$\xi(T) \sim |T - T_c|^{-\nu} \quad (3)$$

The larger the system size, the more pronounced is the divergence.

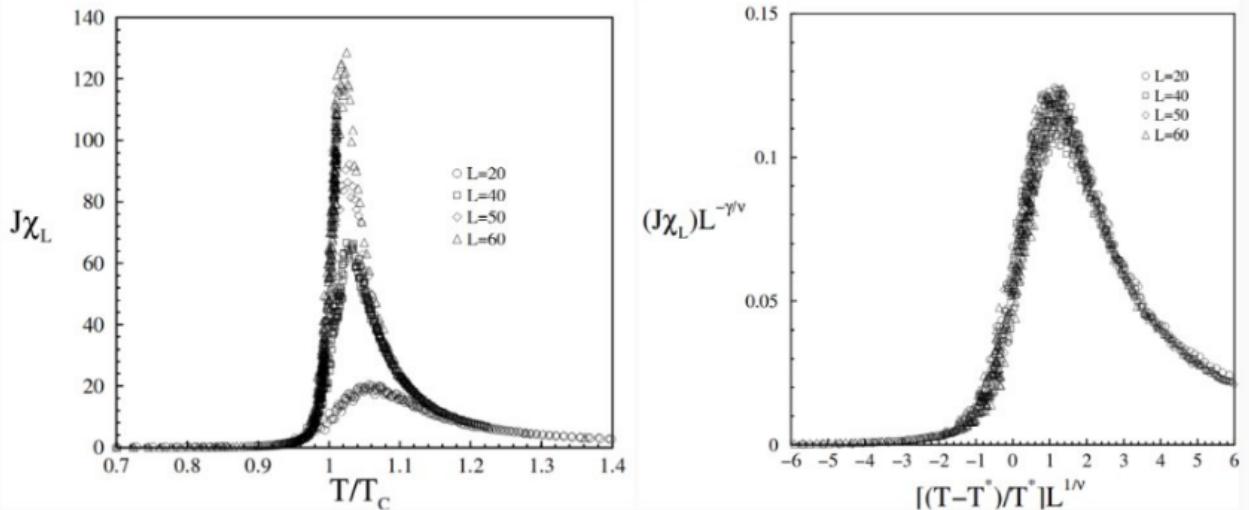


Figure 3: The system size dependence of the susceptibility and the corresponding finite size scaling. The figure is taken from [Da Silva et al., Braz. J. Phys. 32, 2002].

The finite size scaling relation of the susceptibility is given by

$$\chi(T, L) = L^{\frac{\gamma}{\nu}} F_\chi \left[(T - T_c) L^{\frac{1}{\nu}} \right], \quad (4)$$

where F_χ is called susceptibility *scaling function*¹.

¹Based on Eq. (1), we can infer that $F_\chi \left[(T - T_c) L^{\frac{1}{\nu}} \right] \sim (|T - T_c| L^{\frac{1}{\nu}})^{-\gamma}$ as $L \rightarrow \infty$.

In the case of the magnetization, the corresponding finite size scaling relation is

$$M_S(T, L) = L^{-\frac{\beta}{\nu}} F_{M_S} \left[(T - T_c) L^{\frac{1}{\nu}} \right]. \quad (5)$$

Binder Cumulant

We still need a way to determine T_c more precisely. To do that, we make use of the so-called *Binder cumulant*

$$U_L = 1 - \frac{\langle M^4 \rangle_L}{3 \langle M^2 \rangle_L^2}, \quad (6)$$

which is independent of the system size L at T_c since

$$\frac{\langle M^4 \rangle_L}{3 \langle M^2 \rangle_L^2} = \frac{L^{-\frac{4\beta}{\nu}} F_{M^4} \left[(T - T_c) L^{\frac{1}{\nu}} \right]}{\left\{ L^{-\frac{2\beta}{\nu}} F_{M^2} \left[(T - T_c) L^{\frac{1}{\nu}} \right] \right\}^2} = F_C \left[(T - T_c) L^{\frac{1}{\nu}} \right]. \quad (7)$$

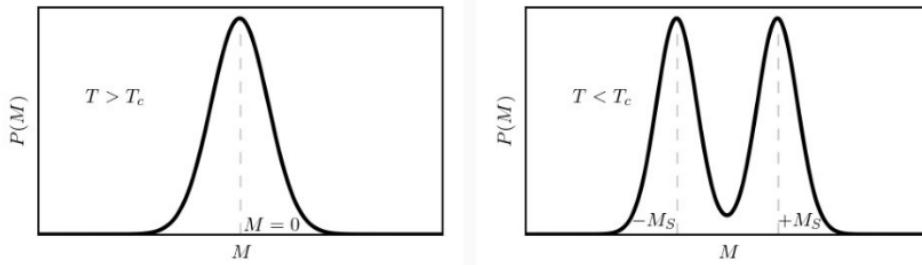


Figure 4: The distribution $P(M)$ of the magnetization M above and below the critical temperature T_c .

Below the critical temperature ($T < T_c$), there exist one ground state with positive and one with negative magnetization and the corresponding distribution is given by

$$P_L(M) = \frac{1}{2} \sqrt{\frac{L^d}{\pi \sigma_L}} \left\{ \exp \left[-\frac{(M - M_S)^2 L^d}{\sigma_L} \right] + \exp \left[-\frac{(M + M_S)^2 L^d}{\sigma_L} \right] \right\} \quad (10)$$

For $T > T_c$, the magnetization is described by a Gaussian distribution

$$P_L(M) = \sqrt{\frac{L^d}{\pi \sigma_L}} \exp \left[-\frac{M^2 L^d}{\sigma_L} \right], \quad (8)$$

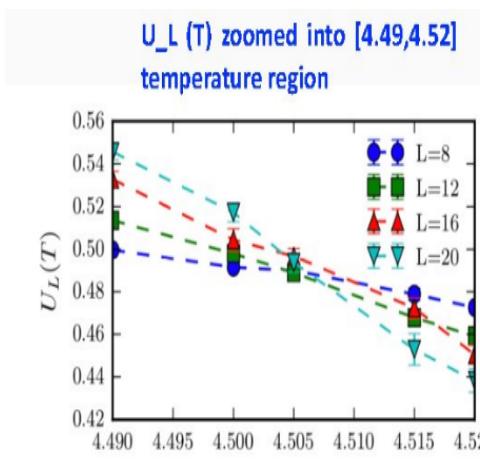
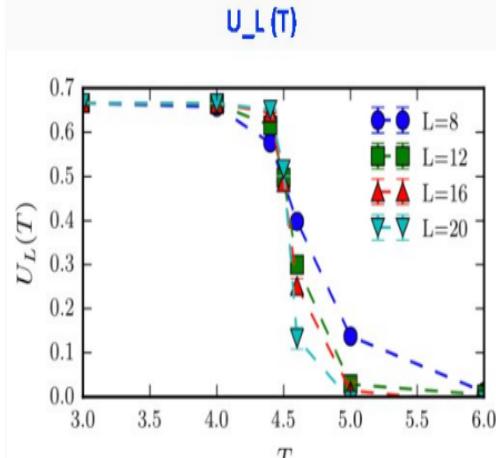
with $\sigma_L = 2k_B T \chi_L$. Since the fourth moment equals three times the second moment squared, i.e.,

$$\langle M^4 \rangle = 3 \langle M^2 \rangle_L^2, \quad (9)$$

it follows that U_L must be zero for $T > T_c$.

For this distribution, it holds that $\langle M^4 \rangle = \langle M^2 \rangle_L^2$ and therefore $U_L = \frac{2}{3}$. In summary, we demonstrated that

$$U_L = \begin{cases} \frac{2}{3} & \text{for } T < T_c \\ \text{const} & \text{for } T = T_c \\ 0 & \text{for } T > T_c \end{cases} \quad (11)$$



First Order Transition

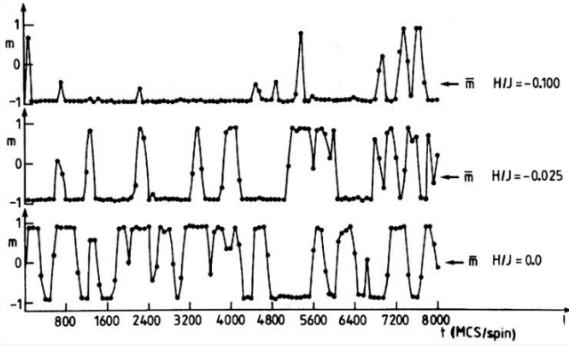


Figure 5: The magnetization exhibits a switching behavior if the field vanishes. For non-zero magnetic fields, the magnetization is driven in the direction of the field. The figure is taken from [Binder and Landau, Phys. Rev. B30 3 (1984)]

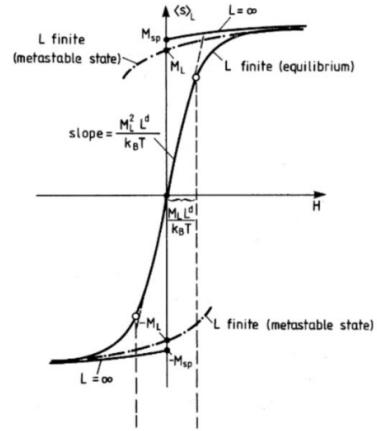


Figure 6: Hysteresis can be found by varying the field from negative to positive values and back. The figure is taken from [Binder and Landau, Phys. Rev. B30 3 (1984)]

Binder showed that the magnetization as a function of the field H is described by $\tanh(\alpha L^d)$ if the distribution of the magnetization is given by Eq. (10). Specifically, we find for the magnetization and susceptibility

$$M(H) = \chi_L^D H + M_L \tanh\left(\beta H M_L L^d\right), \quad (15)$$

$$\chi_L(H) = \frac{\partial M}{\partial H} = \chi_L^D + \frac{\beta M_L L^d}{\cosh^2(\beta H M_L L^d)}. \quad (16)$$

Similarly, to the scaling of a second order transition, we can scale the maximum of the susceptibility ($\chi_L(H=0) \sim L^d$) and the width of the peak ($\Delta\chi_L \sim L^{-d}$). To summarize, a first order phase transition is characterized by

1. A bimodal distribution of the order parameter,
2. stochastic switching between the two states in small systems,
3. hysteresis of the order parameter when changing the field,
4. a scaling of the order parameter, or response function according to Eq. (16).

The Kasteleyn and Fortuin Theorem

We consider the Potts model not on a square lattice but on an arbitrary graph of nodes connected with bonds ν . Each node has q possible states and each connection leads to an energy cost of unity if two connected nodes are in a different state and of zero if they are in the same state, i.e.,

$$E = J \sum_{\nu} \epsilon_{\nu} \quad \text{with} \quad \epsilon_{\nu} = \begin{cases} 0 & \text{if endpoints are in the same state} \\ 1 & \text{otherwise} \end{cases} \quad (18)$$

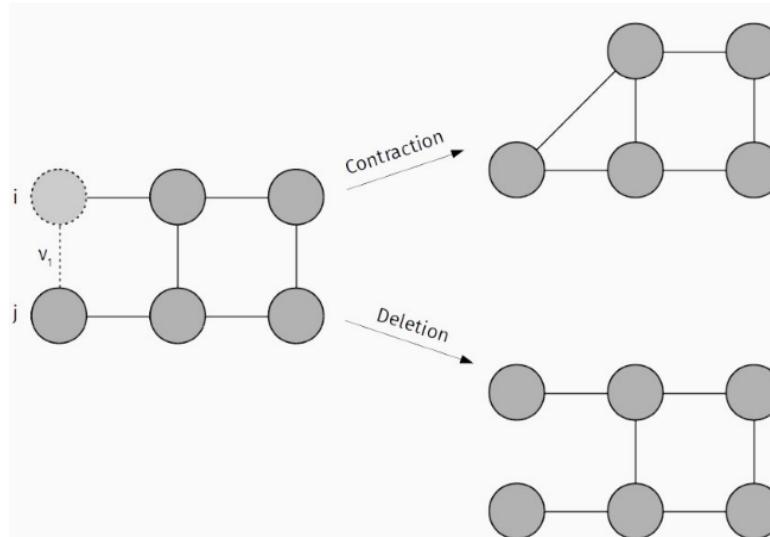


Figure 8: Contraction and deletion on a graph.

The partition function is the sum over all the possible configurations weighted by the Boltzmann factor and thus given by

$$Z = \sum_X e^{-\beta E(X)} \stackrel{(18)}{=} \sum_X e^{-\beta J \sum_{\nu} \epsilon_{\nu}} = \sum_X \prod_{\nu} e^{-\beta J \epsilon_{\nu}}. \quad (19)$$

After applying these operations to every bond, the graph is reduced to a set of separated points corresponding to clusters of nodes which are connected and in the same state out of q states. The partition function reduces to

$$Z = \sum_{\substack{\text{configurations of} \\ \text{bond percolation}}} q^{\# \text{ of clusters}} p^c (1-p)^d = \left\langle q^{\# \text{ of clusters}} \right\rangle_b, \quad (24)$$

where c and d are the numbers of contracted and deleted bonds respectively. In the limit of $q \rightarrow 1$, one obtains the partition function of bond percolation².

²In bond percolation [Broadbent, Hammersley (1957)], an edge of a graph is occupied with probability p and vacant with probability $1 - p$.

Coniglio-Klein clusters

The probability of a given cluster C to be in a certain state σ_0 is independent of the state itself, i.e.,

$$p(C, \sigma_0) = p^{c_C} (1 - p)^{d_C} \sum_{\substack{\text{bond percolation} \\ \text{without cluster } C}} q^{\# \text{ of clusters}} p^c (1 - p)^d. \quad (25)$$

This implies that flipping this particular cluster has no effect on the partition function (and therefore the energy) so that it is possible to accept the flip with probability one. This can be seen by looking at the detailed balance condition of the system

$$p(C, \sigma_1) W [(C, \sigma_1) \rightarrow (C, \sigma_2)] = p(C, \sigma_2) W [(C, \sigma_2) \rightarrow (C, \sigma_1)] \quad (26)$$

and using $p(C, \sigma_1) = p(C, \sigma_2)$.

We then obtain for acceptance probabilities

$$W[(C, \sigma_2) \rightarrow (C, \sigma_1)] = \frac{p(C, \sigma_2)}{p(C, \sigma_1) + p(C, \sigma_2)} = \frac{1}{2} \quad \text{Glauber dyn} \quad (27)$$

$$W[(C, \sigma_2) \rightarrow (C, \sigma_1)] = \min \left[1, \frac{p(C, \sigma_2)}{p(C, \sigma_1)} \right] = 1 \quad \text{Metropolis} \quad (28)$$

Based on these insights, we introduce cluster algorithms which are much faster than single-spin flip algorithms and less prone to the problem of critical slowing down.

Other Ising-like models

One of the possible generalizations of the Ising model is the so called n -vector model. Unlike the Potts model, it describes spins as vectors with n components. This model has applications in modelling magnetism or the Higgs mechanism. The Hamiltonian resembles the one of the Potts model in the sense that it favors spin alignment

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + \vec{H} \sum_i \vec{S}_i. \quad (11)$$

with $\vec{S}_i = (S_i^1, S_i^2, \dots, S_i^n)$ and $|\vec{S}_i| = 1$.

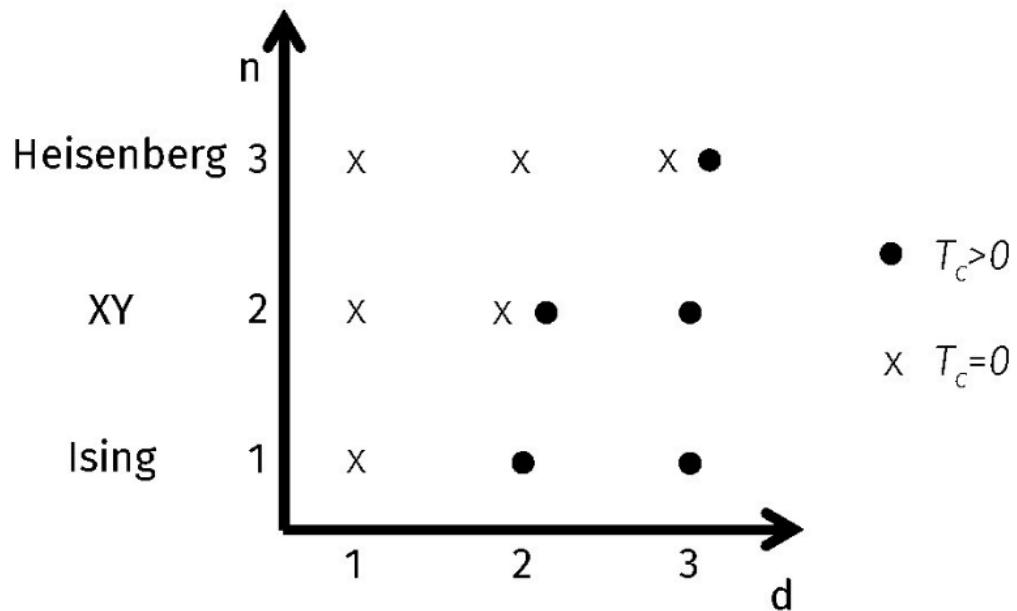


Figure 3: The dependence of the critical temperature on the number of vector components n .

For Monte Carlo simulations with vector-valued spins we have to adapt our simulation methods. The classical strategy is to flip spins by modifying the spin locally through adding a small $\Delta \vec{S}$ such that $\vec{S}'_i = \vec{S}_i + \Delta \vec{S}$ and $\Delta \vec{S} \perp \vec{S}_i$. The classical Metropolis algorithm can then be used in the same fashion as in the Ising model.

Histogram Methods

For computing the thermal average

$$\langle Q \rangle = \frac{1}{Z_T} \sum_X Q(X) e^{-\frac{E(X)}{k_B T}}. \quad (12)$$

we need to sample different configurations at different temperatures. Another possibility would be to determine an average at a certain temperature T_0 and extrapolate to another temperature T . In the case of a canonical ensemble, an extrapolation can be achieved by reweighting the histogram of energies $p_{T_0}(E)$ with the Boltzmann factor $e^{\frac{E}{T} - \frac{E}{T_0}}$.

Such histogram methods have first been described by Salzburg et al. in 1959. We now reformulate the computation of the thermal average of a quantity Q and of the partition function as a sum over all possible energies instead of over all possible configurations and find

$$Q(T_0) = \frac{1}{Z_{T_0}} \sum_E Q(E) p_{T_0}(E) \quad \text{with} \quad Z_{T_0} = \sum_E p_{T_0}(E), \quad (13)$$

where $p_{T_0}(E) = g(E) e^{-\frac{E}{k_B T_0}}$ with $g(E)$ defining the *degeneracy of states*, i.e., the number of states with energy E .

This takes into account the fact that multiple configurations can have the same energy. The goal is to compute the quantity Q at another temperature T

$$Q(T) = \frac{1}{Z_T} \sum_E Q(E) p_T(E). \quad (14)$$

The degeneracy of states contains all the information needed.

Using the definition of $g(E)$ yields

$$p_T(E) = g(E) e^{-\frac{E}{k_B T}} = p_{T_0}(E) \exp \left[-\frac{E}{k_B T} + \frac{E}{k_B T_0} \right] \quad (15)$$

and with $f_{T_0,T}(E) = \exp \left[-\frac{E}{k_B T} + \frac{E}{k_B T_0} \right]$ we finally obtain

$$Q(T) = \frac{\sum_E Q(E) p_{T_0}(E) f_{T_0,T}(E)}{\sum_E p_{T_0}(E) f_{T_0,T}(E)}. \quad (16)$$

Broad histogram methods

Let N_{up} and N_{down} be the numbers of processes which lead to an increasing and decreasing energy, respectively. Furthermore, we have to keep in mind that the degeneracy of states increases exponentially with energy E since the number of possible configurations increases with energy. To explore all energy regions equally, we find a condition equivalent to the one of detailed balance, i.e.,

$$g(E + \Delta E) N_{\text{down}}(E + \Delta E) = g(E) N_{\text{up}}(E). \quad (17)$$

The motion in phase space towards higher energies can then be penalized with a Metropolis-like dynamics:

- Choose a new configuration,
- if the new energy is lower, accept the move,
- if the new energy is higher then accept with probability $\frac{N_{\text{down}}(E + \Delta E)}{N_{\text{up}}(E)}$.

We obtain the function $g(E)$ by taking the logarithm of Eq. (17) and divide by ΔE

$$\log [g(E + \Delta E)] - \log [g(E)] = - \log [N_{\text{up}}(E)] - \log [N_{\text{down}}(E + \Delta E)]. \quad (18)$$

In the limit of small energy differences, we can approximate the latter equation by

$$\frac{\partial \log [g(E)]}{\partial E} = \frac{1}{\Delta E} \log \left[\frac{N_{\text{up}}(E)}{N_{\text{down}}(E + \Delta E)} \right] \quad (19)$$

which we can numerically integrate to obtain $g(E)$.

Distributions of N_{up} and N_{down} can be obtained by keeping track of these numbers for each configuration at a certain energy. In addition, we also need to store the values of the quantity $Q(E)$ we wish to compute as a thermal average according to

$$Q(T) = \frac{\sum_E Q(E) g(E) e^{-\frac{E}{k_B T}}}{\sum_E g(E) e^{-\frac{E}{k_B T}}}. \quad (20)$$

Based on a known degeneracy of states $g(E)$, we can now compute quantities at any temperature.

Renormalization and free energy

To build some intuition for renormalization approaches, we consider a scale transformation of the characteristic length L of our system with that leads to a rescaled characteristic length $\tilde{L} = L/l$

Moreover, we consider the partition function of an Ising system. A scale transformation with $\tilde{L} = L/l$ leaves the partition function

$$Z = \sum_{\{\sigma\}} e^{-\beta H} \quad (1)$$

and the corresponding free energy invariant.

Free energy density of the system also stays invariant under scale transformations. Since the free energy F is an *extensive* quantity¹, it scales with the system size and

$$F(\epsilon, H) = l^{-d} \tilde{F}(\tilde{\epsilon}, \tilde{H}) \quad \text{with} \quad \epsilon = T - T_c, \quad (2)$$

where \tilde{F} is the renormalized free energy.

We can rescale previous equation by setting

$$\tilde{\epsilon} = l^{y_T} \epsilon \quad \text{and} \quad \tilde{H} = l^{y_H} H \quad (3)$$

and obtain in terms of the renormalized free energy

$$\tilde{F}(\tilde{\epsilon}, \tilde{H}) = \tilde{F}(l^{y_T} \epsilon, l^{y_H} H). \quad (4)$$

Since renormalization also affects the correlation length

$$\xi \sim |T - T_c|^{-\nu} = |\epsilon|^{-\nu} \quad (5)$$

we can relate the critical exponent ν to y_T .

The renormalized correlation length $\tilde{\xi} = \xi/l$ scales as

$$\tilde{\xi} \sim \tilde{\epsilon}^{-\nu}. \quad (6)$$

And due to

$$l^{y_T} \epsilon = \tilde{\epsilon} \sim \epsilon l^{\frac{1}{\nu}}, \quad (7)$$

we find $y_T = 1/\nu$.

The critical point is a fixed point of the transformation since $\epsilon = 0$ at T_c and ϵ does not change independent of the value of the scaling factor.

Decimation of the one-dimensional Ising model

Generalization

In general, multiple coupling constants are necessary, e.g., in the two-dimensional Ising model. Thus, we have to construct a renormalized Hamiltonian based on multiple renormalized coupling constants, i.e.,

$$\tilde{H} = \sum_{\alpha=1}^M \tilde{K}_\alpha \tilde{O}_\alpha \text{ with } \tilde{O}_\alpha = \sum_i \prod_{k \in c_\alpha} \tilde{\sigma}_{i+k} \quad (18)$$

where c_α is the configuration subset over which we renormalize and

$$\tilde{K}_\alpha(K_1, \dots, K_M) \quad \text{with} \quad \alpha \in \{1, \dots, M\}.$$

At T_c there exists a fixed point $K_\alpha^* = K_\alpha(K_1^*, \dots, K_M^*)$. A first ansatz to solve this problem is the linearization of the transformation. Thus, we compute the Jacobian $T_{\alpha,\beta} = \frac{\partial \tilde{K}_\alpha}{\partial K_\beta}$ and obtain

$$\tilde{K}_\alpha - K_\alpha^* = \sum_\beta T_{\alpha,\beta}|_{K^*} (K_\beta - K_\beta^*) \quad (19)$$

To analyze the behavior of the system close to criticality, we consider eigenvalues $\lambda_1, \dots, \lambda_M$ and eigenvectors ϕ_1, \dots, ϕ_M of the linearized transformation defined by Eq. (19). The eigenvectors fulfill $\tilde{\phi}_\alpha = \lambda_\alpha \phi_\alpha$ and the fixed point is unstable if $\lambda_\alpha > 1$.

The largest eigenvalue dominates the iteration and we can identify the scaling field $\tilde{\epsilon} = l^{y_T} \epsilon$ with the eigenvector of the transformation, and the scaling factor with eigenvalue $\lambda_T = l^{y_T}$. Then, we compute the exponent ν according to

$$\nu = \frac{1}{y_T} = \frac{\log(l)}{\log(\lambda_T)}. \quad (20)$$

Monte Carlo renormalization group

Since we are dealing with generalized Hamiltonians with many interaction terms, we compute the thermal average using the operators O_α , i.e.,

$$\langle O_\alpha \rangle = \frac{\sum_{\{\sigma\}} O_\alpha e^{\sum_\beta K_\beta O_\beta}}{\sum_{\{\sigma\}} e^{\sum_\beta K_\beta O_\beta}} = \frac{\partial F}{\partial K_\alpha} \quad (21)$$

where F is the free energy.

Using the fluctuation-dissipation theorem, we can also numerically calculate the response functions:

$$\begin{aligned} \chi_{\alpha,\beta} &= \frac{\partial \langle O_\alpha \rangle}{\partial K_\beta} = \langle O_\alpha O_\beta \rangle - \langle O_\alpha \rangle \langle O_\beta \rangle, \\ \tilde{\chi}_{\alpha,\beta} &= \frac{\partial \langle \tilde{O}_\alpha \rangle}{\partial K_\beta} = \langle \tilde{O}_\alpha O_\beta \rangle - \langle \tilde{O}_\alpha \rangle \langle O_\beta \rangle. \end{aligned}$$

Using the chain rule, one can calculate with equation (21) that

$$\tilde{\chi}_{\alpha,\beta}^{(n)} = \frac{\partial \langle \tilde{O}_\alpha^{(n)} \rangle}{\partial K_\beta} = \sum_\gamma \frac{\partial \tilde{K}_\gamma}{\partial K_\beta} \frac{\partial \langle \tilde{O}_\alpha^{(n)} \rangle}{\partial K_\gamma} = \sum_\gamma T_{\gamma,\beta} \chi_{\alpha,\gamma}^{(n)}.$$

It is thus possible to derive a value of $T_{\gamma,\beta}$ from the correlation functions by solving a set of M coupled linear equations. At point $K = K^*$, we can apply this method in an iterative manner to compute critical exponents as suggested by Eq. 20.

There are many error sources in this technique, that originate from the fact that we are using a combination of several tricks to obtain our results:

- Statistical errors,
- Truncation of the Hamiltonian to the M^{th} order,
- Finite number of scaling iterations,
- Finite size effects,
- No precise knowledge of K^* .

Hopfield Network

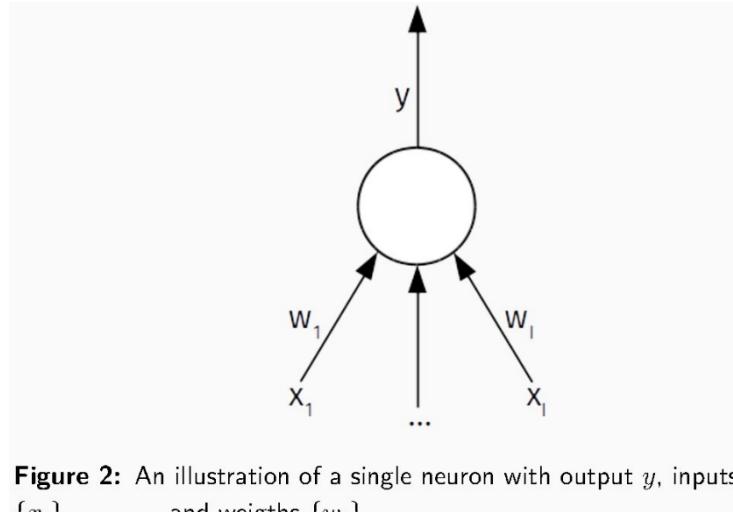


Figure 2: An illustration of a single neuron with output y , inputs $\{x_i\}_{i \in \{1, \dots, I\}}$ and weights $\{w_i\}_{i \in \{1, \dots, I\}}$.

In terms of a Hopfield network, we consider discrete inputs $x_i \in \{-1, 1\}$. The activation function of neuron i is given by

$$a_i = \sum_j w_{ij} a_j, \quad (1)$$

where we sum over the inputs. The weights fulfill $w_{ij} = w_{ji}$ and $w_{ii} = 0$.

Similarly to the Ising model, the associated energy is given by

$$E = -\frac{1}{2} \sum_{i,j} w_{ij} x_i x_j - \sum_i b_i x_i, \quad (2)$$

where b_i is bias term.

The dynamics of a Hopfield network is

$$x_i(a_i) = \begin{cases} 1 & \text{if } a_i \geq 0, \\ -1 & \text{otherwise.} \end{cases} \quad (3)$$

The energy difference ΔE_i after neuron i has been updated is

$$\Delta E_i = E(x_i = -1) - E(x_i = 1) = 2 \left(b_i + \sum_j w_{ij} x_j \right) \quad (4)$$

We can absorb the bias b_i in the sum by having an extra active unit at every node in the network. We thus showed that the activation defined by Eq. (1) amounts to the half of the energy difference ΔE_i .

Boltzmann machine learning

Molecular dynamics

To model interacting particle systems, we use generalized coordinates

$$\mathbf{q}_i = (q_i^1, \dots, q_i^d) \quad \text{and} \quad \mathbf{p}_i = (p_i^1, \dots, p_i^d). \quad (1)$$

in a system where each particle has d degrees of freedom.

The system of N particles is then described by

$$Q = (\mathbf{q}_1, \dots, \mathbf{q}_N) \quad \text{and} \quad P = (\mathbf{p}_1, \dots, \mathbf{p}_N), \quad (2)$$

using the Hamiltonian

$$\mathcal{H}(P, Q) = K(P) + V(Q) \quad (3)$$

with $K(P) = \sum_{i,k} \frac{(p_i^k)^2}{2m_i}$ being the kinetic energy, m_i the mass of the i^{th} particle and $V(Q)$ the potential energy. The sum over $k \in \{1, \dots, d\}$ accounts for the d degrees of freedom.



The potential (e.g., an attractive or repulsive electromagnetic potential) determines the mutual interactions of all particles and therefore their dynamics. An expansion of the potential energy yields:

$$V(Q) = \sum_i v_1(q_i) + \sum_i \sum_{j>i} v_2(q_i, q_j) + \sum_i \sum_{j>i} \sum_{k>j} v_3(q_i, q_j, q_k) + \dots \quad (4)$$

Typically three or more body interactions are neglected and their effect is considered in an effective two body interaction described by

$$v_2^{\text{eff}}(q_i, q_j) = v^{\text{attr}}(r) + v^{\text{rep}}(r) \quad \text{with} \quad r = |\mathbf{q}_i - \mathbf{q}_j|, \quad (5)$$

where $v^{\text{attr}}(r)$ and $v^{\text{rep}}(r)$ represent attractive and repulsive part of the effective potential, respectively.

For now, we only consider potentials that depend on distance, not particle orientation. Analytically, the simplest potential is the hard sphere interaction potential

$$v^{\text{rep}}(r) = \begin{cases} \infty & \text{if } r < \sigma, \\ 0 & \text{if } r \geq \sigma. \end{cases} \quad (6)$$

Equation of motion

The first order Taylor approximation of a symmetric attractive or repulsive potential is given by an elastic potential. For two particles with radii R_1 and R_2 , the potential is given by

$$v^{\text{rep}}(r) = \begin{cases} \frac{k}{2} (R - r)^2 & \text{if } r < R \\ 0 & \text{if } r > R \end{cases} \quad \text{with} \quad R = R_1 + R_2, \quad (7)$$

where k is the elastic spring constant.

Another very important form of potential typically used to describe the interaction between molecules is the *Lennard-Jones* potential

$$v^{\text{LJ}}(r) = 4\epsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right], \quad (8)$$

where ϵ is the attractive energy and σ the interaction range.

LJ potential approximates the spherical symmetric interaction between a pair of neutral atoms or molecules.

Once the interaction potential has been defined, we can easily derive the equations of motion using the Hamilton equations

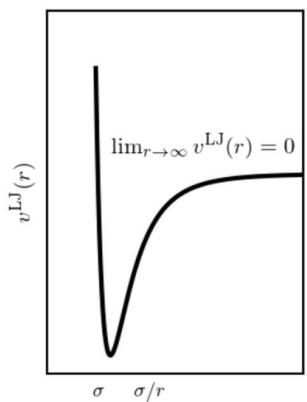
$$\dot{q}_i^k = \frac{\partial \mathcal{H}}{\partial p_i^k}, \quad \dot{p}_i^k = -\frac{\partial \mathcal{H}}{\partial q_i^k}, \quad (9)$$

where $k \in \{1, \dots, d\}$ and $i \in \{1, \dots, N\}$.

For every particle, we identify q_i with the position vector x_i and $\dot{q}_i = \dot{x}_i$ with the velocity vector $\dot{\mathbf{v}}_i$. Due to $\dot{\mathbf{x}}_i = \mathbf{v}_i = \frac{\mathbf{p}_i}{m}$ and $\dot{\mathbf{p}}_i = -\nabla V(Q) = \mathbf{f}_i$, the equations of motion are:

$$m_i \ddot{\mathbf{x}}_i = \mathbf{f}_i = \sum_j \mathbf{f}_{ij}, \quad (10)$$

where \mathbf{f}_{ij} is the force exerted by particle j on particle i .



Contact time

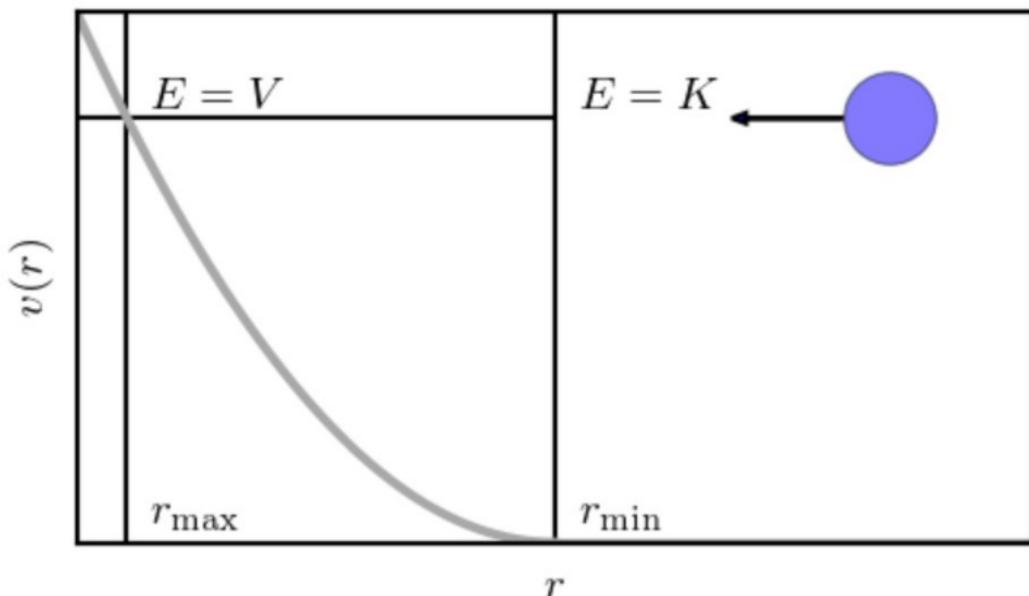


Figure 5: Derivation of the contact time.

Using the equations for energy

$$E = \frac{1}{2}m\dot{r}^2 + V(r) = \text{const.} \quad (11)$$

and radial velocity

$$\frac{dr}{dt} = \left[\frac{2}{m} (E - V(r)) \right]^{\frac{1}{2}}, \quad (12)$$

we derive the contact time

$$t_c = 2 \int_0^{\frac{1}{2}t_c} dt = 2 \int_{r_{\min}}^{r_{\max}} \frac{dt}{dr} dr = 2 \int_{r_{\min}}^{r_{\max}} \left[\frac{2}{m} (E - V(r)) \right]^{-\frac{1}{2}} dr, \quad (13)$$

where r_{\min} and r_{\max} are the range of the potential and the turning point of a colliding particle, respectively.

We expect reasonable results only if the time step is not larger than the smallest contact time. The time integration of the equations of motion is then possible using an integration method such as

- Euler's method,
- Runge-Kutta methods,
- Predictor-corrector methods,
- Verlet methods,
- Leap-frog methods.

Verlet method

We begin with a Taylor expansion of $x(t + \Delta t)$ for sufficiently small time steps Δt so that

$$\begin{aligned}\mathbf{x}(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{v}(t) + \frac{1}{2} \Delta t^2 \ddot{\mathbf{x}}(t) + \mathcal{O}(\Delta t^3), \\ \mathbf{x}(t - \Delta t) &= \mathbf{x}(t) - \Delta t \mathbf{v}(t) + \frac{1}{2} \Delta t^2 \ddot{\mathbf{x}}(t) - \mathcal{O}(\Delta t^3).\end{aligned}\quad (14)$$

Adding the latter two expressions yields

$$\mathbf{x}(t + \Delta t) = 2\mathbf{x}(t) - \mathbf{x}(t - \Delta t) + \Delta t^2 \ddot{\mathbf{x}}(t) + \mathcal{O}(\Delta t^4). \quad (15)$$

Newton's second law enables us to express $\ddot{\mathbf{x}}(t)$ as

$$\ddot{\mathbf{x}}_i(t) = \frac{1}{m_i} \sum_j \mathbf{f}_{ij}(t) \quad \text{with} \quad \mathbf{f}_{ij}(t) = -\nabla V(r_{ij}(t)). \quad (16)$$

The particle trajectories are then computed by plugging in the latter results in Eq. (15). Typically, we use a time step of approximately $\Delta t \approx t_c/20$, with t_c a contact time defined in Eq. (13).

Some general remarks about the Verlet method:

- Two time steps need to be stored (t and $t - \Delta t$).
- Velocities can be computed with $\mathbf{v}(t) = \frac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t-\Delta t)}{2\Delta t}$.
- The local numerical error is of order $\mathcal{O}(\Delta t^4)$, i.e. it is globally a third order algorithm.
- The numbers which are added are of order $\mathcal{O}(\Delta t^0)$ and $\mathcal{O}(\Delta t^2)$.
- Improvable by systematical inclusion of higher orders (very inefficient).
- The method is time reversible, which allows to estimate the error accumulation by reversing the process and comparing it to the initial conditions.

Leapfrog method

For the derivation of the Leapfrog method, we consider velocities at intermediate steps:

$$\mathbf{v} \left(t + \frac{1}{2} \Delta t \right) = \mathbf{v}(t) + \frac{1}{2} \Delta t \dot{\mathbf{v}}(t) + \mathcal{O}(\Delta t^2), \quad (17)$$

$$\mathbf{v} \left(t - \frac{1}{2} \Delta t \right) = \mathbf{v}(t) - \frac{1}{2} \Delta t \dot{\mathbf{v}}(t) + \mathcal{O}(\Delta t^2). \quad (18)$$

Taking the difference of the two equations leads to

$$\mathbf{v} \left(t + \frac{1}{2} \Delta t \right) = \mathbf{v} \left(t - \frac{1}{2} \Delta t \right) + \Delta t \ddot{\mathbf{x}}(t) + \mathcal{O}(\Delta t^3) \quad (19)$$

and we then update the positions according to

$$\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \Delta t \mathbf{v} \left(t + \frac{1}{2} \Delta t \right) + \mathcal{O}(\Delta t^4). \quad (20)$$

The analogies and differences between the Leapfrog method

$$\begin{aligned} \dot{\mathbf{v}}(t + \Delta t) &= \frac{f(\mathbf{x}(t))}{m}, \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \Delta t \dot{\mathbf{v}}(t + \Delta t), \\ \mathbf{x}(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{v}(t + \Delta t) \end{aligned} \quad (21)$$

and the forward Euler integration

$$\begin{aligned} \dot{\mathbf{v}}(t + \Delta t) &= \frac{f(\mathbf{x}(t))}{m}, \\ \mathbf{x}(t + \Delta t) &= \mathbf{x}(t) + \Delta t \mathbf{v}(t), \\ \mathbf{v}(t + \Delta t) &= \mathbf{v}(t) + \Delta t \dot{\mathbf{v}}(t + \Delta t) \end{aligned} \quad (22)$$

are the following: The update of the variables is done in a different order (both methods rely on explicit forward integration). In the case of the Leapfrog method, the position is not updated using the previous velocity, as it is done in the usual Euler method.

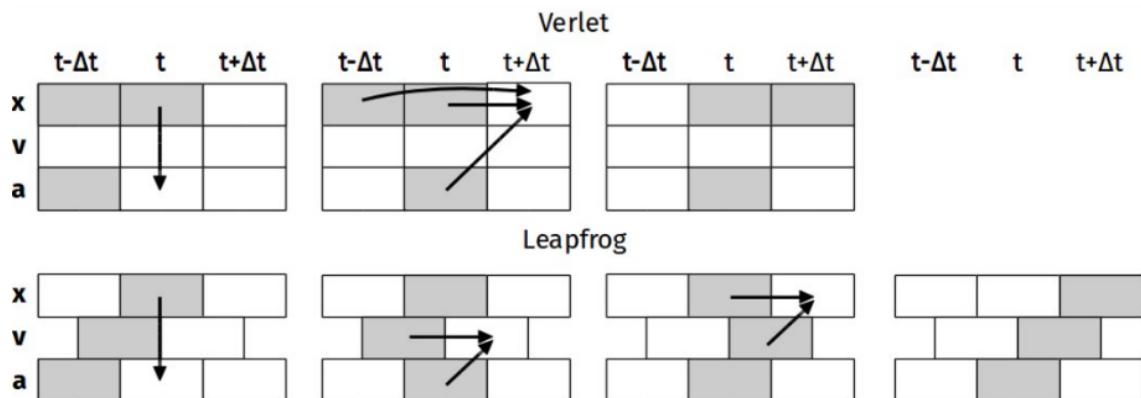


Figure 6: A comparison between Verlet and Leapfrog update schemes.

Velocity Verlet

$$\mathbf{a}^n = \frac{1}{m} \mathbf{F}(\mathbf{x}^n, \mathbf{v}^{n-1/2})$$
$$\mathbf{v}^{n+1/2} = \mathbf{v}^{n-1/2} + \mathbf{a}^n \Delta t$$
$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{v}^{n+1/2} \Delta t$$

```
#verlet step
"""
r_next = 2.0*r_current - r_previous + dt2_m*F

v_current = (r_next - r_previous)*dt2_inv

for i in range(N):
    if any((r_current[i,:] > L) & (r_next[i,:] > L)):
        #if any(r_current[i,:] > L):
        r_current[i,:] %= L
        r_next[i,:] %= L
"""

#velocity verlet
v_current += dt_m*F
r_next = r_current + v_current*dt

r_next %= L
```

VV doesn't need information about $r(t-dt)$! Simpler than the Verlet method.

Optimization

For MD simulations of N particles – we have an operation of complexity $\mathcal{O}(N^2)$.

Alternatively, let our potential be a function $v(r) \sim r^{-2n}$ with $n \geq 1$. We can then omit the computation of the square root in

$$r_{ij} = \sqrt{\sum_{\alpha=1}^d (x_i^\alpha - x_j^\alpha)^2} \quad (23)$$

since for the chosen potential $\mathbf{f} = -\nabla r^{-2n} \propto r^{-2(n-1)} \mathbf{r}$ and $\mathbf{f}_i = f(r^{-2(n-1)}) \mathbf{r}_i$.

If the potential is not a simple function and its calculation would imply a lot of tedious calculations, discretizing the potential and storing its values in a lookup table might be helpful. For short range potentials, we define a cutoff r_c and discretize the interval $(0, r_c^2)$ in K pieces, i.e.,

$$l_k = \frac{k}{K} r_c^2. \quad (24)$$

The force values stored in a lookup table are $f_k = f(\sqrt{l_k})$ and the corresponding index k is given by

$$k = \left\lfloor S \sum_{\alpha=1}^d (x_i^\alpha - x_j^\alpha)^2 \right\rfloor + 1, \quad (25)$$

where $\lfloor \cdot \rfloor$ denotes the floor function and $S = K/r_c^2$.

The definition of a cutoff makes it necessary that we introduce a cutoff potential $\tilde{v}(r)$ according to

$$\tilde{v}(r) = \begin{cases} v(r) - v(r_c) - \frac{\partial v}{\partial r} \Big|_{r=r_c} (r - r_c) & \text{if } r \leq r_c, \\ 0 & \text{if } r > r_c, \end{cases} \quad (26)$$

where $v(r_c)$ is the value of the original potential at r_c . Without adding the derivative term to the potential $\tilde{v}(r)$, there would be a discontinuity in the corresponding force.

In the case of the Lennard-Jones potential, a value of $r_c = 2.5\sigma$ is typically used. Care must be taken for potentials decaying as r^{-1} , since forces at large distances are not negligible.

Linked-Cell Method

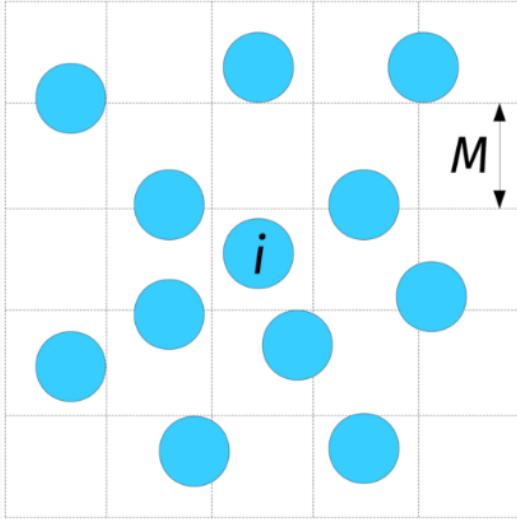


Figure 6: An illustration of the linked-cell method. A grid with grid spacing M ($\frac{r_c}{2} < M < r_c$) is placed on top of the MD simulation geometry. Only interactions between particles in a certain cell neighborhood have to be considered [D. Knuth, *The Art of Computer Programming*, Vol. 3, Sect. 4.2.2]. In d dimensions there are 3^d cells of interest. On average, we thus have to compute the interactions of $N3^dN/M^d$ particles. To keep track of the locations of all particles, we define a vector FIRST of length $N_M = M^d$ to store the index of a particle located in cell j in FIRST [j]. If cell j is empty, then FIRST [j] = 0. In a second vector LIST of length N , the indices of the remaining particles located in the same cell are stored. If the particle i is the last one in a cell, then LIST [i] = 0.

The following code shows an example of how to extract the particles located in cell $i = 2$.

```
i=2;
A[1]=FIRST[i];
while(M[i-1] !=0)
{
    A[j]=LIST[M[j-1]];
}
```

```
#celllist properties
l = int(L/rc)
l2 = l*l
n_cells = l2*l
cellsize = L/l

def update_celllist(r_current):
    celllist = [ [] for _ in range(n_cells) ] #empty list

    for idx in range(N):
        r = r_current[idx,:]
        i_x = int(r[0]/cellsize)
        i_y = int(r[1]/cellsize)
        i_z = int(r[2]/cellsize)

        celllist[i_x*l2 + i_y*l + i_z].append(idx)

    return celllist

def get_cell_coord(cell_idx):
    cell_idx_yz = cell_idx % l2
    x = int(cell_idx / l2)
    y = int(cell_idx_yz / l)
    z = cell_idx_yz % l

    return x, y, z

E_kin = 0.5*m.sum(np.square(v_current))
E_pot = 0.0

F = np.zeros((N,3))
for cell_idx in range(n_cells):
    main_cell = celllist[cell_idx] #main cell

    cell_idx_x, cell_idx_y, cell_idx_z = get_cell_coord(cell_idx)

    for idx_i in range(len(main_cell)):
        i = main_cell[idx_i]
        for idx_x in [(cell_idx_x-1+l)*l, cell_idx_x, (cell_idx_x+1)*l]:
            for idx_y in [(cell_idx_y-1+l)*l, cell_idx_y, (cell_idx_y+1)*l]:
                for idx_z in [(cell_idx_z-1+l)*l, cell_idx_z, (cell_idx_z+1)*l]:
                    other_cell = celllist[idx_x*l2 + idx_y*l + idx_z]
                    for idx_j in range(len(other_cell)):
                        j = other_cell[idx_j]
                        if i < j:
                            r_vec = r_rel_pbc(r_current[i,:], r_current[j,:])
                            r2 = r_vec.dot(r_vec)
                            if r2 < rc2:
                                force, potential = force_potential(r_vec, r2)
                                E_pot += potential - V_c
                                F[i,:] += force
                                F[j,:] -= force
```

When a particle changes the cell, FIRST and LIST are updated locally to avoid loops over all particles. The algorithm is thus of order $\mathcal{O}(N)$. In addition, this method is well suited for parallelization (domain composition).

Lagrange multipliers

One of the first description of composed particle systems based on an additional force term in the equations of motions has been suggested in [Ryckaert, Ciccotti and Berendsen, 1977]. The idea is to rewrite the equation of motion for each particle as

$$m_i \ddot{\mathbf{x}}_i = \underbrace{\mathbf{f}_i}_{\text{external interaction}} + \underbrace{\mathbf{g}_i}_{\text{internal constraints}}, \quad (28)$$

where the first term accounts for interactions between different composed particles and the second one describes the constraint forces.

We now impose such constraints to enforce the geometric arrangement of the molecules, e.g., certain distances d_{12} and d_{23} between atoms. Therefore, we define a potential such that the constraint forces \mathbf{g}_i are proportional to the difference of the actual and the desired distance of the particles. Considering a water molecule consisting of three particles, the two distance measures

$$\chi_{12} = r_{12}^2 - d_{12}^2, \quad (29)$$

$$\chi_{23} = r_{23}^2 - d_{23}^2, \quad (30)$$

are zero if the particles have the desired distance.

With $r_{ij} = \|\mathbf{r}_{ij}\|$ and $\mathbf{r}_{ij} = \mathbf{x}_i - \mathbf{x}_j$ we obtain

$$\mathbf{g}_k = \frac{\lambda_{12}}{2} \nabla_{\mathbf{x}_k} \chi_{12} + \frac{\lambda_{23}}{2} \nabla_{\mathbf{x}_k} \chi_{23}, \quad (31)$$

for $k \in \{1, 2, 3\}$.

The yet undetermined Lagrange multipliers are defined by λ_{12} and λ_{23} . We compute these multipliers by imposing the constraints.

According to Eq. (31), the constraint forces are

$$\mathbf{g}_1 = \lambda_{12} \mathbf{r}_{12}, \quad \mathbf{g}_2 = \lambda_{23} \mathbf{r}_{23} - \lambda_{12} \mathbf{r}_{12}, \quad \mathbf{g}_3 = -\lambda_{23} \mathbf{r}_{23}. \quad (32)$$

The previous equations describe nothing but a linear spring with a yet to be determined spring constant $\lambda_{(.)}$. To obtain the values of the Lagrange multipliers $\lambda_{(.)}$, the Verlet algorithm is executed in two steps. We first compute the Verlet update without constraint to obtain

$$\tilde{\mathbf{x}}_i(t + \Delta t) = 2\mathbf{x}_i - \mathbf{x}_i(t - \Delta t) + \Delta t^2 \frac{\mathbf{f}_i}{m_i}. \quad (33)$$

Then we correct the value using the constraints according to

$$\mathbf{x}_i(t + \Delta t) = \tilde{\mathbf{x}}_i(t + \Delta t) + \Delta t^2 \frac{\mathbf{g}_i}{m_i}. \quad (34)$$

By combining Eqs. (34) and (31), the updated positions are given by

$$\mathbf{x}_1(t + \Delta t) = \tilde{\mathbf{x}}_1(t + \Delta t) + \Delta t^2 \frac{\lambda_{12}}{m_1} \mathbf{r}_{12}(t), \quad (35)$$

$$\mathbf{x}_2(t + \Delta t) = \tilde{\mathbf{x}}_2(t + \Delta t) + \Delta t^2 \frac{\lambda_{23}}{m_2} \mathbf{r}_{23}(t) - \Delta t^2 \frac{\lambda_{12}}{m_2} \mathbf{r}_{12}(t), \quad (36)$$

$$\mathbf{x}_3(t + \Delta t) = \tilde{\mathbf{x}}_3(t + \Delta t) - \Delta t^2 \frac{\lambda_{23}}{m_3} \mathbf{r}_{23}(t). \quad (37)$$

With these expressions, we now obtain λ_{12} and λ_{23} by inserting (35), (36) and (37) into the constraint condition, i.e.,

$$\begin{aligned} |\mathbf{x}_1(t + \Delta t) - \mathbf{x}_2(t + \Delta t)|^2 &= d_{12}^2, \\ |\mathbf{x}_2(t + \Delta t) - \mathbf{x}_3(t + \Delta t)|^2 &= d_{23}^2, \end{aligned} \quad (38)$$

and finally

$$\begin{aligned} \left| \tilde{\mathbf{r}}_{12}(t + \Delta t) + \Delta t^2 \lambda_{12} \left(\frac{1}{m_1} + \frac{1}{m_2} \right) \mathbf{r}_{12}(t) - \Delta t^2 \frac{\lambda_{23}}{m_2} \mathbf{r}_{23}(t) \right|^2 &= d_{12}^2, \\ \left| \tilde{\mathbf{r}}_{23}(t + \Delta t) + \Delta t^2 \lambda_{23} \left(\frac{1}{m_2} + \frac{1}{m_3} \right) \mathbf{r}_{23}(t) - \Delta t^2 \frac{\lambda_{12}}{m_2} \mathbf{r}_{12}(t) \right|^2 &= d_{23}^2, \end{aligned} \quad (39)$$

where $\tilde{\mathbf{r}}_{ij} = \tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_j$. These coupled quadratic equations are solved in practice perturbatively by linearising in t .

Rigid Bodies 2D

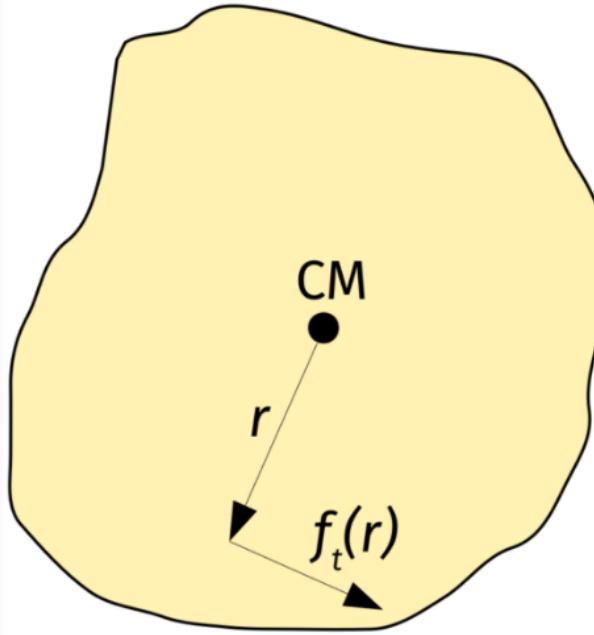


Figure 2: An example of a rigid body in two dimensions. The black dot show the center of mass (CM), and $f_t(r)$ represents the tangential force component.

In two dimensions, the moment of inertia and the torque are given by

$$I = \int \int_A r^2 \rho(r) dA \quad \text{and} \quad M = \int \int_A r f_t(r) dA, \quad (3)$$

where f_t is the tangential force. In general, the mass density may be constant or depending on the actual position and not only on the radius r . The equation of motion is given by

$$I \dot{\omega} = M. \quad (4)$$

We now apply the Verlet algorithm to \mathbf{x} and the rotation angle ϕ to compute the corresponding time evolutions according to

$$\begin{aligned} \phi(t + \Delta t) &= 2\phi(t) - \phi(t - \Delta t) + \Delta t^2 \frac{M(t)}{I}, \\ \mathbf{x}(t + \Delta t) &= 2\mathbf{x}(t) - \mathbf{x}(t - \Delta t) + \Delta t^2 M^{-1}(t) \sum_{j \in A} f_j(t), \end{aligned} \quad (5)$$

where the total torque is the sum over all the torques acting on the rigid body, i.e.,

$$M(t) = \sum_{j \in A} \left[f_j^y(t) d_j^x(t) - f_j^x(t) d_j^y(t) \right]. \quad (6)$$

Rigid Bodies 3D

To describe the motion of rigid bodies in three dimensions, we consider a lab-fixed and a body-fixed coordinate system x and y , respectively. The transformation between both systems is given by

$$\mathbf{x} = R(t)\mathbf{y}, \quad (7)$$

where $R(t) \in \text{SO}(3)$ denotes a rotation matrix¹.

¹The group $\text{SO}(3)$ is the so-called three dimensional rotation group, or special orthogonal group. All rotation matrices $R \in \text{SO}(3)$ fulfill $R^T R = R R^T = 1$.

Furthermore, we define $\Omega = R^T R$ and find with $R^T R = 1$ that

$$R^T \dot{R} + \dot{R}^T R = \Omega + \Omega^T = 0. \quad (8)$$

The latter equation implies that Ω is skew-symmetric and thus of the form

$$\Omega = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \quad \text{and} \quad \Omega \mathbf{y} = \boldsymbol{\omega} \wedge \mathbf{y}, \quad (9)$$

where $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$.

The angular momentum is then given by

$$\mathbf{L} = \sum_{i=1}^n m_i \mathbf{x}_i \wedge \dot{\mathbf{x}}_i = \sum_{i=1}^n m_i R \mathbf{y}_i \wedge \dot{R} \mathbf{y}_i. \quad (10)$$

Combining Eqs. (10) and (9) yields

$$\mathbf{L} = R \sum_{i=1}^n m_i \mathbf{y}_i \wedge (\boldsymbol{\omega} \wedge \mathbf{y}_i) = R \sum_{i=1}^n m_i [\boldsymbol{\omega} (\mathbf{y}_i \cdot \mathbf{y}_i) - \mathbf{y}_i (\boldsymbol{\omega} \cdot \mathbf{y}_i)]. \quad (11)$$

The components of the inertia tensor are defined as

$$I_{jk} = \sum_{i=1}^n m_i [(\mathbf{y}_i \cdot \mathbf{y}_i) \delta_{jk} - \mathbf{y}_i^j \mathbf{y}_i^k] \quad (12)$$

and thus

$$\mathbf{L} = R \mathbf{S} \quad \text{with} \quad S_j = \sum_{k=1}^3 I_{jk} \boldsymbol{\omega}_k. \quad (13)$$

where I is the inertia tensor.

Considering a coordinate system whose axes are parallel to the principal axes of inertia of the body, the inertia tensor takes the form

$$I = \begin{pmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{pmatrix} \quad \text{and} \quad S_j = I_j \omega_j. \quad (14)$$

With Eq. (13), the equations of motion are determined by

$$\dot{\mathbf{L}} = \dot{R} \mathbf{S} + R \dot{\mathbf{S}} = \widetilde{\mathbf{M}}, \quad (15)$$

where $\widetilde{\mathbf{M}} = R \mathbf{M}$ represents the torque in the lab-fixed coordinate system. By multiplying the latter equation with R^T , we find the *Euler equations* in the principal axes coordinate system, i.e.,

$$\dot{\omega}_1 = \frac{M_1}{I_1} + \left(\frac{I_2 - I_3}{I_1} \right) \omega_2 \omega_3, \quad (16)$$

$$\dot{\omega}_2 = \frac{M_2}{I_2} + \left(\frac{I_3 - I_1}{I_2} \right) \omega_3 \omega_1, \quad (17)$$

$$\dot{\omega}_3 = \frac{M_3}{I_3} + \left(\frac{I_1 - I_2}{I_3} \right) \omega_1 \omega_2. \quad (18)$$

The angular velocities are then integrated according to

$$\omega_1(t + \Delta t) = \omega_1(t) + \Delta t \frac{M_1(t)}{I_1} + \Delta t \left(\frac{I_2 - I_3}{I_1} \right) \omega_2 \omega_3, \quad (19)$$

$$\omega_2(t + \Delta t) = \omega_2(t) + \Delta t \frac{M_2(t)}{I_2} + \Delta t \left(\frac{I_3 - I_1}{I_2} \right) \omega_3 \omega_1, \quad (20)$$

$$\omega_3(t + \Delta t) = \omega_3(t) + \Delta t \frac{M_3(t)}{I_3} + \Delta t \left(\frac{I_1 - I_2}{I_3} \right) \omega_1 \omega_2. \quad (21)$$

From these expressions, we obtain the angular velocity in the laboratory frame

$$\tilde{\boldsymbol{\omega}}(t + \Delta t) = R \boldsymbol{\omega}(t + \Delta t). \quad (22)$$

Since the particles are moving all the time, the rotation matrix is not constant. We therefore have to find an efficient way to determine and update R at every step in our simulation. In the following, we therefore discuss Euler angles and quaternions.

Euler angles

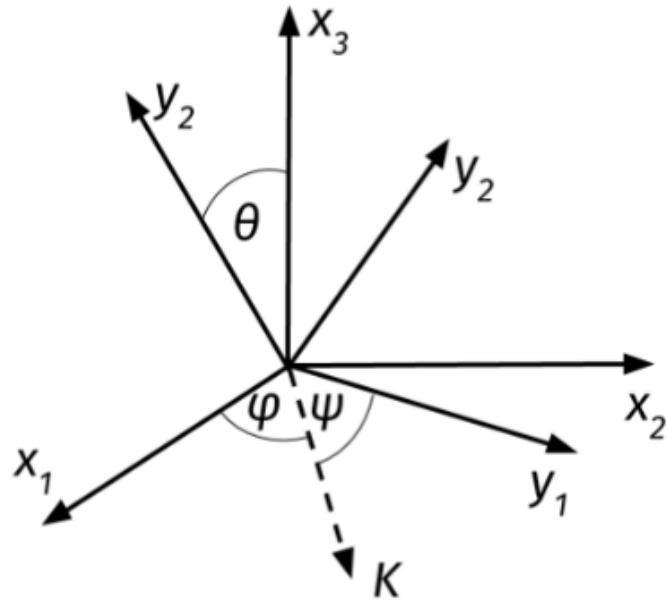


Figure 4: Euler angle parameterization of a rotation matrix.

One possible parameterization of the rotation matrix R is the following:

$$R = R(\phi, \theta, \psi)$$

$$= \begin{pmatrix} \cos \phi & -\sin \phi & 0 \\ \sin \phi & \cos \phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (23)$$

As a consequence of the occurrence of products of multiple trigonometric functions for arbitrary rotations, this parameterization is not well-suited for efficient computations. We have to keep in mind that this operation has to be performed for every particle and every time step, making this approach computationally too expensive. For the computation of angular velocities, derivatives of Eq. (23) have to be considered.

Quaternions

Quaternions are a generalization of complex numbers, where four basis vectors span a four-dimensional space. By defining

$$q_0 = \cos\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi + \psi}{2}\right), \quad (24)$$

$$q_1 = \sin\left(\frac{\theta}{2}\right) \cos\left(\frac{\phi - \psi}{2}\right), \quad (25)$$

$$q_2 = \sin\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi - \psi}{2}\right), \quad (26)$$

$$q_3 = \cos\left(\frac{\theta}{2}\right) \sin\left(\frac{\phi + \psi}{2}\right), \quad (27)$$

with $0 < q_i < 1$ and $\sum_i q_i = 1$ for $i \in \{1, \dots, 4\}$, we represent the angles in dependence of a set of quaternions q_i . The Euclidean norm of q equals unity and thus there exist only three independent parameters.

The rotation matrix as defined in Eq. (23) has a quaternion representation, i.e.,

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 + q_0 q_1) \\ 2(q_1 q_3 + q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (28)$$

We now found a more efficient way of computing rotations without the necessity of computing lengthy products of sine and cosine functions. This approach much faster than one of Eq. (23). The angular velocities are then computed according to

$$\begin{pmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \begin{pmatrix} 0 \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix} \quad (29)$$

Since the world of quaternions and the normal Euclidean space are connected by a diffeomorphism, there is always the possibility of calculating the values of the Euler angles if needed

$$\phi = \arctan \left[\frac{2(q_0 q_1 + q_2 q_3)}{1 - 2(q_1^2 + q_2^2)} \right] \quad (30)$$

$$\theta = \arcsin [2(q_0 q_2 - q_1 q_3)] \quad (31)$$

$$\psi = \arctan \left[\frac{2(q_0 q_3 + q_1 q_2)}{1 - 2(q_2^2 + q_3^2)} \right] \quad (32)$$

There is no need of calculating the Euler angles at each integration step. We now simulate our rigid body dynamics in quaternion representation according to the following strategy:

- Compute the torque $M(t)$ in the body frame.
- Obtain $\omega(t + \Delta t)$ according to Eq. (21) (quaternion representation).
- Update the rotation matrix as defined in Eq. (28) by computing $q(t + \Delta t)$ according to Eq. (29).

Canonical Ensemble

Experiments are often conducted at constant temperature and not at constant energy. This is a common situation, since systems are usually able to exchange energy with their environment. We therefore first couple our system to a heat bath to realize this situation. There are various options to do this

- Rescaling of velocities,
- Introducing constraints (Hoover),
- Nosé-Hoover thermostat,
- Stochastic method (Anderson).

However, before focusing on the discussion of the latter methods, we shall define the concept of temperature used in the subsequent sections. We start from the equipartition theorem

$$\left\langle q_\mu \frac{\partial \mathcal{H}}{\partial q_\nu} \right\rangle = \left\langle p_\mu \frac{\partial \mathcal{H}}{\partial p_\nu} \right\rangle = \delta_{\mu\nu} kT \quad (1)$$

for a Hamiltonian \mathcal{H} with the generalized coordinates \mathbf{q} and \mathbf{p} . We consider a classical system whose Hamiltonian is given by

$$\mathcal{H} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i} + V(\mathbf{x}_1, \dots, \mathbf{x}_N) \quad (2)$$

and we define the instantaneous temperature

$$\mathcal{T} = \frac{2}{3k(N-1)} \sum_{i=1}^N \frac{\mathbf{p}_i^2}{2m_i}. \quad (3)$$

Velocity rescaling

Intuitively, we should be able to adjust the system's instantaneous temperature by rescaling the velocities of the particles according to

$$\mathbf{v}_i \rightarrow \alpha \mathbf{v}_i. \quad (4)$$

The measured temperature is proportional to the squared velocities and thus

$$\mathcal{T} \rightarrow \alpha^2 \mathcal{T}. \quad (5)$$

Therefore, we have to set

$$\alpha = \sqrt{\frac{T}{\mathcal{T}}} \quad (6)$$

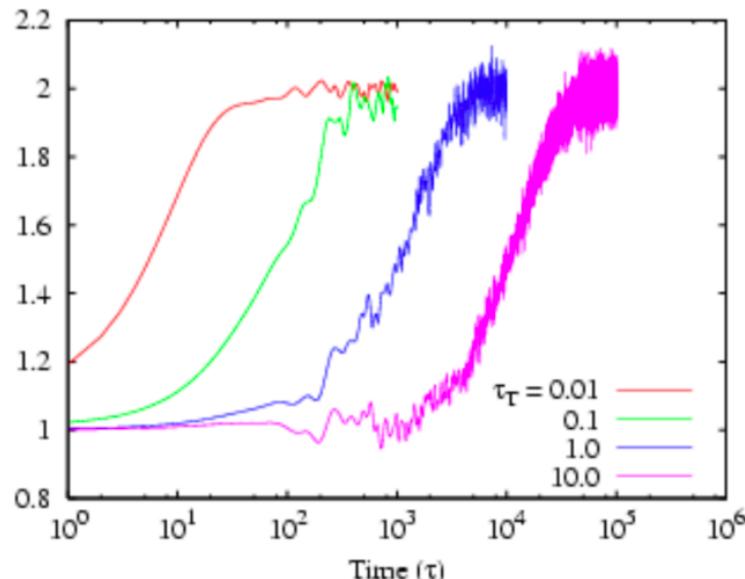
to stay at a fixed desired temperature T .

This method is very easy to implement. However, the problem is that we change the physics and in particular the time and the resulting velocity distribution deviates from the canonical one.

This method might seem to be very effective. A modification of this method makes use of an additional parameter t_T (relaxation time) which describes the coupling to heat bath. The scaling factor is then (Berendsen thermostat)

$$\alpha = \sqrt{1 + \frac{\Delta t}{t_T} \left(\frac{T}{\mathcal{T}} - 1 \right)}. \quad (7)$$

Still we do not recover the canonical velocity distribution (Maxwell–Boltzmann). Velocity rescaling should be only applied to initialize a configuration at given temperature.



Nosé-Hoover thermostat

In order to overcome the problem of the wrong velocity distribution, we are now going to discuss the Nosé-Hoover thermostat as the correct method to simulate heat bath particle dynamics. Shuichi Nosé introduced a new degree of freedom s that describes the heat bath. The corresponding potential and kinetic energy are

$$\begin{aligned}\mathcal{V}(s) &= (3N + 1) k_B T \ln s, \\ K(s) &= \frac{1}{2} Q s^2.\end{aligned}\quad (13)$$

The new degree of freedom s rescales the time step dt and momenta \mathbf{p}_i according to

$$dt' = s dt \quad \text{and} \quad \mathbf{p}'_i = s \mathbf{p}_i. \quad (14)$$

Similarly, velocities are also rescaled since

$$\mathbf{v}'_i = \frac{d\mathbf{x}_i}{dt'} = \frac{d\mathbf{x}_i}{dt} \frac{dt}{dt'} = \frac{\mathbf{v}_i}{s}. \quad (15)$$

Note that we also used the chain rule in the second step of the equation for the rescaling of momenta:

$$\mathbf{p}'_i = \nabla_{\mathbf{v}'_i} \left(\frac{1}{2} \sum_{i=1}^N m_i \mathbf{v}'_i^2 \right) = s \nabla_{\mathbf{v}_i} \left(\frac{1}{2} \sum_{i=1}^N m_i \mathbf{v}_i^2 \right) = s \mathbf{p}_i, \quad (16)$$

The Hamiltonian is thus

$$\mathcal{H} = \sum_{i=1}^N \frac{\mathbf{p}'_i^2}{2m_i s^2} + \frac{1}{2} Q s^2 + V(\mathbf{x}_1, \dots, \mathbf{x}_N) + \mathcal{V}(s), \quad (17)$$

with $p_s = Qs$ being the momentum corresponding to s . The velocities are

$$\begin{aligned}\frac{d\mathbf{x}_i}{dt'} &= \nabla_{\mathbf{p}'_i} \mathcal{H} = \frac{\mathbf{p}'_i}{m_i s^2}, \\ \frac{ds}{dt'} &= \frac{\partial \mathcal{H}}{\partial p_s} = \frac{p_s}{Q}.\end{aligned}\quad (18)$$

With $\mathbf{p}'_i = m_i s^2 \dot{\mathbf{x}}_i$ we find

$$\mathbf{f}_i = \frac{d\mathbf{p}'_i}{dt'} = -\frac{\partial \mathcal{H}}{\partial \mathbf{x}_i} = -\nabla_{\mathbf{x}_i} V(\mathbf{x}_1, \dots, \mathbf{x}_N) = 2m_i s \dot{s} \dot{\mathbf{x}}_i + m_i s^2 \ddot{\mathbf{x}}_i \quad (19)$$

and

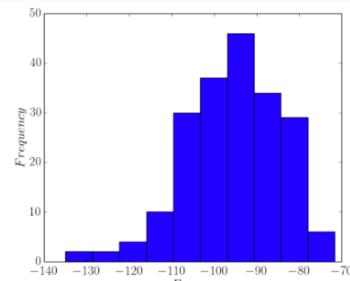
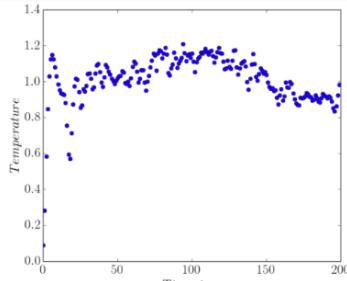
$$\frac{dp_s}{dt'} = -\frac{\partial \mathcal{H}}{\partial s} = \frac{1}{s} \left[\sum_{i=1}^N \frac{\mathbf{p}'_i^2}{m_i s^2} - (3N + 1) k_B T \right]. \quad (20)$$

Based on the latter Hamilton equations, we find for the equations of motion in virtual time t'

$$m_i s^2 \ddot{\mathbf{x}}_i = \mathbf{f}_i - 2m_i s \dot{s} \dot{\mathbf{x}}_i \quad \text{with} \quad i \in \{1, \dots, N\} \quad (21)$$

and

$$Q \ddot{s} = \sum_{i=1}^N m_i s \dot{\mathbf{x}}_i^2 - \frac{1}{s} (3N + 1) k_B T. \quad (22)$$



In order to obtain the equations of motion in real time, we have to remind ourselves that $dt = dt'/s$ and $\mathbf{p}'_i = s \mathbf{p}_i$. Thus, we find for the velocities

$$\begin{aligned}\frac{d\mathbf{x}_i}{dt} &= s \frac{d\mathbf{x}_i}{dt'} = \frac{\mathbf{p}'_i}{m_i s} = \frac{\mathbf{p}_i}{m_i}, \\ \frac{ds}{dt} &= s \frac{ds}{dt'} = s \frac{\mathbf{p}_s}{Q},\end{aligned}\quad (23)$$

and for the forces

$$\begin{aligned}\frac{d\mathbf{p}_i}{dt} &= s \frac{d}{dt'} \left(\frac{\mathbf{p}'_i}{s} \right) = \frac{d\mathbf{p}'_i}{dt'} - \frac{1}{s} \frac{ds}{dt'} \mathbf{p}'_i = \mathbf{f}_i - \frac{1}{s} \frac{ds}{dt} \mathbf{p}_i, \\ \frac{d\mathbf{p}_s}{dt} &= s \frac{dp_s}{dt'} = \sum_{i=1}^N \frac{\mathbf{p}_i^2}{m_i} - (3N + 1) kT.\end{aligned}\quad (24)$$

With $\xi = \frac{d \ln(s)}{dt} = \frac{\dot{s}}{s}$ representing a friction term, the equations of motions (65) and (66) are given in real time by

$$\ddot{\mathbf{x}}_i = \frac{\mathbf{f}_i}{m_i} - \xi \dot{\mathbf{x}}_i \quad (25)$$

and

$$Q \dot{\xi} = \sum_{i=1}^N m_i \dot{\mathbf{x}}_i^2 - (3N + 1) k_B T. \quad (26)$$

The first term in Eq. (70) denotes the **measured** kinetic energy whereas the second one corresponds to the **desired** kinetic energy. The quantity Q represents the coupling to the heat bath and the higher the value of Q , the stronger the system reacts to temperature fluctuations. For $Q \rightarrow \infty$, we recover microcanonical MD.

A reasonable value of Q is characterized by the fact that normal temperature fluctuations are observed, i.e.,

$$\overline{\Delta T} = \sqrt{\frac{2}{Nd} \overline{T}}, \quad (27)$$

where d is the system's dimension and N the number of particles. We now show that the Nosé–Hoover thermostat recovers the canonical partition function. Therefore, we start from microcanonical MD and the corresponding partition function

$$Z = \int \delta(\mathcal{H} - E) ds dp_s d^3x' d^3p', \quad (28)$$

where the x and p integration has to be taken over a three dimensional space with N particles. With $\mathcal{H} = \mathcal{H}_1 + (3N + 1) kT \ln(s)$ and in real time, we find

$$\begin{aligned}Z &= \int \delta[(\mathcal{H}_1 - E) + (3N + 1) kT \ln(s)] s^{3N} ds dp_s d^3x' d^3p' \\ &= \int \delta[s - e^{-\frac{\mathcal{H}_1 - E}{(3N + 1) kT}}] \frac{s^{3N+1}}{(3N + 1) kT} ds dp_s d^3x' d^3p',\end{aligned}\quad (29)$$

where we used the identity $\delta[f(s)] = \delta(s - s_0)/f'(s)$ with $f(s_0) = 0$ in the second step.

Integrating Eq. (73) over s yields

$$\begin{aligned}Z &= \int \frac{1}{(3N + 1) kT} e^{-\frac{\mathcal{H}_1 - E}{kT}} dp_s d^3x d^3p \\ &= \int e^{-\frac{\mathcal{H}_1 - E}{kT}} d^3x d^3p \int \frac{1}{(3N + 1) kT} dp_s,\end{aligned}\quad (30)$$

with $\mathcal{H}_1 = \mathcal{H}_0 + \frac{p_s^2}{2Q}$. The first term of the last equation is the canonical partition function and the last term a constant prefactor.

Constant pressure

Another important situation is the one of constant pressure.

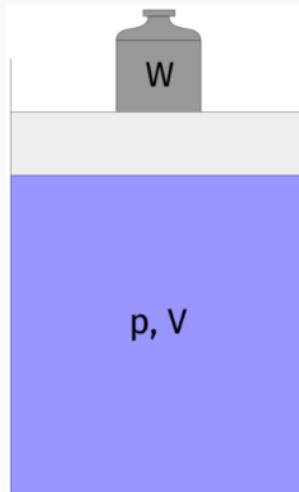


Figure 1: A weight of mass W exerts a pressure p on the system with volume V .

We will again consider the equipartition theorem (45) with the Hamiltonian

$$\mathcal{H} = K(\mathbf{p}) + V(\mathbf{x})$$

Taking the derivative of the \mathcal{H} with respect to the spatial component yields

$$\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot [\nabla_{\mathbf{x}_i} V(\mathbf{x})] \right\rangle = NkT.$$

We define

$$w = -\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot (\mathbf{f}_i^{\text{part}}) \right\rangle \quad (34)$$

as the viral. Based on

$$\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot (\mathbf{f}_i^{\text{part}}) \right\rangle = -\frac{1}{3} \int_{\Gamma} p \mathbf{x} d\mathbf{A} = -\frac{1}{3} p \int_V (\nabla \cdot \mathbf{x}) d\mathbf{V} = -pV \quad (35)$$

we define the instantaneous pressure \mathcal{P} by

$$PV \equiv Nk_B T + \langle w \rangle \quad (36)$$

Similarly to Nosé-Hoover thermostat, we introduce a sort of "pressure bath", i.e a parameter W which adjusts the pressure of the system. The volume change can be written as:

$$V = 1 - \alpha_T \frac{\Delta}{t_p} (p - \mathcal{P}) \quad (37)$$

where α_T is the isothermal compressibility and t_p is a relaxation time for the pressure.

Elastic Collisions

One of the first examples for event-driven programming applied to molecular dynamics is a work by Alder in 1957.

In this method only the exchange of the particles' momenta is taken into account and no forces are calculated. Furthermore, only binary collisions are considered and interactions between three or more particles are neglected. Between two collision events, the particles follow ballistic trajectories. To perform an event-driven MD simulation, we need to determine the time t_c between two collisions to then obtain the velocities of the two particles after the collision from the velocities of the particles before the collision using a look-up table.

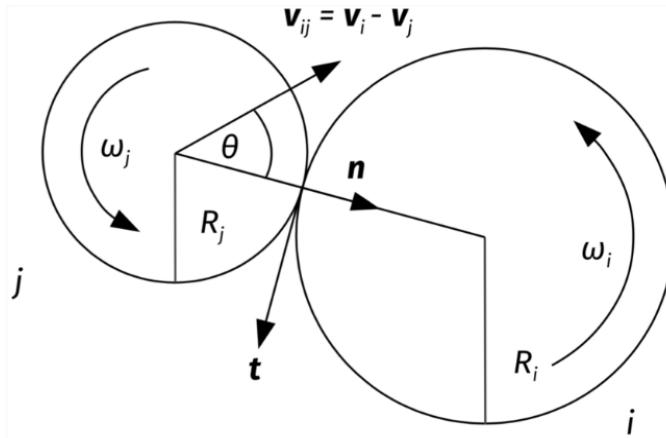


Figure 2: Two particles collide elastically.

For the moment, we are not taking into account the influence of friction and thus neglect the exchange of angular momentum. We compute the times t_{ij} , at which the next collision between the particle i and the particle j would occur. At time t_{ij} , the distance between the two particles is

$$|\mathbf{r}_{ij}(t_{ij})| = |R_i + R_j| \quad (38)$$

Given a relative velocity v_{ij} at time t_0 , the contact time t_{ij} of two particles can be obtained from

$$v_{ij}^2 t_{ij}^2 + 2 [\mathbf{r}_{ij}(t_0) \mathbf{v}_{ij}] t_{ij} + [r_{ij}(t_0)]^2 - (R_i + R_j)^2 = 0. \quad (39)$$

We should bear in mind that Eq. (83) are only meaningful if the trajectories of particles i and j cross with each other. The time t_c when the next collision occurs, is the minimum over all pairs, i.e.,

$$t_c = \min_{ij} (t_{ij}). \quad (40)$$

Thus, in the time interval $[t_0, t_c]$ the particles' positions and angular orientations evolve according to

$$\mathbf{r}_i(t_0 + t_c) = \mathbf{r}_i(t_0) + \mathbf{v}_i(t_0) t_c \quad \text{and} \quad \phi_i(t_0 + t_c) = \phi_i(t_0) + \omega_i(t_0) t_c. \quad (41)$$

Lubachevsky method

Instead of going through all the particle pairs ($\mathcal{O}(N^2)$), we create a list of events for each particle. The reordering of the event list takes a time in the order of $\mathcal{O}(N \log N)$.

In practice, this can be implemented in six arrays (event times, new partners, positions and velocities) of dimension N (number of particles in the system). Alternatively, one creates a list of pointers pointing to a data structure for each particle consisting of six variables.

Storing the last event is needed as particles are only updated after being involved in an event. For each particle i , the time $t^{(i)}$ is the minimal time of all possible collisions involving this particle, i.e.,

$$t^{(i)} = \min_j (t_{ij}). \quad (42)$$

Comparing particle i with $N - 1$ others can be improved by dividing the systems in sectors such that only neighboring sectors have to be considered in this step.

These sector boundaries have to be treated similar to obstacles such that when particles cross sector boundaries a collision event happens. For each particle i , this step would then be of order $\mathcal{O}(1)$ instead of $\mathcal{O}(N)$. The next collision occurs at time

$$t_c = \min_i (t^{(i)}). \quad (43)$$

We store $t^{(i)}$ in increasing order in a stack:

- The vector $\text{part}[m]$ points to particle i which is at position m in the stack. (Sometimes also a vector $\text{pos}[i]$ is used to store position m of particle i in the stack.)
- This constitutes an implicit ordering of the collision times $t^{(i)}$, where $m = 1$ points to the smallest time.
- $\text{part}[1]$ is the particle with minimal collision time:
$$t_c = t^{(\text{part}[1])}$$
- After the event for both particles all 6 entries (event times, new partners, positions and velocities) have to be updated. Additionally, the vector $\text{part}[m]$ has to be reordered.

Reordering the times $t^{(i)}$ after each event is of order $\mathcal{O}(\log N)$ when using, e.g., binary trees for sorting. The advantages of this method are that it is not necessary to minimize all the collision times of all the pairs at every step, and that it is unnecessary to update the positions of particles that do not collide. Only the position and velocity of the particle involved in the collision event are updated.

Collision with rotation

We now consider two spheres i and j of the same radius R and mass m . Due to friction, angular momentum is exchanged if particles collide with nonzero tangential velocity. The equations of motion for rotation are

$$I \frac{d\omega_i}{dt} = \mathbf{r} \wedge \mathbf{f}_i, \quad (50)$$

where I denotes the moment of inertia and \mathbf{f}_i the forces exerted on particle i .

In the case of two colliding disks of radius R , moment of inertia I and mass m , the exchange of angular momentum is

$$\begin{aligned} I(\omega'_i - \omega_i) &= -Rm\mathbf{n} \wedge (\mathbf{v}'_i - \mathbf{v}_i), \\ I(\omega'_j - \omega_j) &= Rm\mathbf{n} \wedge (\mathbf{v}'_j - \mathbf{v}_j), \end{aligned} \quad (51)$$

with the primed velocities representing the ones after the collision.

Together with the conservation of momentum

$$\mathbf{v}'_i + \mathbf{v}'_j = \mathbf{v}_i + \mathbf{v}_j, \quad (52)$$

we obtain the rule for computing the new angular velocities after the collision, i.e.,

$$\omega'_i - \omega_i = \omega'_j - \omega_j = -\frac{Rm}{I} (\mathbf{v}'_i - \mathbf{v}_i) \wedge \mathbf{n}. \quad (53)$$

The relative velocity between particles i and j is

$$\mathbf{u}_{ij} = \mathbf{v}_i - \mathbf{v}_j - R(\omega_i + \omega_j) \wedge \mathbf{n}. \quad (54)$$

We decompose the relative velocities \mathbf{u} of the particles into their normal and tangential components \mathbf{u}^n and \mathbf{u}^t , respectively.

It is important to keep in mind that we are at this point not interested in the relative velocities of the centers of mass of the particles. For the angular momentum exchange, the relevant quantity to consider is the relative velocity of the particle surfaces at the contact point. The normal and tangential velocities are given by

$$\begin{aligned} \mathbf{u}_{ij}^n &= (\mathbf{u}_{ij} \mathbf{n}) \mathbf{n}, \\ \mathbf{u}_{ij}^t &= \mathbf{u}_{ij} \wedge \mathbf{n} = [(\mathbf{v}_i - \mathbf{v}_j) - R(\omega_i + \omega_j)] \wedge \mathbf{n}. \end{aligned} \quad (55)$$

General slips are described by

$$\mathbf{u}_{ij}^{t'} = e_t \mathbf{u}_{ij}^t, \quad (56)$$

where the the *tangential restitution coefficient* e_t accounts for different slip types. The perfect slip collision is recovered for $e_t = 1$ which implies that no rotation energy is transferred from one particle to the other. No slip at all corresponds to $e_t = 0$. Energy conservation only holds if $e_t = 1$. In the case of $e_t < 1$, energy is dissipated.

If we compute the difference of the relative tangential velocities before and after the slip we get

$$\begin{aligned} (1 - e_t) \mathbf{u}_{ij}^t &= \mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} \\ &= -[(\mathbf{v}'_i - \mathbf{v}_i - \mathbf{v}'_j + \mathbf{v}_j) - R(\omega'_i - \omega_i + \omega'_j - \omega_j) \wedge \mathbf{n}]. \end{aligned}$$

Combining the previous equation with Eq. (97), we obtain an expression without angular velocities

$$\begin{aligned} \mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} &= (1 - e_t) \mathbf{u}_{ij}^t \\ &= -[2(\mathbf{v}_i^{t'} - \mathbf{v}_i^t) + 2q(\mathbf{v}_i^{t'} - \mathbf{v}_i^t)] \end{aligned} \quad (59)$$

and finally

$$\mathbf{v}_i^{t'} = \mathbf{v}_i^t - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)} \quad \text{with } q = \frac{mR^2}{I}. \quad (60)$$

Analogously, we find for the remaining quantities

$$\begin{aligned} \mathbf{v}_j^{t'} &= \mathbf{v}_j^t + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \omega'_i &= \omega_i - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}, \\ \omega'_j &= \omega_j - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}. \end{aligned} \quad (61)$$

And the updated velocities are

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \mathbf{u}_{ij}^n - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \mathbf{v}_j' &= \mathbf{v}_j + \mathbf{u}_{ij}^n + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}. \end{aligned} \quad (62)$$

Inelastic collisions (of rotating particles)

Elastic collisions correspond to $r = 1$ whereas perfect plasticity is described by $r = 0$. Similar to our previous discussion of collisions with rotations, we also distinguish between normal and tangential energy transfer and define the corresponding coefficients

$$e_n = \sqrt{r_n} = \frac{v_n^{\text{after}}}{v_n^{\text{before}}}, \quad (27)$$

$$e_t = \sqrt{r_t} = \frac{v_t^{\text{after}}}{v_t^{\text{before}}}. \quad (28)$$

In the case of a bouncing ball, the restitution coefficient accounts for effects such as air friction, deformations and thermal dissipation.

These coefficients strongly depend on the material, the shape of the particles, the energies involved in the events, the angle of impact and other factors. Usually, they are determined experimentally.

The relative velocity of the particles at their contact point is

$$\mathbf{u}_{ij}^n = (\mathbf{u}_{ij} \mathbf{n}) \mathbf{n} = [(\mathbf{v}_i - \mathbf{v}_j) \mathbf{n}] \mathbf{n}. \quad (29)$$

The normal velocity components are affected by inelasticity. In the case of an inelastic collision, dissipation effects lead to reduced normal velocities

$$\mathbf{u}_{ij}^{n'} = -e_n \mathbf{u}_{ij}^n \quad (30)$$

For $e_n = 1$, there is no dissipation whereas dissipation effects occur for $e_n < 1$.

Similar to Eq. (19) and following derivations, we obtain the expressions for the velocities of each particle after the collision

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n, \\ \mathbf{v}_j' &= \mathbf{v}_j + \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n. \end{aligned} \quad (31)$$

In the case of perfect slip, the momentum exchange is

$$\Delta \mathbf{p}_n = -m_{\text{eff}}(1 + e_n) [(\mathbf{v}_i - \mathbf{v}_j) \mathbf{n}] \mathbf{n}. \quad (32)$$

With $q = \frac{m_{\text{eff}} R^2}{I_{\text{eff}}}$, the equations for the velocities after the collision are

$$\begin{aligned} \mathbf{v}_i' &= \mathbf{v}_i - \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n - \frac{(1 - e_t)}{2(1 + q)} \mathbf{u}_{ij}^t, \\ \mathbf{v}_j' &= \mathbf{v}_j + \frac{(1 + e_n)}{2} \mathbf{u}_{ij}^n + \frac{(1 - e_t)}{2(1 + q)} \mathbf{u}_{ij}^t, \\ \omega_i' &= \omega_i - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}, \\ \omega_j' &= \omega_j + \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}. \end{aligned} \quad (33)$$

These equations describe inelastic collisions of rotating particles.

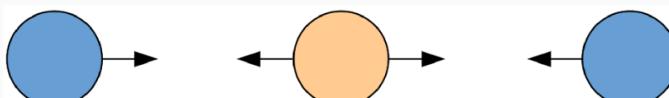


Figure 2: The orange particle at the center is bouncing between the blue particles which approach each other.



Every time the ball hits the surface its kinetic energy is lowered according to Eq.(26). As a consequence, the ball will not reach the initial height anymore and the time between two contacts with the surface approaches zero. After a finite time, the ball comes to a rest, but the simulation takes infinite time to run. In a event-driven simulation, the ball never stops its motion and the number of events per time step increases. A similar problem is the famous *Zenon Paradox*¹.

Since the height is directly proportional to the energy, the height also scales with the restitution coefficient at every bounce. Consequently, at the i^{th} bounce the damping of the height is proportional to r^i . The total time is given by

$$\begin{aligned} t_{\text{tot}} &= \sum_{i=1}^{\infty} t_i \\ &= 2 \sqrt{\frac{2h^{\text{initial}}}{g}} \sum_{i=1}^{\infty} \sqrt{r^i} \\ &= 2 \sqrt{\frac{2h^{\text{initial}}}{g}} \left(\frac{1}{1 - \sqrt{r}} - 1 \right). \end{aligned} \quad (34)$$

Luding and McNamara introduced in 1998 a coefficient of restitution that is dependent of the time elapsed since the last event occurred. If the time since the last collision of one of the interacting particles $t^{(i)}$ or $t^{(j)}$ is less than t_{contact} , then the coefficient is set to unity, i.e.,

$$r^{i,j} = \begin{cases} r, & \text{for } t^{(i)} > t_{\text{contact}} \text{ or } t^{(j)} > t_{\text{contact}} \\ 1, & \text{otherwise.} \end{cases} \quad (35)$$

With this redefinition of the restitution coefficient, the collision type changes from inelastic to elastic if too many collisions occur during t_{contact} .

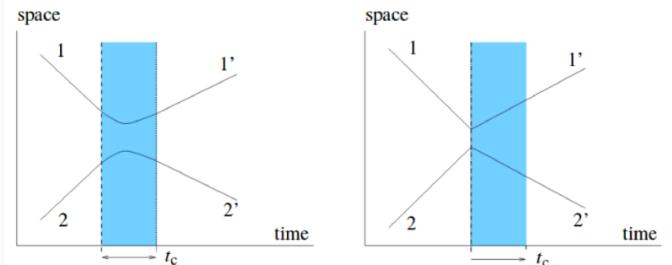


Figure 4: Trajectories of soft (left) and hard (right) particles. The figure

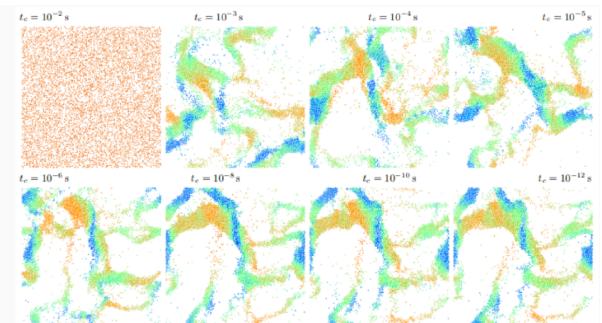


Figure 5: Examples of different contact times t_c . The figure is taken

Contact Dynamics

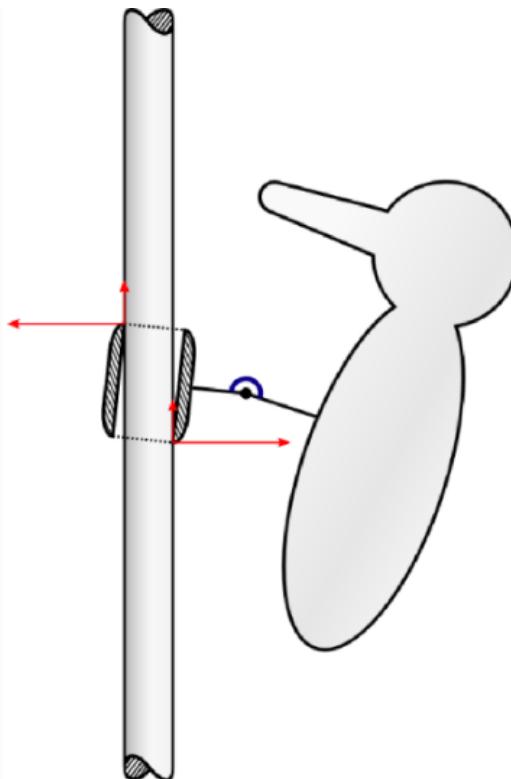


Figure 6: A woodpecker toy as a benchmark example for contact dynamics problems.



Figure 7: Per Lötstedt and Jean-Jacques Moreau contributed to development of contact dynamics.

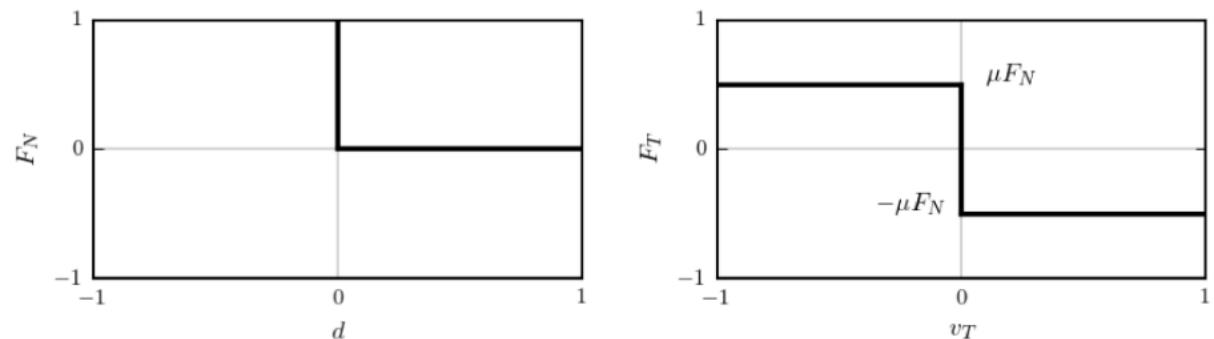


Figure 8: Signorini (left) and Coulomb (right) graphs.

1D Contact

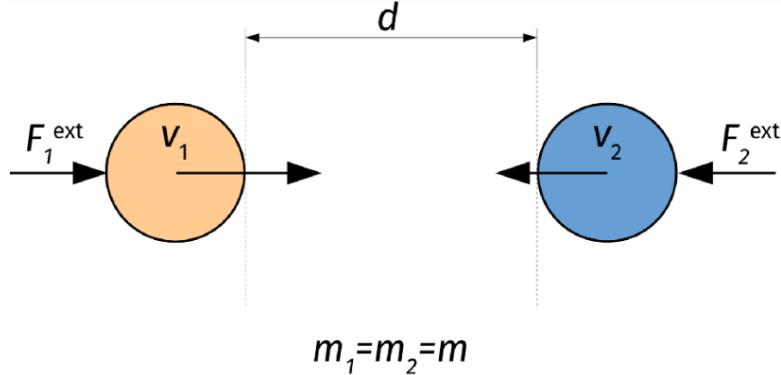


Figure 9: Illustration of a one-dimensional contact.

We have to make sure that these two particles do not overlap. Therefore, we impose constraint forces in such a way that they compensate all other forces which would lead to overlaps. These constraint forces should be defined in such a way that they have no influence on the particle dynamics before and after the contact. The time evolution of the particles' positions and velocities is described by an implicit Euler scheme which is given by

$$\begin{aligned}\mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t) + \Delta t \frac{1}{m_i} \mathbf{F}_i(t + \Delta t), \\ \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t),\end{aligned}\quad (36)$$

where the force consists of an external and a contact term, i.e., $\mathbf{F}_i(t) = \mathbf{F}_i^{\text{ext}}(t) + \mathbf{R}_i(t)$.

So far, we only considered forces that act on the center of mass. However, contact forces act locally on the contact point and not on the center of mass. We therefore introduce a matrix H which transforms local contact forces into particle forces, and the corresponding transpose H^T transforms particle velocities into relative velocities.

This leads to

$$v_n^{\text{loc}} = v_2 - v_1 = \begin{pmatrix} -1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \quad (37)$$

and local forces

$$\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} = \begin{pmatrix} -R_n^{\text{loc}} \\ R_n^{\text{loc}} \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} R_n^{\text{loc}} = H R_n^{\text{loc}}. \quad (38)$$

The equations of motion for both particles are

$$\frac{d}{dt} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \frac{1}{m} \left[\begin{pmatrix} R_1 \\ R_2 \end{pmatrix} + \begin{pmatrix} F_1^{\text{ext}} \\ F_2^{\text{ext}} \end{pmatrix} \right]. \quad (39)$$

Combining the last equation with the transformation rule of Eq. (38) we find

$$\begin{aligned}\frac{dv_n^{\text{loc}}}{dt} &= \begin{pmatrix} -1 & 1 \end{pmatrix} \frac{1}{m} \left[\begin{pmatrix} -1 \\ 1 \end{pmatrix} R_n^{\text{loc}} + \begin{pmatrix} F_1^{\text{ext}} \\ F_2^{\text{ext}} \end{pmatrix} \right] \\ &= \frac{1}{m_{\text{eff}}} R_n^{\text{loc}} + \frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}}),\end{aligned}\quad (40)$$

where $m_{\text{eff}} = m/2$ is the effective mass and $\frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}})$ the acceleration without contact forces.

We integrate the last equation with an implicit Euler method and find:

$$\frac{v_n^{\text{loc}}(t + \Delta t) - v_n^{\text{loc}}(t)}{\Delta t} = \frac{1}{m_{\text{eff}}} R_n^{\text{loc}}(t + \Delta t) + \frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}}). \quad (41)$$

The unknown quantities in this equation are v_n^{loc} and R_n^{loc} . To find a solution, we make use of the Signorini constraint and compute

$$R_n^{\text{loc}}(t + \Delta t) = \frac{v_n^{\text{loc}}(t + \Delta t) - v_n^{\text{loc, free}}(t + \Delta t)}{\Delta t} \quad (42)$$

with

$$v_n^{\text{loc, free}}(t + \Delta t) = v_n^{\text{loc}}(t) + \Delta t \frac{1}{m} (F_2^{\text{ext}} - F_1^{\text{ext}}). \quad (43)$$

We distinguish between the possible cases:

- Particles are not in contact,
- Particles are in closing contact,
- Particles are in persisting contact and
- Particles are in opening contact.

3D Contact

We now extend the described contact dynamics approach to three dimensions. In particular, we consider particle interactions without friction. Thus, we do not need to take into account angular velocities and torques. In three dimensions, velocities and forces are given by

$$\mathbf{v}_{12} = \begin{pmatrix} v_{12}^x \\ v_{12}^y \\ v_{12}^z \end{pmatrix} \quad \mathbf{R}_{12} = \begin{pmatrix} R_{12}^x \\ R_{12}^y \\ R_{12}^z \end{pmatrix} \quad \mathbf{F}_{12}^{\text{ext}} = \begin{pmatrix} F_{12}^{x,\text{ext}} \\ F_{12}^{y,\text{ext}} \\ F_{12}^{z,\text{ext}} \end{pmatrix}. \quad (44)$$

Only normal components v_n^{loc} and R_n^{loc} have to be considered during particle contact. We therefore project all necessary variables onto the normal vector

$$\mathbf{n} = \begin{pmatrix} n^x \\ n^y \\ n^z \end{pmatrix}, \quad (45)$$

and obtain

$$v_n^{\text{loc}} = \mathbf{n} \cdot (\mathbf{v}_2 - \mathbf{v}_1) \quad \mathbf{R}_1 = -\mathbf{n} R_n^{\text{loc}} \quad \mathbf{R}_2 = \mathbf{n} R_n^{\text{loc}}. \quad (46)$$

From the projection, we obtain the matrix H for the coordinate transformation

$$v_n^{\text{loc}} = H^T \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{pmatrix}, \quad \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{pmatrix} = H R_n^{\text{loc}}, \quad (47)$$

with

$$H^T = (-n_x, -n_y, -n_z, n_x, n_y, n_z) \quad (48)$$

Friction can be included by considering angular velocities and torques.

Particles in Fluids

Simulations of particle dynamics in fluids is highly relevant for optimizing certain structures in the sense of minimizing friction and turbulence effects. We therefore consider an incompressible fluid of density ρ and dynamic viscosity μ . It is described by the incompressible *Navier-Stokes equations*

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}(\nabla \mathbf{u}) = -\frac{1}{\rho} \nabla p + \mu \Delta \mathbf{u} \quad (1)$$

The velocity and pressure fields are denoted by $\mathbf{u}(\mathbf{x})$ and $p(\mathbf{x})$, respectively. In the case of constant density ρ , the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \mathbf{u}) = 0 \quad (2)$$

yields $\nabla \cdot \mathbf{u} = 0$.

We classify the fluid flow according to the *Reynold's number*

$$Re = \frac{uh}{\mu} = \begin{cases} \ll 1 & \text{Stokes limit,} \\ \gg 1 & \text{turbulent flow,} \end{cases} \quad (3)$$

where u and h represent a characteristic velocity and length scale, respectively.

There are two possibilities of modeling particle-fluid interactions. First, in a continuum approach the fluid is described by differential equations such as Eqs. 1 and 2. Second, it is possible to use particle-based models of fluids. Different methods are applicable to solve such problems. Some examples include

- Penalty method with MAC
- Finite volume method (FLUENT)
- $k-\epsilon$ model or spectral methods for the turbulent case
- Lattice-Boltzmann methods
- Discrete simulation methods

Based on the fluid motion described by the Navier-Stokes equations, we are able to extract the forces exerted on the particles which enables us to solve their equations of motion. The total drag force is obtained by integrating the **stress tensor** Θ of the fluid over the particles' surfaces

$$\mathbf{F}_D = \int_{\Gamma} \Theta d\mathbf{A}. \quad (4)$$

The stress tensor of the fluid is given by:

$$\Theta_{ij} = -p\delta_{ij} + \eta \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (5)$$

where $\eta = \rho\mu$ is the static viscosity and p the hydrostatic pressure.

In the Stokes limit for $Re \ll 1$, the drag law is given by:

$$F_D = 6\pi\eta Ru, \quad (6)$$

where η is the viscosity of the fluid, R the radius of the particle, u the velocity of the fluid relative to the particle. The Stokes law is exact for $Re = 0$. In the case of turbulent flow for $Re \gg 1$, the drag force is (Newton's law)

$$F_D = 0.22\pi\rho R^2 u^2. \quad (7)$$

The general drag law is

$$F_D = \frac{\pi\eta^2}{8\rho} C_D Re^2. \quad (8)$$

where C_D denotes the drag coefficient. It depends on the velocity of the particle in the fluid, and on the density and the viscosity of the fluid.

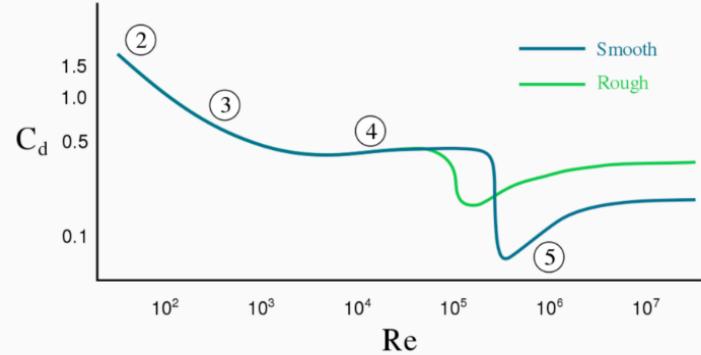


Figure 1: Dependence of the drag coefficient C_D on the Reynold's number.

These laws are based on the assumption of spherical particles and other simplifications, and we may encounter substantial deviations in experiments. In certain cases, it is important to also consider the influence of pressure or velocity gradients which lead to lift forces

$$F_L = \frac{1}{2} C_L \rho A u^2, \quad (9)$$

where C_L denotes the lift coefficient.

In addition to drag and lift forces, rotating particles experience a torque

$$T = \int_{\Gamma} \mathbf{r}_{cm} \wedge \Theta d\mathbf{A} \quad (10)$$

For cylinder of radius R and angular velocity ω , the corresponding Magnus force is

$$F_M = 2\pi R^2 \rho u \omega. \quad (11)$$

There exist empirical relation for drag coefficient in certain Reynold's numbers regimes. For example, one may adopt the following drag coefficient dependence:

$$C_D = \begin{cases} 1 & Re_\rho < 1000, \\ 0.44 & Re_\rho \geq 1000, \end{cases} \quad (12)$$

where $Re_\rho = \frac{\rho_f |v-u| D_s}{\nu}$. Here D_s is the diameter of the particle, and $|v-u|$ is the absolute value of the particle velocities compared to the fluid.

Lattice Boltzmann Method

Based on the Chapman–Enskog theory, it is possible to derive the Navier-Stokes equations from the Boltzmann equation. This connection between fluid dynamics and Boltzmann transport theory allows us to simulate the motion of fluids by solving the corresponding Boltzmann equation on a lattice. The basic idea is that we define on each site x of a lattice on each outgoing bond i a velocity distribution function $f(x, v_i, t)$ whose updates are given by

$$f(x + v_i, v_i, t + 1) - f(x, v_i, t) + F(v_i) = \frac{1}{\tau} [f_i^{\text{eq}} - f(x, v_i, t)] \quad (17)$$

the equilibrium distribution is

$$f_i^{\text{eq}} = n\omega_i \left[1 + \frac{3}{c^2} \mathbf{u} \mathbf{v}_i + \frac{9}{2c^4} (\mathbf{u} \mathbf{v}_i)^2 - \frac{3}{2c^2} \mathbf{u} \mathbf{u} \right]. \quad (18)$$

One possible choice of the weights in two dimensions is

$$\omega_i = \begin{cases} 4/9 & i = 0, \\ 1/9 & i = 1, 2, 3, 4, \\ 1/36 & i = 5, 6, 7, 8. \end{cases} \quad (19)$$

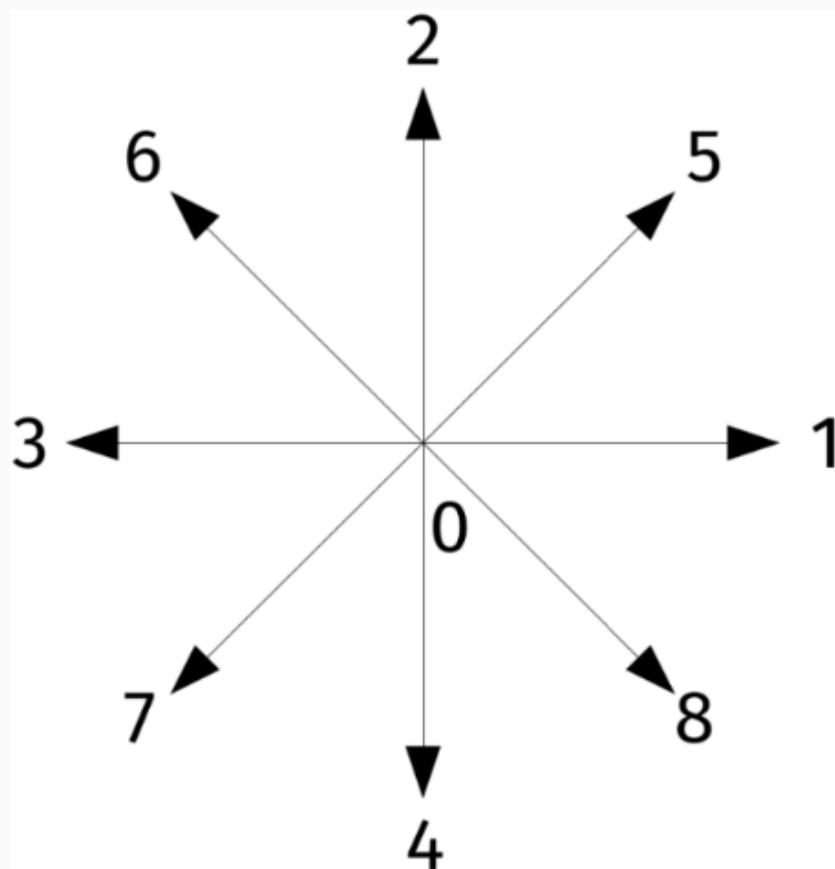


Figure 2: Lattice Boltzmann weights in 2 dimensions

Direct simulation Monte Carlo

Direct Simulation Monte Carlo (DSMC) is a particle-based simulation technique which is appropriate to model particle systems at large Knudsen numbers

$$Kn = \frac{\lambda}{L} \quad (23)$$

where λ is the mean free path and L a characteristic system length scale. It is very popular in aerospace modeling, because the atmosphere is very thinned out at high altitudes and the corresponding Knudsen numbers are large.

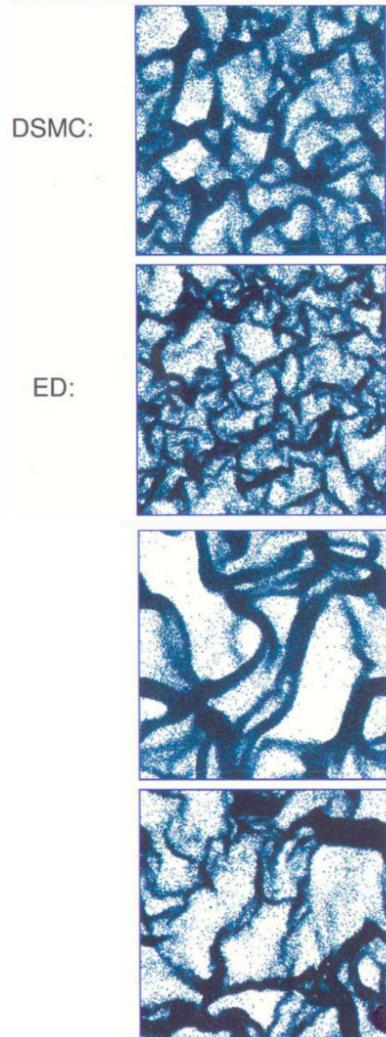
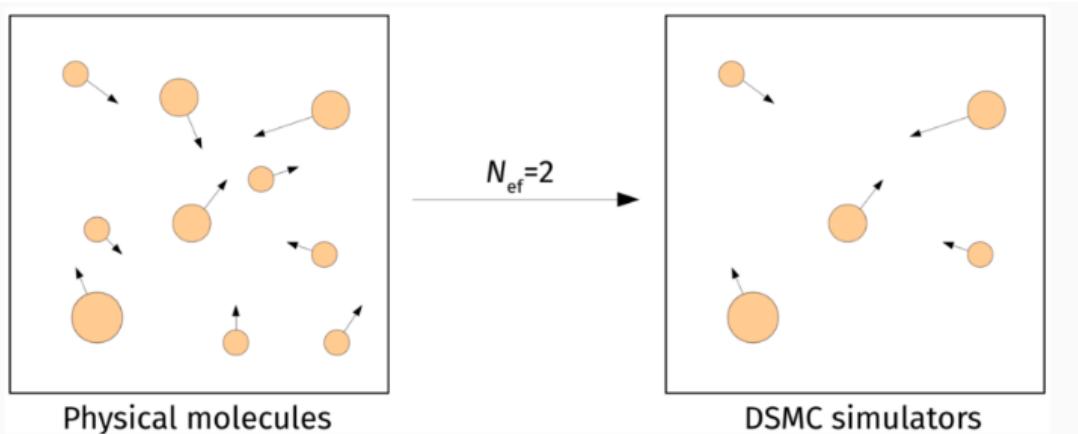


Figure 3: In DSMC N_{ef} simulators represent one physical particle. Collision are modeled by sorting particles into spatial collision cells. We then iterate over all cells and

1. compute the collision frequency in each cell,
2. randomly select collision partners within cell,
3. process each collision.

We note that collision pairs with large relative velocity are more likely to collide but they do not have to be on a collision trajectory. The material surface may be treated with a thermal wall, which resets the velocity of a particle as a biased-Maxwellian distribution

$$P_{v_x}(v_x) = \pm \frac{m}{k_B T_W} v_x e^{-\frac{mv_x^2}{2k_B T_W}} \quad (24)$$

$$P_{v_y}(v_y) = \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{m(v_y - u_W)^2}{2k_B T_W}} \quad (25)$$

$$P_{v_z}(v_z) = \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{mv_z^2}{2k_B T_W}} \quad (26)$$