

Computational Statistical Physics

Part I: Statistical Physics and Phase Transitions

Marina Marinkovic

February 23, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Dates and further information

- Lectures: Wednesday 9.45–11.45 in HCI J 7
- Exercises : Friday 9.45–11.45 in HPT C 103
- Tutors: Doruk Efe Gökmen and Pascal Engler
- Oral exams take place in Summer Exam Session 2022
- Moodle Course Page: <https://moodle-app2.let.ethz.ch/course/view.php?id=17219>

Dates and further information

- 80% of homeworks "meaningfully completed" and submitted
→ 0.25 bonus points. For details consult [course catalogue](#) and [Exercise sheet 00](#) on the course moodle page -> attempt before Friday, February 25
- Homework submission via gitlab: <https://gitlab.ethz.ch> [Detailed instructions in [Exercise sheet 00](#)] on moodle page under **Exercise Materials**.
- If not familiar with git, visit this week's exercises class
- Friday, 25.02., 9.45: git tutorial/setting up homework submission in [HPT C 103](#) Please bring a laptop with you!

Course offered in (selection)

- Mathematics (Master)
- Physics (Master)
- Integrated Building Systems Master (Specialised Courses)
- Civil Engineering (Additional Courses)
- Biomedical Engineering (Master)
- Computational Science and Engineering (Bachelor and Master)
- Computer Science (Bachelor)
- Doctoral programs (qualifying exam: please get in touch before the end of the teaching term)

Phase transitions

- 23.02. Introduction to statistical physics and the Ising model
- 02.03. Monte Carlo methods
- 09.03. Finite size methods and cluster algorithms
- 16.03. Histogram methods
- 23.03. Renormalization group
- 30.03. Boltzmann machines
- 06.04. Non-equilibrium phase transitions

Molecular dynamics

- 13.04. Molecular Dynamics, Verlet scheme, Leapfrog scheme
- 20.04. ETH vacations
- 27.04. Optimization, Linked cell, Lagrange multipliers
- 04.05. Rigid bodies, quaternions
- 11.05. Nosé-Hoover thermostat, stochastic method, constant pressure ensemble

Event-driven dynamics

- 18.05. Event driven, inelastic collisions, friction
- 25.05. Contact dynamics
- 01.06. Advanced topics

Classical statistical mechanics

Classical statistical mechanics



Ringier reception in Zofingen, CH [Source:Wikipedia]

Phase space

Let us consider a classical physical system with N particles whose canonical coordinates and the corresponding conjugate momenta are given by q_1, \dots, q_{3N} and p_1, \dots, p_{3N} , respectively. The $6N$ -dimensional space Γ defined by the latter set of coordinates defines the *phase space*.

Ensemble average

The assumption that all states in an ensemble are reached by the time evolution of the corresponding system is referred to as *ergodicity hypothesis*. We define the *ensemble average* of a quantity $Q(p, q)$ as

$$\langle Q \rangle = \frac{\int Q(p, q) \rho(p, q) \, dp \, dq}{\int \rho(p, q) \, dp \, dq}, \quad (1)$$

where ρ denotes the *phase space density* and $dp \, dq$ is a shorthand notation of $dp^{3N} \, dq^{3N}$.

Hamiltonian

The dynamics of the considered N particles is described by their *Hamiltonian* $\mathcal{H}(p, q)$, i.e., the equations of motions are

$$\dot{p}_i = -\frac{\partial \mathcal{H}}{\partial q_i} \quad \text{and} \quad \dot{q}_i = \frac{\partial \mathcal{H}}{\partial p_i} \quad (i = 1, \dots, 3N). \quad (2)$$

Moreover, we introduce the *phase space density* ρ and find for its temporal change in a volume V with boundary A

$$\frac{\partial}{\partial t} \int \rho \, dV + \int_A \rho v \, dA = 0, \quad (3)$$

where $v = (\dot{p}_1, \dots, \dot{p}_{3N}, \dot{q}_1, \dots, \dot{q}_{3N})$ is a generalized velocity vector.

Liouville Theorem

Applying the divergence theorem to Eq. (3), we find that ρ satisfies the continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0, \quad (4)$$

where $\nabla = (\partial/\partial p_1, \dots, \partial/\partial p_{3N}, \partial/\partial q_1, \dots, \partial/\partial q_{3N})$.

Rewriting Eq. (4) using Poisson brackets

yields *Liouville's Theorem*

$$\frac{\partial \rho}{\partial t} = \{\mathcal{H}, \rho\} \quad (5)$$

which describes the time evolution of the phase space density ρ .

Thermal equilibrium

In *thermal equilibrium*, the system reaches a steady state in which the distribution of the configurations is constant and time-indepedent, i.e., $\partial\rho/\partial t = 0$. Liouville's theorem leads to the following condition

$$v \cdot \nabla \rho = \{\mathcal{H}, \rho\} = 0. \quad (6)$$

The latter equation is satisfied if ρ depends on quantities which are conserved during the time evolution of the system. We then use such a ρ to replace the *time average*

$$\langle Q \rangle = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T Q(p(t), q(t)) dt. \quad (7)$$

by its ensemble average as defined by Eq. (1).

Ensemble average

Considering a discrete and general configuration X , we define the ensemble average as

$$\langle Q \rangle = \frac{1}{\Omega} \sum_X Q(X) \rho(X), \quad (8)$$

where Ω is the normalizing volume such that $\Omega^{-1} \sum_X \rho(X) = 1$. With this definition, systems can be described by means of some macroscopic quantities, such as temperature, energy and pressure.

Ensembles

- Microcanonical ensemble: constant E, V, N
- Canonical ensemble: constant T, V, N
- Canonical pressure ensemble: constant T, p, N
- Grandcanonical ensemble: constant T, V, μ

Microcanonical ensemble

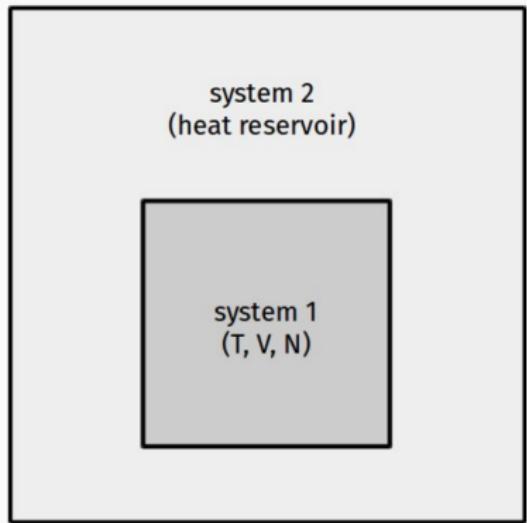
The *microcanonical ensemble* is defined by a constant number of particles, volume and energy. Thus, any configuration X of the system has the same energy $E(X) = \text{const}$. Without proof, the phase space density is also constant and given by

$$\rho(X) = \frac{1}{Z_{\text{mc}}} \delta(\mathcal{H}(X) - E), \quad (9)$$

with Z_{mc} being the *partition function* of the microcanonical ensemble

$$Z_{\text{mc}} = \sum_X \delta(\mathcal{H}(X) - E).$$

Canonical ensemble



In a canonical ensemble setup, the system we study (system 1) is coupled to a heat reservoir (system 2) that guarantees a constant temperature. [Böttcher, Herrmann, Comp.Stat.Phys., Cambridge University Press, 2021]

Canonical ensemble

At a given temperature T , the probability for a system to be in a certain configuration X with energy $E(X)$ is given by

$$\rho(X) = \frac{1}{Z_T} \exp\left[-\frac{E(X)}{k_B T}\right], \quad (10)$$

with

$$Z_T = \sum_X \exp\left[-\frac{E(X)}{k_B T}\right] \quad (11)$$

being the partition function of the canonical ensemble. According to the prior definition in Eq. (8), the ensemble average of a quantity Q is then given by

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

Ising model

Ising Model

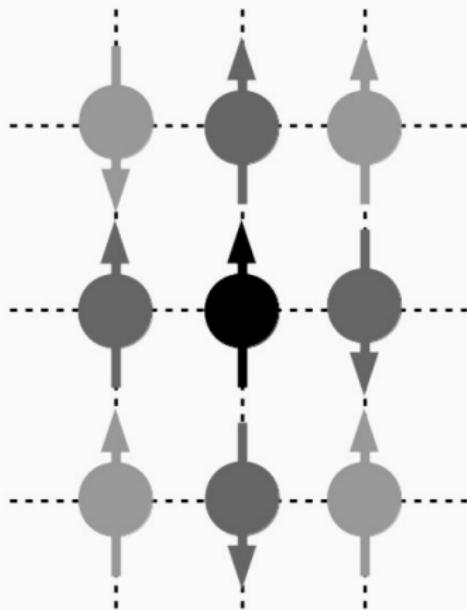


Figure 1: An illustration of the interaction of a magnetic dipole (black) with its nearest neighbors (dark grey) on a two-dimensional lattice.
[Böttcher, Herrmann, Comp.Stat.Phys., Cambridge University Press, 2021]

Ising Model

We consider a two-dimensional lattice with sites $\sigma_i \in \{1, -1\}$, which only interact with their nearest neighbors. Their interaction is given by the Hamiltonian

$$\mathcal{H}(\{\sigma\}) = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j - H \sum_{i=1}^N \sigma_i, \quad (13)$$

where the first term denotes the interaction between all nearest neighbors represented by a sum over $\langle i, j \rangle$, and the second one the interaction of each site with an external magnetic field H .

- Ferromagnetic case: $J > 0$ (parallel spins)
- Antiferromagnetic case: $J < 0$ (anti-parallel spins)
- No interaction: $J = 0$

Phase Transition in the Ferromagnetic Case

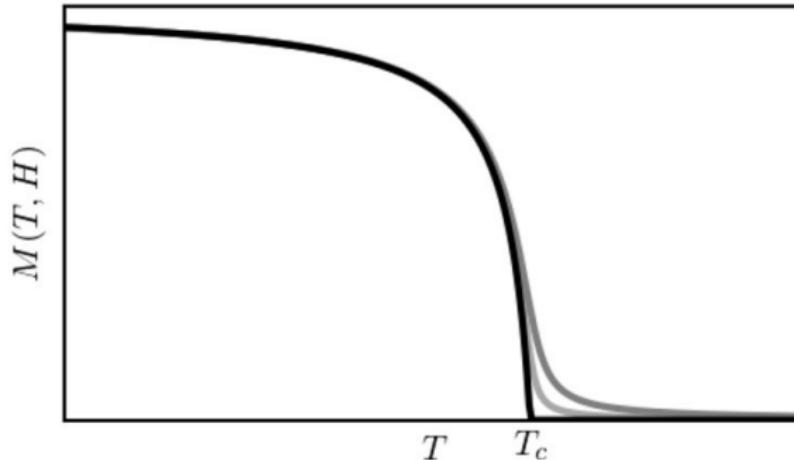


Figure 2: The magnetization $M(T, H)$ for different fields $H \geq 0$ as a function of T . For $T \leq T_c$, the system is characterized by a ferromagnetic phase. The black solid line represents the spontaneous magnetization $M_S(T)$ for $H = 0$ and should be interpreted in the sense that $\lim_{H \rightarrow 0^+} M(T, H)$. [Böttcher, Herrmann, Comp. Stat. Phys., Cambridge University Press, 2021]

Phase Transition in the Ferromagnetic Case

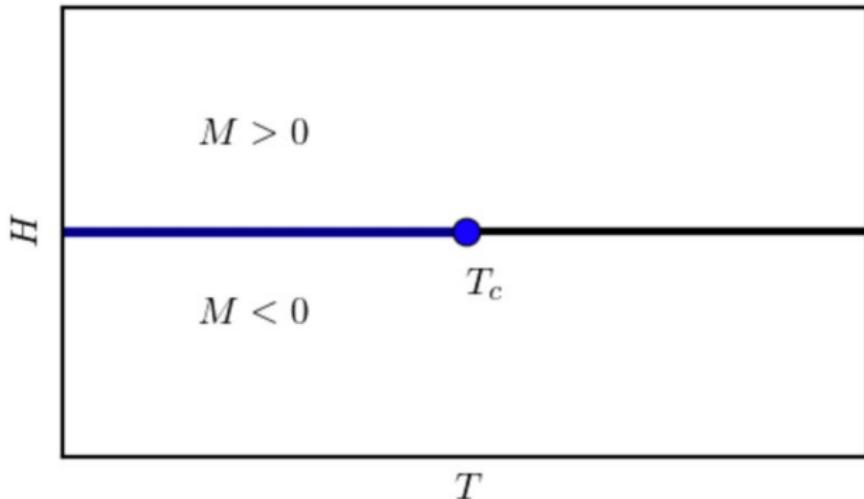


Figure 3: The first-order transition as a consequence of a sign change of the external field. [Böttcher, Herrmann, Comp.Stat.Phys., Cambridge University Press, 2021]

Order parameter

The magnetization is defined as

$$M(T, H) = \left\langle \frac{1}{N} \sum_{i=1}^N \sigma_i \right\rangle, \quad (14)$$

and corresponds to the ensemble average of the mean value of all spins.

On average, $M(T)$ vanishes since for every configuration there exists one of opposite sign which neutralizes the other one. As a consequence, we define the *order parameter* of the Ising model as

$$M_S(T) = \lim_{H \rightarrow 0^+} \left\langle \frac{1}{N} \sum_{i=1}^N \sigma_i \right\rangle \quad (15)$$

and refer to it as the *spontaneous magnetization*.

Magnetic Domains

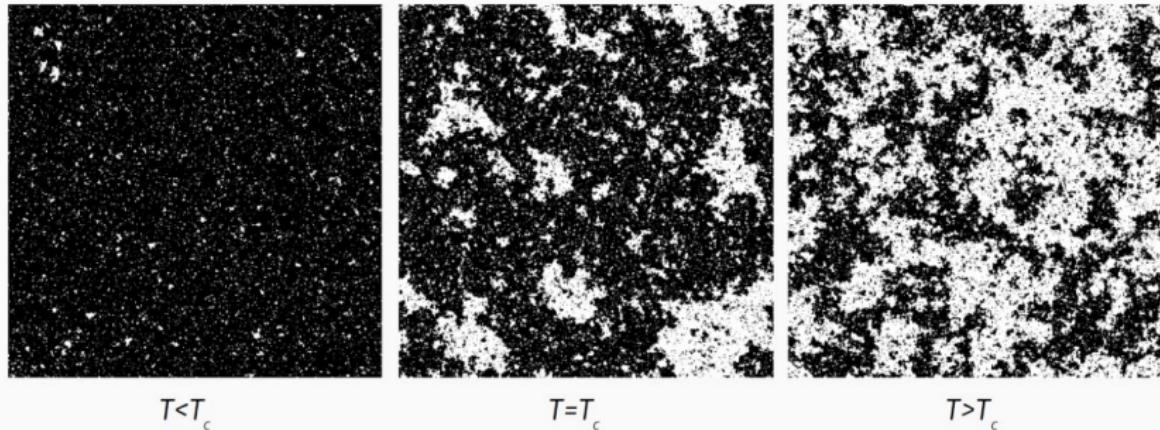


Figure 4: The formation of magnetic domains in the Ising model for temperatures $T < T_c$. For larger temperatures, the configurations are random due to thermal fluctuations. The simulations have been performed on a square lattice with 512×512 sites using <https://mattbierbaum.github.io/isng.js/>.

Critical Exponents I

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 $\delta = 4.790$

Fluctuations

The magnetic susceptibility is defined as the change of the magnetization M in response to an applied magnetic field H , i.e.,

$$\chi(T) = \frac{\partial M(T, H)}{\partial H}. \quad (18)$$

We now use the definition of the spontaneous magnetization given by Eq. (15) and plug it into Eq. (18) leading to

$$\chi(T) = \lim_{H \rightarrow 0^+} \frac{\partial \langle M(T, H) \rangle}{\partial H} \quad (19)$$

$$= \lim_{H \rightarrow 0^+} \frac{\partial}{\partial H} \underbrace{\frac{\sum_{\{\sigma\}} \sum_{i=1}^N \sigma_i \exp\left(\frac{E_0 + H \sum_{i=1}^N \sigma_i}{k_B T}\right)}{\sum_{\{\sigma\}} \exp\left(\frac{E_0 + H \sum_{i=1}^N \sigma_i}{k_B T}\right)}}_{=Z_T(\mathcal{H})}, \quad (20)$$

Fluctuations

Using the product rule yields

$$\begin{aligned}\chi(T) &= \lim_{H \rightarrow 0^+} \frac{1}{Nk_B T} \frac{\sum_{\{\sigma\}} \left(\sum_{i=1}^N \sigma_i \right)^2 \exp \left(\frac{E_0 + H \sum_{i=1}^N \sigma_i}{k_B T} \right)}{Z_T(\mathcal{H})} \\ &\quad - \frac{1}{Nk_B T} \frac{\left[\sum_{\{\sigma\}} \sum_{i=1}^N \sigma_i \exp \left(\frac{E_0 + H \sum_{i=1}^N \sigma_i}{k_B T} \right) \right]^2}{[Z_T(\mathcal{H})]^2} \\ &= \frac{N}{k_B T} \left[\langle M_S(T)^2 \rangle - \langle M_S(T) \rangle^2 \right] \geq 0.\end{aligned}\tag{21}$$

The last equation defines fluctuation-dissipation theorem for the magnetic susceptibility. Analogously, the specific heat is connected to energy fluctuations as:

$$C(T) = \lim_{H \rightarrow 0^+} \frac{\partial \langle E \rangle}{\partial T} = \frac{1}{(k_B T)^2} \left[\langle E(T)^2 \rangle - \langle E(T) \rangle^2 \right].\tag{22}$$

Critical Exponents II

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 $\lim_{s \rightarrow 0} \frac{|x|^{-s} - 1}{s} = -\ln|x|$.

Critical Exponents II

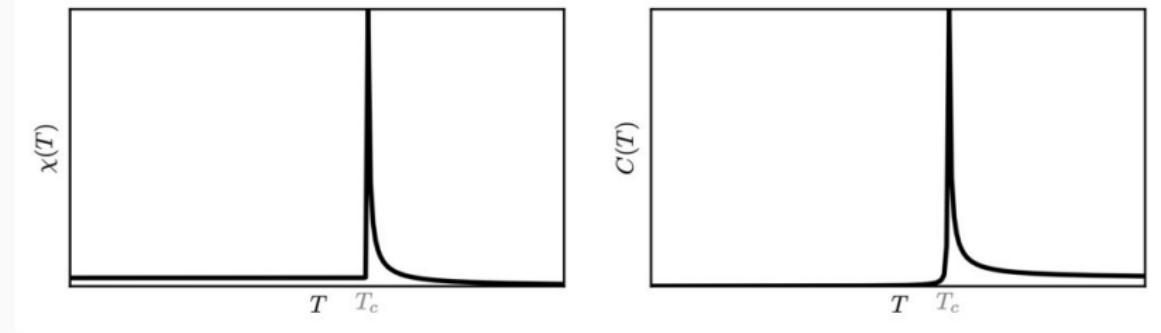


Figure 5: Susceptibility and specific heat as a function of temperature for the three dimensional Ising model. Both quantities diverge at the critical temperature T_c in the thermodynamic limit. [Böttcher, Herrmann, Comp.Stat.Phys., Cambridge University Press, 2021]

Correlation length

The correlation function is defined by

$$G(r_1, r_2; T, H) = \langle \sigma_1 \sigma_2 \rangle - \langle \sigma_1 \rangle \langle \sigma_2 \rangle, \quad (25)$$

where the vectors r_1 and r_2 pointing in the direction of lattice sites 1 and 2. If the system is translational and rotational invariant, the correlation function only depends on $r = |r_1 - r_2|$. At the critical point, the correlation function decays as

$$G(r; T_c, 0) \sim r^{-d+2-\eta}, \quad (26)$$

where η is another critical exponent and d the dimension of the system.

- 2D: $\eta = 1/4$
- 3D: $\eta \approx 0.036$

Correlation length

For temperatures away from the critical temperature, the correlation function exhibits an exponential decay

$$G(r; T, 0) \sim r^{-\vartheta} e^{-r/\xi}, \quad (27)$$

where ξ defines the *correlation length*. The exponent ϑ equals 2 above and 1/2 below the transition point. In the vicinity of T_c , the correlation length ξ diverges since

$$\xi(T) \sim |T - T_c|^{-\nu} \quad (28)$$

- 2D: $\nu = 1$
- 3D: $\nu \approx 0.63$

Critical exponents and universality

The aforementioned six critical exponents are connected by four scaling laws

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

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

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

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

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.

Critical exponents and universality

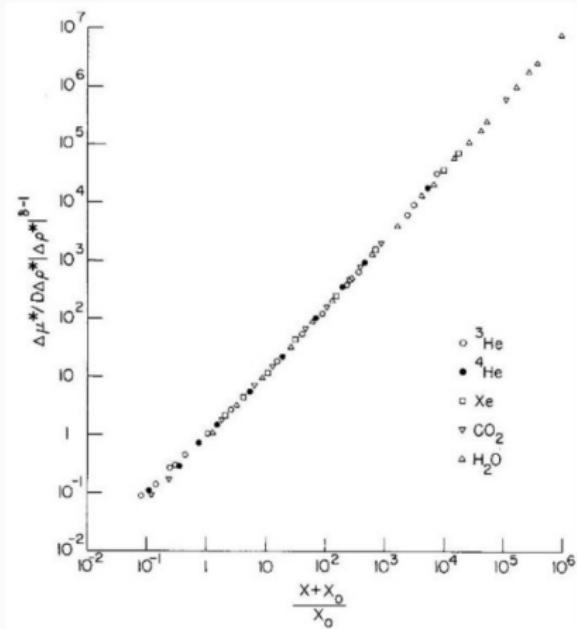


Figure 6: Universal scaling for five different gases. The scaling variable is defined as $x = \Delta T |\Delta \rho|^{-1/\beta}$ and x_0 depends on the amplitude B of the power-law for the coexistence curve $\Delta \rho = B \Delta T^\beta$

[Sengers, Sengers, Croxton, Prog. in liquid physics, Wiley, 1978]

Critical exponents and universality

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)

Computational Statistical Physics

Part I: Statistical Physics and Phase Transitions

Marina Marinkovic

March 1, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



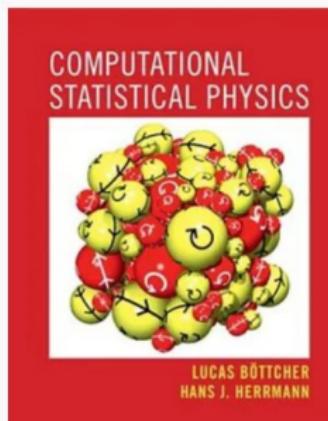
402-0812-00L
FS 2022

Dates and further information

- Lectures: Wednesday 9.45–11.45 in HCI J 7
- Exercises : Friday 9.45–11.45 in HPT C 103
- Tutors: Doruk Efe Gökmen and Pascal Engler
- Oral exams take place in Summer Exam Session 2022
- Moodle Course Page: <https://moodle-app2.let.ethz.ch/course/view.php?id=17219>

Sources

- Hans J. Herrmann and Lucas Böttcher lecture notes
- Slides from FS 2019 by Lucas Böttcher
- Book by L.Böttcher and H. J. Herrmann "Computational Statistical Physics", Cambridge University Press, 2021



Monte Carlo methods

Monte Carlo Methods

Monte Carlo: a numerical method for estimating high-dimensional integrals by random sampling

JOURNAL OF THE AMERICAN STATISTICAL ASSOCIATION

Number 247

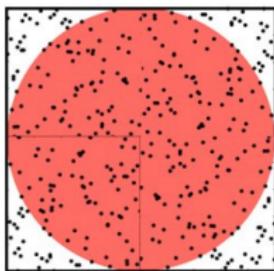
SEPTEMBER 1949

Volume 44

THE MONTE CARLO METHOD

NICHOLAS METROPOLIS AND S. ULAM
Los Alamos Laboratory

We shall present here the motivation and a general description of a method dealing with a class of problems in mathematical physics. The method is, essentially, a statistical approach to the study of differential equations, or more generally, of integro-differential equations that occur in various branches of the natural sciences.



Monte Carlo Methods

The main steps of the Monte Carlo sampling are

1. Choose randomly a new configuration in phase space based on a Markov chain.
2. Accept or reject the new configuration, depending on the strategy used (e.g., Glauber dynamics).
3. Compute the physical quantity and add it to the averaging procedure.
4. Repeat the previous steps.

Markov chains

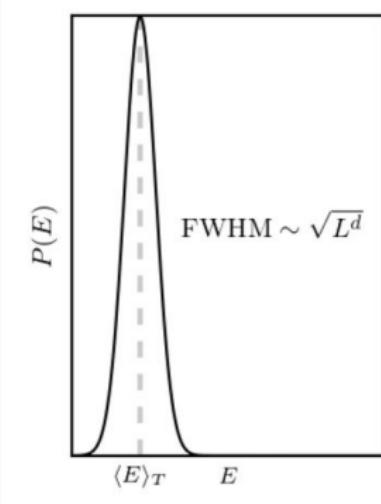


Figure 1: Example of an energy distribution with a system size L dependence of the distribution width which scales as $\sim \sqrt{L^d}$ where d is the system dimension.

Markov chains

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*: $T(X \rightarrow Y) = T(Y \rightarrow X)$.

Markov chains

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

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

Markov chains

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

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

Markov chains

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

To sample all the relevant regions of the phase space of our system, the Markov chain probability $W(\cdot)$ has to depend on the system properties. To achieve that, we impose the distribution of the stationary states p_{st} as the equilibrium distribution of the physical system p_{eq} (a real and measurable distribution):

$$\frac{dp(X, \tau)}{d\tau} = 0 \Leftrightarrow p_{\text{st}} \stackrel{!}{=} p_{\text{eq}}. \quad (3)$$

Markov chains

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*.

Markov chains

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).

M(RT)² algorithm (*Metropolis-Hastings algorithm*)

THE JOURNAL OF CHEMICAL PHYSICS

VOLUME 21, NUMBER 6

JUNE, 1953

Equation of State Calculations by Fast Computing Machines

NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
Los Alamos Scientific Laboratory, Los Alamos, New Mexico

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.



M(RT)² algorithm

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 $\exp\left(-\frac{\Delta E}{k_B T}\right)$.

M(RT)² algorithm

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

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.

Glauber dynamics

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

Glauber dynamics

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

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.

Glauber dynamics

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

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

Glauber dynamics

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*.

Binary mixtures

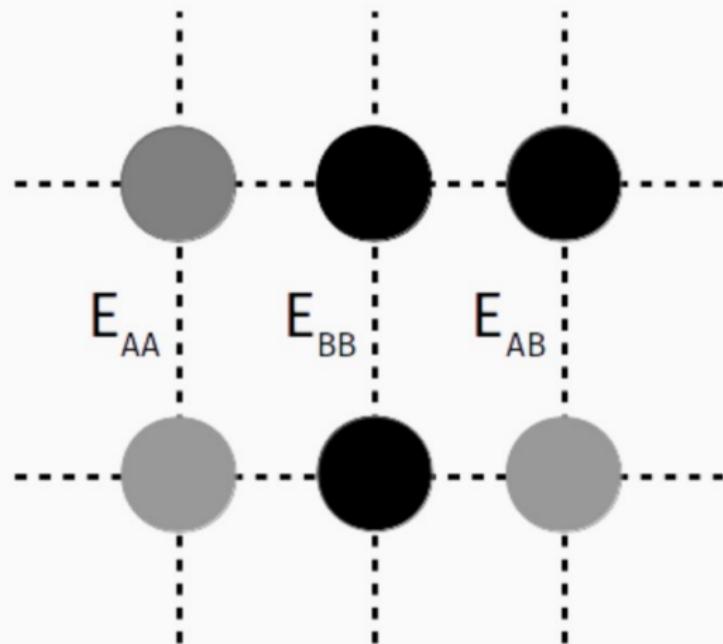


Figure 2: An example of a binary mixture consisting of two different atoms A and B.

Binary mixtures

Kawasaki dynamics

- Choose a $A - B$ bond,
- Compute ΔE for $A - B \rightarrow B - A$,
- Metropolis: If $\Delta E \leq 0$ flip, else flip with probability

$$p = \exp\left(\frac{-\Delta E}{k_B T}\right),$$

- Glauber: Flip with probability

$$p = \exp\left(-\frac{\Delta E}{k_B T}\right) / \left[1 + \exp\left(-\frac{\Delta E}{k_B T}\right)\right].$$

This procedure is very similar to the previously discussed update schemes. The only difference is that the magnetization is kept constant.

Creutz Algorithm

VOLUME 50, NUMBER 19

PHYSICAL REVIEW LETTERS

9 MAY 1983

Microcanonical Monte Carlo Simulation

Michael Creutz

Department of Physics, Brookhaven National Laboratory, Upton, New York 11973

(Received 24 February 1983)

A new algorithm for the simulation of statistical systems is presented. The procedure produces a random walk through configurations of a constant total energy. It is computationally simple and applicable to systems of both discrete and continuous variables.

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

Creutz Algorithm

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

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.

Creutz Algorithm

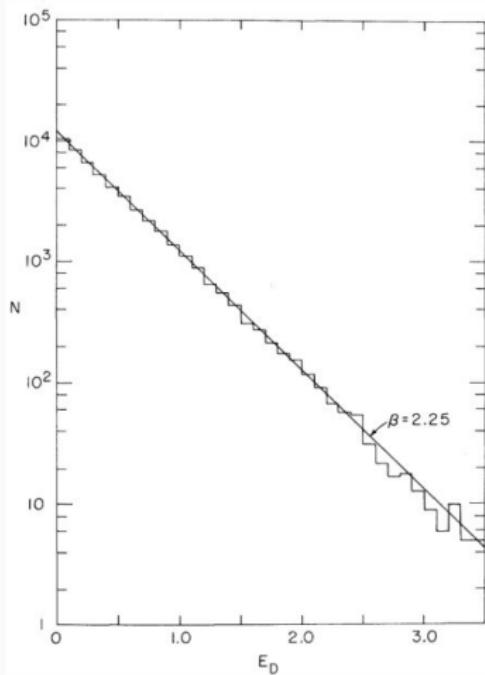


Figure 4: The distribution of the demon energy E_D is exponentially distributed. Based on the Boltzmann factor, it is possible to extract the inverse temperature $\beta = (k_B T)^{-1} = 2.25$. The figure is taken from Ref. shown in Figure 3.

Boundary conditions

For finite lattices, the following boundary conditions might be used:

- Open boundaries, i.e., no neighbors at the edges of the system,
- fixed boundary conditions,
- and periodic boundaries.

Correlation length

The correlation function is defined by

$$G(r_1, r_2; T, H) = \langle \sigma_1 \sigma_2 \rangle - \langle \sigma_1 \rangle \langle \sigma_2 \rangle, \quad (25)$$

where the vectors r_1 and r_2 pointing in the direction of lattice sites 1 and 2. If the system is translational and rotational invariant, the correlation function only depends on $r = |r_1 - r_2|$. At the critical point, the correlation function decays as

$$G(r; T_c, 0) \sim r^{-d+2-\eta}, \quad (26)$$

where η is another critical exponent and d the dimension of the system.

- 2D: $\eta = 1/4$
- 3D: $\eta \approx 0.036$

Correlation length

For temperatures away from the critical temperature, the correlation function exhibits an exponential decay

$$G(r; T, 0) \sim r^{-\vartheta} e^{-r/\xi}, \quad (27)$$

where ξ defines the *correlation length*. The exponent ϑ equals 2 above and 1/2 below the transition point. In the vicinity of T_c , the correlation length ξ diverges since

$$\xi(T) \sim |T - T_c|^{-\nu} \quad (28)$$

- 2D: $\nu = 1$
- 3D: $\nu \approx 0.63$

Critical exponents and universality

The aforementioned six critical exponents are connected by four scaling laws

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

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

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

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

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.

Critical exponents and universality

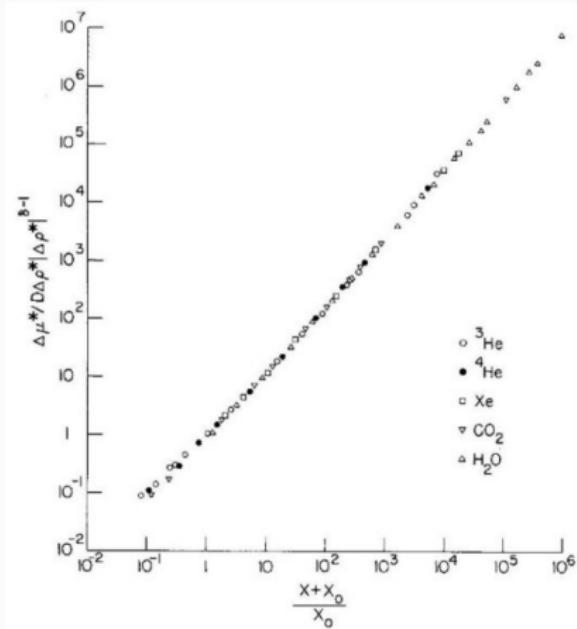


Figure 6: Universal scaling for five different gases. The scaling variable is defined as $x = \Delta T |\Delta \rho|^{-1/\beta}$ and x_0 depends on the amplitude B of the power-law for the coexistence curve $\Delta \rho = B \Delta T^\beta$

[Sengers, Sengers, Croxton, Prog. in liquid physics, Wiley, 1978]

Critical exponents and universality

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)

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).

Temporal Correlations

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 $A(\infty)$.

Temporal Correlations

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

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

Temporal Correlations

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 !

Temporal Correlations

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.

Temporal Correlations

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

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

Temporal Correlations

The *linear* correlation time τ_A^{nl} describes the relaxation towards equilibrium

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

Temporal Correlations

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.

Temporal Correlations

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

$$z_\sigma - z_\sigma^{\text{nl}} = \beta, \tag{25}$$

$$z_E - z_E^{\text{nl}} = 1 - \alpha, \tag{26}$$

$$\tag{27}$$

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

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

Decorrelated configurations

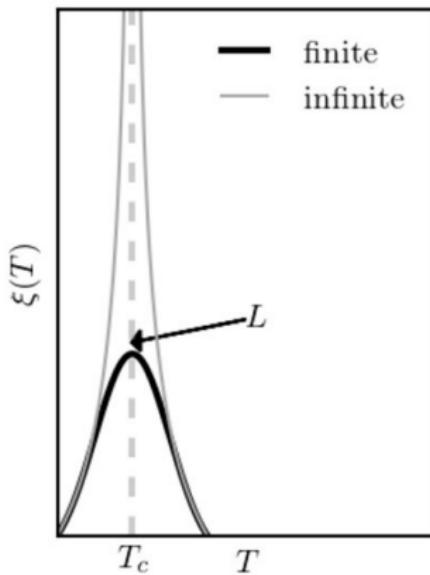


Figure 5: The correlation length diverges in an infinite system at T_c according to the definition of the correlation length from last week's lecture. In a finite system, however, we observe a round off and the correlation length approaches the system size L at T_c .

Decorrelated configurations

To ensure not to sample correlated configurations one should

- first reach equilibrium (discard $n_0 = c\tau^{\text{nl}}(T)$ configurations),
- only sample every $n_e^{\text{th}} = c\tau(T)$ 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.

Computational Statistical Physics

Part I: Statistical Physics and Phase Transitions

Marina Marinkovic

March 9, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Monte Carlo Methods – cont.

Creutz Algorithm

VOLUME 50, NUMBER 19

PHYSICAL REVIEW LETTERS

9 MAY 1983

Microcanonical Monte Carlo Simulation

Michael Creutz

Department of Physics, Brookhaven National Laboratory, Upton, New York 11973

(Received 24 February 1983)

A new algorithm for the simulation of statistical systems is presented. The procedure produces a random walk through configurations of a constant total energy. It is computationally simple and applicable to systems of both discrete and continuous variables.

Figure 1: An algorithm to perform microcanonical Monte Carlo simulations, i.e., system at constant energy.

Creutz Algorithm

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

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.

Creutz Algorithm

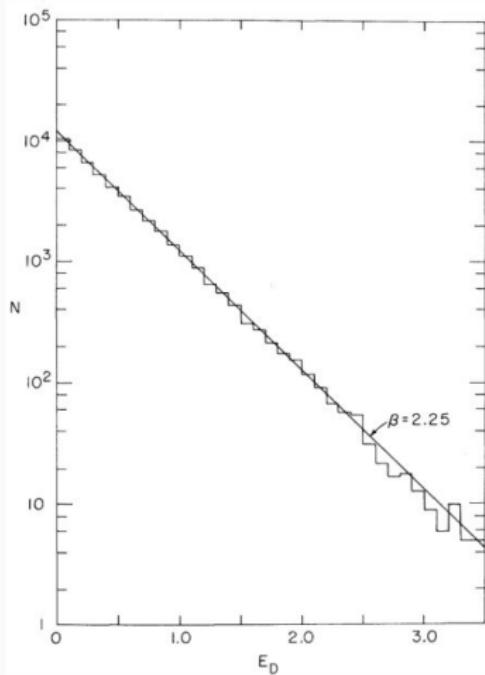


Figure 2: The distribution of the demon energy E_D is exponentially distributed. Based on the Boltzmann factor, it is possible to extract the inverse temperature $\beta = (k_B T)^{-1} = 2.25$. The figure is taken from Ref. shown in Figure 3.

Finite size methods

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.

Finite size methods

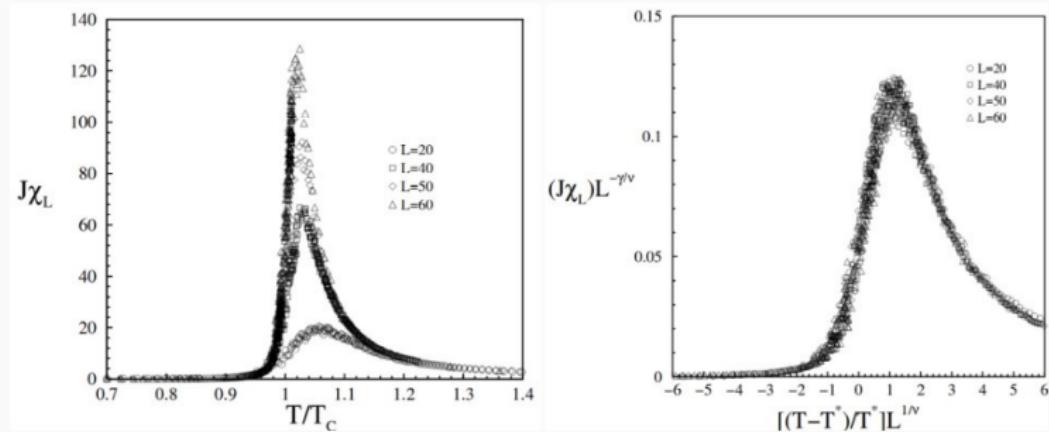


Figure 3: The system size dependence of the susceptibility and the corresponding finite size scaling. The figure is taken from [Da Silva et al., Braz. J. Phys. 32, 2002].

Finite size methods

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 $F_{\chi} \left[(T - T_c) L^{\frac{1}{\nu}} \right] \sim \left(|T - T_c| L^{\frac{1}{\nu}} \right)^{-\gamma}$ as $L \rightarrow \infty$.

Finite size methods

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

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

Binder Cumulant

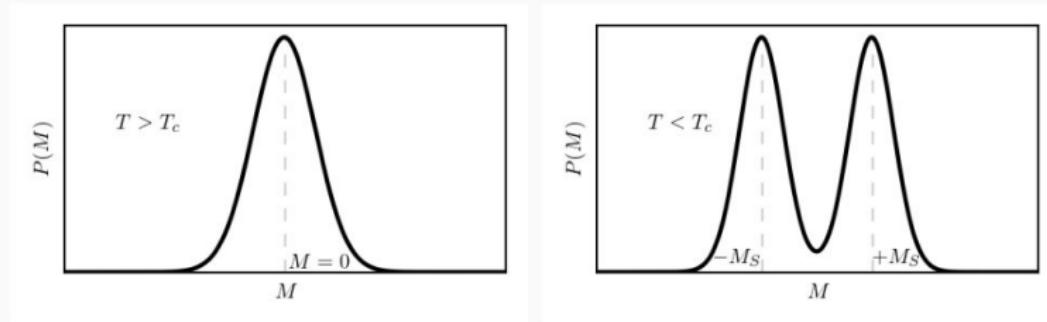


Figure 4: The distribution $P(M)$ of the magnetization M above a below the critical temperature T_c .

Binder Cumulant

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 $T > T_c$.

Binder Cumulant

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

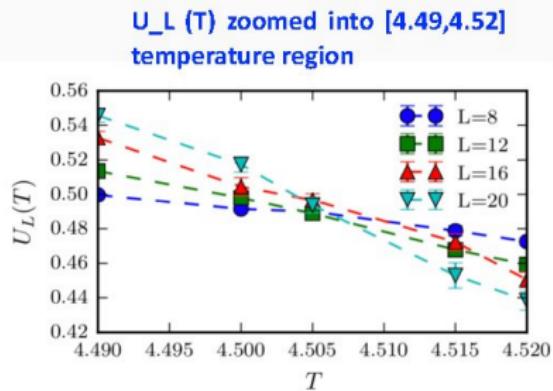
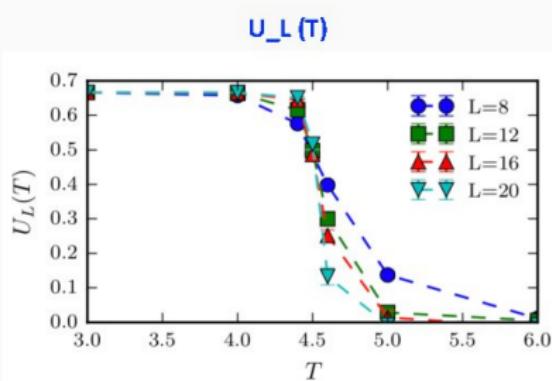
$$P_L(M) = \frac{1}{2} \sqrt{\frac{L^d}{\pi \sigma_l}} \left\{ \exp \left[-\frac{(M - M_S)^2 L^d}{\sigma_L} \right] + \exp \left[-\frac{(M + M_S)^2 L^d}{\sigma_L} \right] \right\} \quad (10)$$

Binder Cumulant

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

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

Binder Cumulant



Corrections to Scaling

Far away from T_c we cannot observe a clear power law behavior anymore, and corrections to scaling have to be employed, i.e.,

$$M(T) = A_0 (T_c - T)^\beta + A_1 (T_c - T)^{\beta_1} + \dots, \quad (12)$$

$$\xi(T) = C_0 (T_c - T)^\nu + C_1 (T_c - T)^{\nu_1} + \dots, \quad (13)$$

with $\beta_1 > \beta$ and $\nu_1 < \nu$.

Corrections to Scaling

These corrections are very important for high quality data, where the errors are small and the deviations become visible. The scaling functions must also be generalized as

$$M(T, L) = L^{-\frac{\beta}{\nu}} F_M \left[(T - T_c) L^{\frac{1}{\nu}} \right] + L^{-x} F_M^1 \left[(T - T_c) L^{\frac{1}{\nu}} \right] + \dots \quad (14)$$

with $x = \max \left[\frac{\beta}{\nu}, \frac{\beta}{\nu_1}, \frac{\beta}{\nu} - 1 \right]$.

First Order Transition

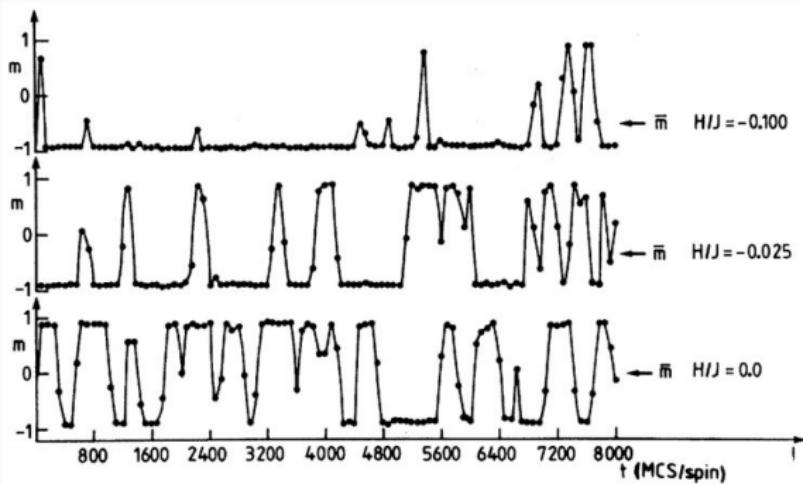


Figure 5: The magnetization exhibits a switching behavior if the field vanishes. For non-zero magnetic fields, the magnetization is driven in the direction of the field. The figure is taken from [Binder and Landau, Phys. Rev. B30 3 (1984)]

First Order Transition

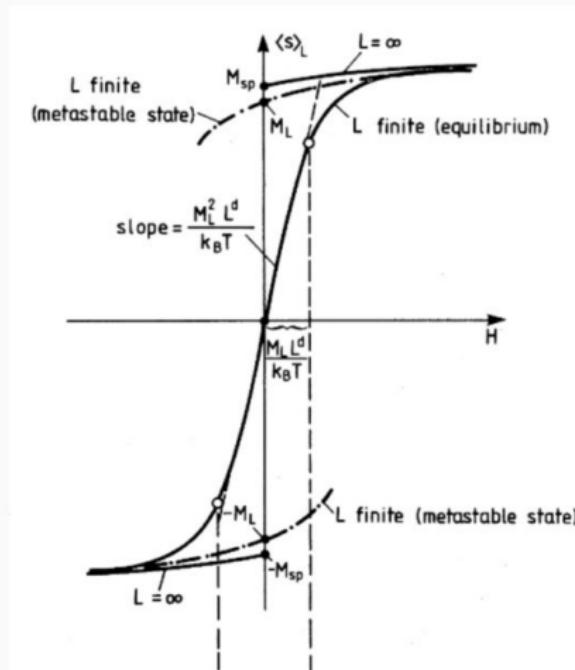


Figure 6: Hysteresis can be found by varying the field from negative to positive values and back. The figure is taken from [Binder and Landau, Phys. Rev. B30 3 (1984)]

First Order Transition

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

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

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

First Order Transition

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).

Cluster algorithms

Potts model

The Hamiltonian of the system is defined as

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \delta_{\sigma_i \sigma_j} - H \sum_i \sigma_i, \quad (17)$$

where $\sigma_i \in \{1, \dots, q\}$ and $\delta_{\sigma_i \sigma_j}$ is unity when nodes i and j are in the same state. The Potts model exhibits a first order transition at the critical temperature in two dimensions for $q > 4$, and for $q > 2$ for dimensions larger than two.

Potts model

For $q = 2$, the Potts model is equivalent to the Ising model.

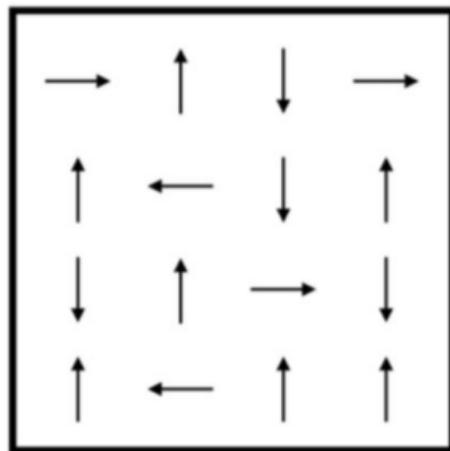


Figure 7: Example of a 4x4 Potts lattice for $q=4$.

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.,

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

The Kasteleyn and Fortuin Theorem

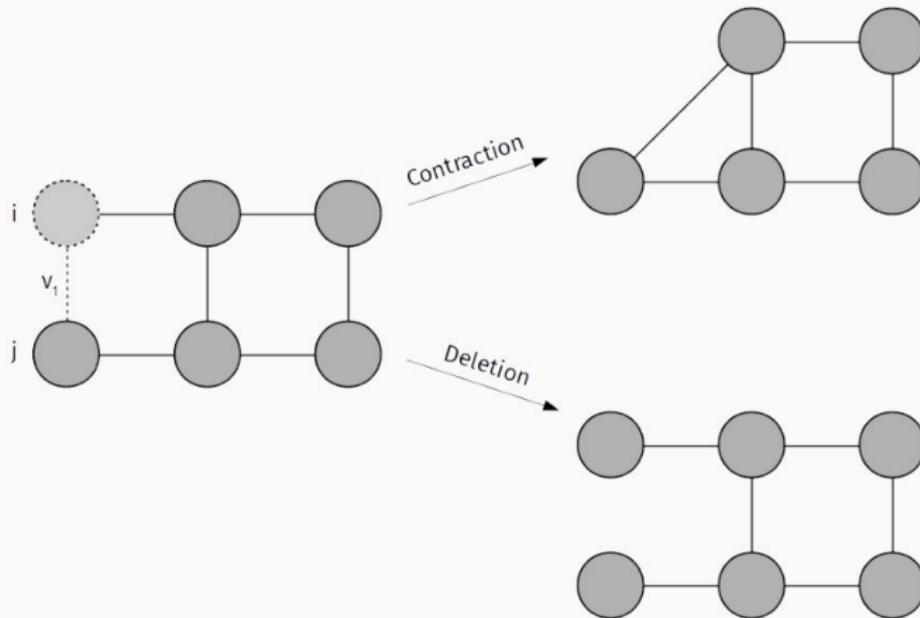


Figure 8: Contraction and deletion on a graph.

The Kasteleyn and Fortuin Theorem

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

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

The Kasteleyn and Fortuin Theorem

We now consider a graph where bond ν_1 connects two nodes i and j with states σ_i and σ_j , respectively. If we would delete bond ν_1 , the partition function is

$$Z_D = \sum_X \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu}. \quad (20)$$

The Kasteleyn and Fortuin Theorem

We can thus rewrite Eq. (19) as

$$Z = \sum_X e^{-\beta J \epsilon_{\nu_1}} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} = \sum_{X: \sigma_i = \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} + e^{-\beta J} \sum_{X: \sigma_i \neq \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu}$$

where the first part is the partition function of the contracted graph Z_C and the second part is given by the identity

$$\sum_{X: \sigma_i \neq \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} = \sum_X \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} - \sum_{X: \sigma_i = \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} = Z_D - Z_C. \quad (21)$$

The Kasteleyn and Fortuin Theorem

Summarizing the latter results, we find

$$Z = Z_C + e^{-\beta J} (Z_D - Z_C) = pZ_C + (1-p)Z_D, \quad (22)$$

where $p = -e^{-\beta J}$. To be more precise, we expressed the partition function Z as the contracted and deleted partition functions at bond ν_1 . We apply the latter procedure to another bond ν_2 and find

$$Z = p^2 Z_{C_{\nu_1}, C_{\nu_2}} + p(1-p) Z_{C_{\nu_1}, D_{\nu_2}} + (1-p)^2 Z_{D_{\nu_1}, D_{\nu_2}}. \quad (23)$$

The Kasteleyn and Fortuin Theorem

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

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

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.,

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

Coniglio-Klein clusters

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

Coniglio-Klein clusters

We then obtain for acceptance probabilities

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

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

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

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \vec{S}_i \cdot \vec{S}_j + \vec{H} \sum_i \vec{S}_i. \quad (29)$$

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

Other Ising-like models

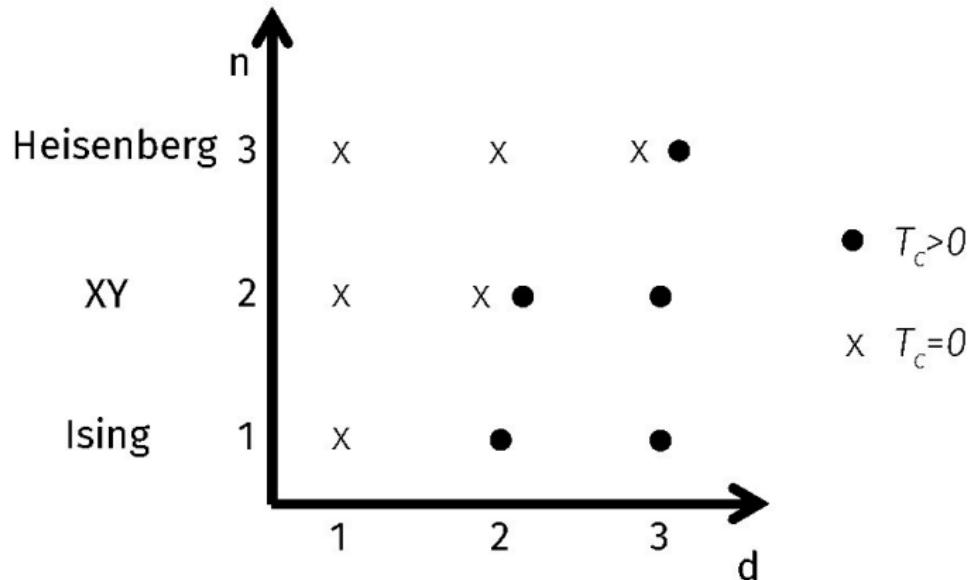


Figure 9: The dependence of the critical temperature on the number of vector components n .

Computational Statistical Physics

Part I: Statistical Physics and Phase Transitions

Marina Marinkovic

March 16, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

The Kasteleyn and Fortuin Theorem

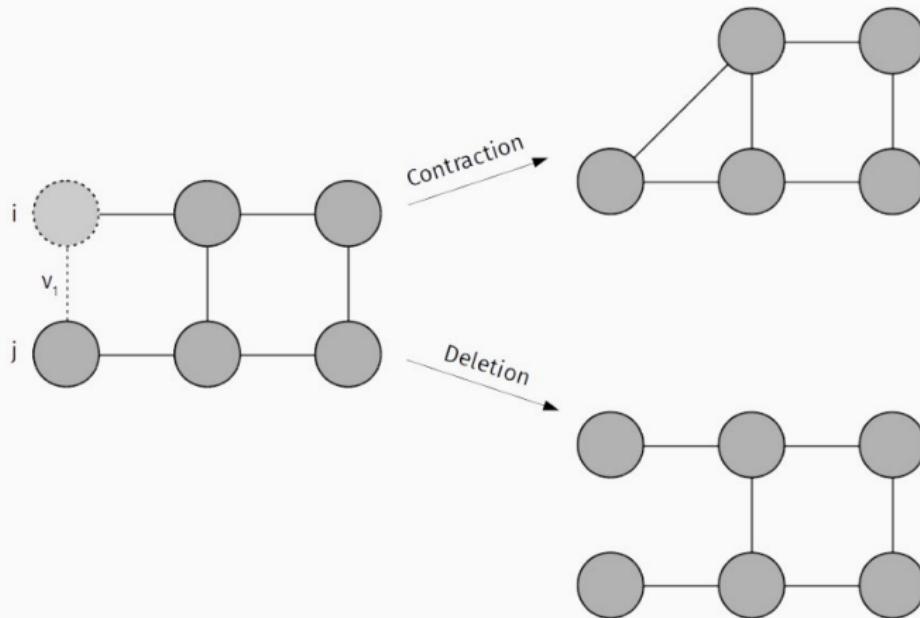


Figure 1: Contraction and deletion on a graph.

The Kasteleyn and Fortuin Theorem

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

$$Z = \sum_X e^{-\beta E(X)} \stackrel{(\text{??})}{=} \sum_X e^{-\beta J \sum_\nu \epsilon_\nu} = \sum_X \prod_\nu e^{-\beta J \epsilon_\nu}. \quad (1)$$

The Kasteleyn and Fortuin Theorem

We now consider a graph where bond ν_1 connects two nodes i and j with states σ_i and σ_j , respectively. If we would delete bond ν_1 , the partition function is

$$Z_D = \sum_X \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu}. \quad (2)$$

The Kasteleyn and Fortuin Theorem

We can thus rewrite Eq. (1) as

$$Z = \sum_X e^{-\beta J \epsilon_{\nu_1}} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} = \sum_{X: \sigma_i = \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} + e^{-\beta J} \sum_{X: \sigma_i \neq \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu}$$

where the first part is the partition function of the contracted graph Z_C and the second part is given by the identity

$$\sum_{X: \sigma_i \neq \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} = \sum_X \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} - \sum_{X: \sigma_i = \sigma_j} \prod_{\nu \neq \nu_1} e^{-\beta J \epsilon_\nu} = Z_D - Z_C. \quad (3)$$

The Kasteleyn and Fortuin Theorem

Summarizing the latter results, we find

$$Z = Z_C + e^{-\beta J} (Z_D - Z_C) = pZ_C + (1-p)Z_D, \quad (4)$$

where $p = 1 - e^{-\beta J}$. To be more precise, we expressed the partition function Z as the contracted and deleted partition functions at bond ν_1 . We apply the latter procedure to another bond ν_2 and find

$$Z = p^2 Z_{C_{\nu_1}, C_{\nu_2}} + p(1-p) Z_{C_{\nu_1}, D_{\nu_2}} + (1-p)^2 Z_{D_{\nu_1}, D_{\nu_2}}. \quad (5)$$

The Kasteleyn and Fortuin Theorem

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

$$Z = \sum_{\substack{\text{configurations of} \\ \text{bond percolation}}} q^{\# \text{ of clusters}} p^c (1-p)^d = \left\langle q^{\# \text{ of clusters}} \right\rangle_b, \quad (6)$$

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 percolation¹.

¹In 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.,

$$p(C, \sigma_0) = p^{c_C} (1-p)^{d_C} \sum_{\substack{\text{bond percolation} \\ \text{without cluster } C}} q^{\# \text{ of clusters}} p^c (1-p)^d. \quad (7)$$

Coniglio-Klein clusters

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

$$p(C, \sigma_1)W[(C, \sigma_1) \rightarrow (C, \sigma_2)] = p(C, \sigma_2)W[(C, \sigma_2) \rightarrow (C, \sigma_1)] \quad (8)$$

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

Coniglio-Klein clusters

We then obtain for acceptance probabilities

$$W[(C, \sigma_2) \rightarrow (C, \sigma_1)] = \frac{p(C, \sigma_2)}{p(C, \sigma_1) + p(C, \sigma_2)} = \frac{1}{2} \quad \text{Glauber dyn. (9)}$$

$$W[(C, \sigma_2) \rightarrow (C, \sigma_1)] = \min \left[1, \frac{p(C, \sigma_2)}{p(C, \sigma_1)} \right] = 1 \quad \text{Metropolis (10)}$$

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.

Swendsen-Wang algorithm

Swendsen-Wang algorithm

- Occupy the bonds with probability $p = 1 - e^{-\beta J}$ if sites are in the same state.
- Identify the clusters with the Hoshen-Kopelman algorithm.
- Flip the clusters with probability 0.5 for Ising or always choose a new state for $q > 2$.
- Repeat the procedure.

Hoshen-Kopelman algorithm

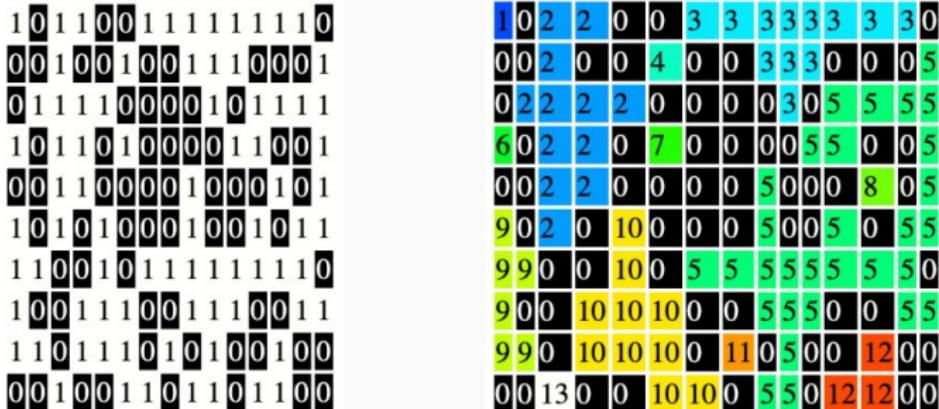


Figure 2: Left: spin configuration, 1's represent spin up; 0's are spin down. Right: the result of applying the HK algorithm to the grid on the left. Credit: <https://www.ocf.berkeley.edu/~fricke/projects/hoshenkopelman/hoshenkopelman.html>

Swendsen-Wang algorithm

Swendsen-Wang algorithm

- Occupy the bonds with probability $p = 1 - e^{-\beta J}$ if sites are in the same state.
- Identify the clusters with the Hoshen-Kopelman algorithm.
- Flip the clusters with probability 0.5 for Ising or always choose a new state for $q > 2$.
- Repeat the procedure.

Wolff algorithm

Wolff algorithm

- Choose a site randomly.
- If the neighboring sites are in the same state, add them to the cluster with probability $p = 1 - e^{-\beta J}$.
- Repeat this for any site on the boundaries of the cluster, until all the bonds of the cluster have been checked exactly once.
- Choose a new state for the cluster.
- Repeat the procedure.

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 $|\vec{S}_i| = 1$.

Other Ising-like models

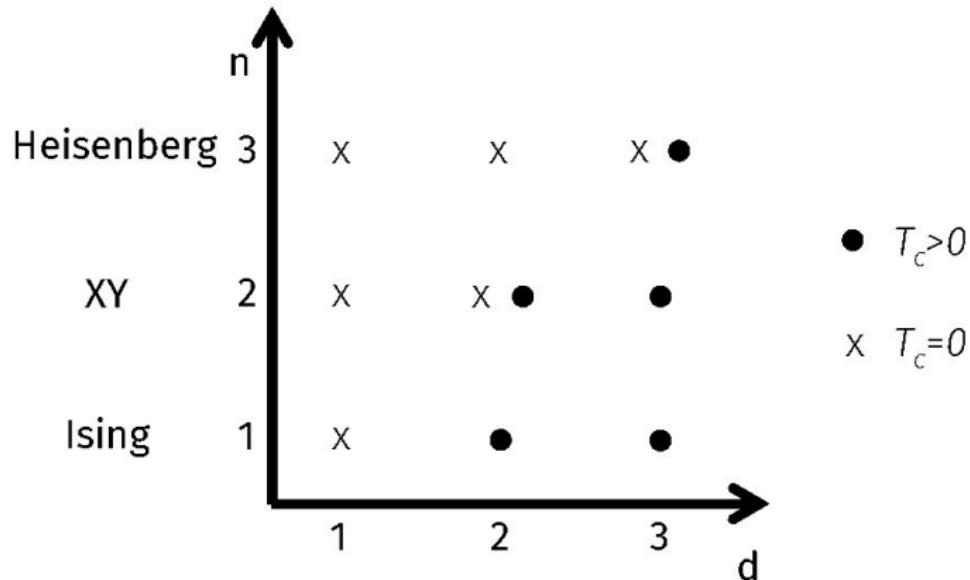


Figure 3: The dependence of the critical temperature on the number of vector components n .

Other Ising-like models

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

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 $e^{\frac{E}{T} - \frac{E}{T_0}}$.

Histogram methods

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 .

Histogram methods

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

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

Histogram methods

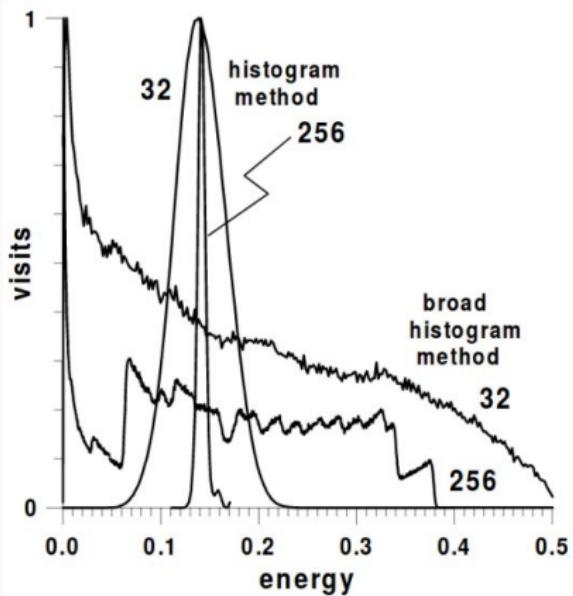


Figure 4: An example of the histogram and the broad histogram method for different system sizes. The figure is taken from [Oliveria et al. (1996)]

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.,

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

Broad histogram methods

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 $\frac{N_{\text{down}}(E+\Delta E)}{N_{\text{up}}(E)}$.

Broad histogram methods

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

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

Broad histogram methods

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 $g(E)$.

Broad histogram methods

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.

Flat histogram methods

Flat histogram method

- Start with $g(E) = 1$ and set $f = e$.
- Make a Monte Carlo update with $p(E) = 1/g(E)$.
- If the attempt is successful at E : $g(E) \leftarrow f \cdot g(E)$.
- Obtain a histogram of energies $H(E)$.
- If $H(E)$ is flat enough, then $f \leftarrow \sqrt{f}$.
- Stop when $f \leq 10^{-8}$.

Umbrella sampling

The Umbrella sampling technique was developed and proposed in [Torrie and Valleau, J. Comp. Phys., 1979]. The aim is to overcome the problem of the lacking ergodicity for certain energy landscapes.

As an example, in the Ising model the system could have difficulties in jumping from a positive to a negative magnetization or vice versa if the system is very large.

Umbrella sampling

The basic idea is to multiply transition probabilities with a function that is large at the free energy barrier and later remove this correction in the averaging step.

$$\tilde{p}(C) = \frac{w(C) e^{-\frac{E(C)}{k_B T}}}{\sum_C w(C) e^{-\frac{E(C)}{k_B T}}} \quad \text{with} \quad \langle A \rangle = \frac{\langle A/w \rangle_w}{\langle 1/w \rangle_w}. \quad (21)$$

Summary of the Histogram Methods

Summarizing, some of the most common techniques related to the histogram methods are

- Wang-Landau method [Wang 2001, Zhou 2006]
- Multiple histogram method [Ferrenberg 1989]
- Multicanonical Monte Carlo [Berg 2000]
- Flat Histogram method [Wang 2001]
- Umbrella sampling [Torrie and Valleau, 1979]

Computational Statistical Physics

Part I: Statistical Physics and Phase Transitions

Marina Marinkovic

March 23, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Renormalization group

Self-similarity

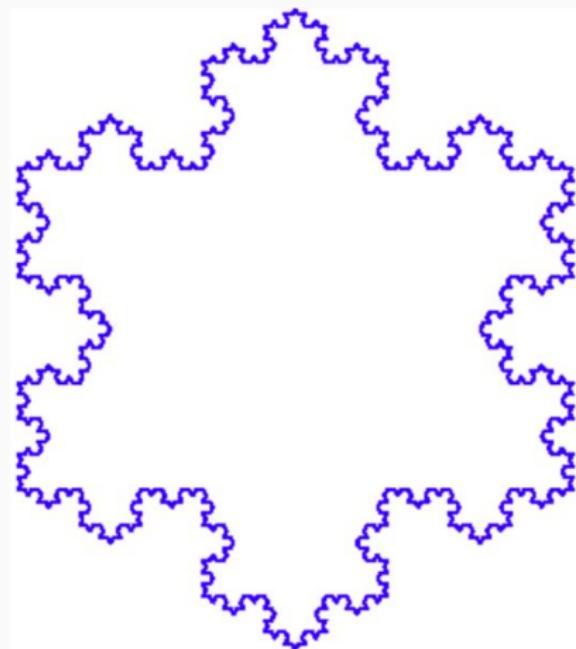


Figure 1: The Koch snowflake as an example of a self-similar pattern.

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

$$Z = \sum_{\{\sigma\}} e^{-\beta H} \tag{1}$$

and the corresponding free energy invariant.

Renormalization and free energy

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

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

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

¹Extensive quantities such as volume or the total mass of a gas are proportional to the system size. Intensive quantities are not dependent on the system size, e.g., the energy density or the temperature.

Renormalization and 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

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

Renormalization and free energy

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

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

Renormalization and free energy

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.

Majority rule

A straightforward example which can be regarded as renormalization of spin systems is the *majority rule*. Instead of considering all spins in a certain neighborhood separately, one just takes the direction of the net magnetization of these regions as new spin value, i.e.,

$$\tilde{\sigma}_i = \text{sign} \left(\sum_{\text{region}} \sigma_i \right). \quad (8)$$

Majority rule

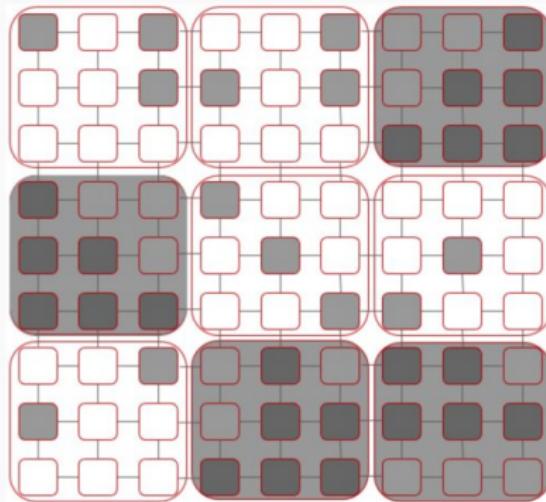


Figure 2: An illustration of the majority rule renormalization.

Decimation of the one-dimensional Ising model

Another possible rule is *decimation* which eliminates certain spins, generally in a regular pattern.

The spins only interact with their nearest neighbors and the coupling constant $K = J/(k_B T)$ is the same for all spins.

Decimation of the one-dimensional Ising model



Figure 3: An example of a one-dimensional Ising chain.

Decimation of the one-dimensional Ising model

To further analyze this system, we compute its partition function Z and obtain

$$\begin{aligned} Z &= \sum_{\{\sigma\}} e^{K \sum_i \sigma_i} = \sum_{\sigma_{2i}=\pm 1} \prod_{2i} \left[\sum_{\sigma_{2i+1}=\pm 1} \prod_{2i+1} e^{K(\sigma_{2i}\sigma_{2i+1} + \sigma_{2i+1}\sigma_{2i+2})} \right] \\ &= \sum_{\sigma_{2i}=\pm 1} \prod_{2i} \{2\cosh[K(\sigma_{2i} + \sigma_{2i+2})]\} \\ &= \sum_{\sigma_{2i}=\pm 1} \prod_{2i} z(K) e^{K' \sigma_{2i} \sigma_{2i+2}} \\ &= [z(K)]^{\frac{N}{2}} \sum_{\sigma_{2i}=\pm 1} \prod_{2i} e^{K' \sigma_{2i} \sigma_{2i+2}}, \end{aligned} \tag{9}$$

where we used in the third step that the $\cosh(\cdot)$ function only depends on even spins.

Decimation of the one-dimensional Ising model

According to Eq. (9), the relation

$$Z(K, N) = [z(K)]^{\frac{N}{2}} Z(K', N/2) \quad (10)$$

holds as a consequence of the decimation method. The function $z(K)$ is the spin-independent part of the partition function and K' is the renormalized coupling constant.

Decimation of the one-dimensional Ising model

We compute the relation

$z(K) e^{K' s_{2i} s_{2i+2}} = 2 \cosh [K (s_{2i} + s_{2i+2})]$ explicitly and find

$$z(K) e^{K' s_{2i} s_{2i+2}} = \begin{cases} 2 \cosh(2K) & \text{if } s_{2i} = s_{2i+2}, \\ 2 & \text{otherwise.} \end{cases} \quad (11)$$

Decimation of the one-dimensional Ising model

Dividing and multiplying the latter two expressions yields

$$e^{2K'} = \cosh(2K) \quad \text{and} \quad z^2(K) = 4 \cosh(2K). \quad (12)$$

And the renormalized coupling constant K' in terms of K is given by

$$K' = \frac{1}{2} \ln [\cosh(2K)]. \quad (13)$$

Decimation of the one-dimensional Ising model

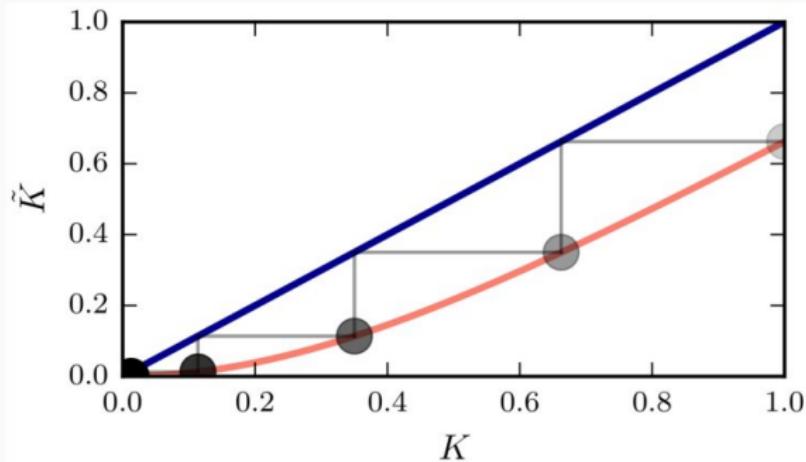


Figure 4: An illustration of the fixed point iteration defined by Eq. 13.

Decimation of the one-dimensional Ising model

Given the partition function, we now compute the free energy according to $F = -k_B T N f(K) = -k_B T \ln(Z)$, with $f(K)$ being the free energy density. Taking the logarithm of Eq. 10, we get:

$$\ln[Z(K, N)] = N f(K) = \frac{1}{2} N \ln[z(K)] + \frac{1}{2} N f(\tilde{K}) \quad (14)$$

Based on the previous equation, we can derive the following recursive relation for the free energy density

$$f(\tilde{K}) = 2f(K) - \ln[2 \cosh (2K)^{\frac{1}{2}}] \quad (15)$$

Decimation of the one-dimensional Ising model

There exists one *stable fixed point* at $K^* = 0$ and another *unstable one* at $K^* \rightarrow \infty$. Every fixed point ($K^* = \tilde{K}$) implies that Eq. 15 can be rewritten due to $f(\tilde{K}) = f(K^*)$.

The case of $K^* = 0$ corresponds to the high-temperature limit where the free energy approaches the value

$$F = -k_B T N f(K^*) = -k_B T N \ln(2). \quad (16)$$

In this case, the entropy dominates the free energy.

Decimation of the one-dimensional Ising model

For $K^* \rightarrow \infty$, the system approaches the low temperature limit and the free energy is given by

$$F = -k_B T N f(K^*) = -k_B T N K = -N J, \quad (17)$$

i.e., the free energy of the system is given by its internal energy.

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

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

Generalization

At T_c there exists a fixed point $K_\alpha^* = \tilde{K}_\alpha(K_1^*, \dots, K_M^*)$. 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

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

Generalization

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 $\lambda_\alpha > 1$.

Generalization

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.

Monte Carlo renormalization group

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

$$\chi_{\alpha,\beta} = \frac{\partial \langle O_\alpha \rangle}{\partial K_\beta} = \langle O_\alpha O_\beta \rangle - \langle O_\alpha \rangle \langle O_\beta \rangle,$$

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

Monte Carlo renormalization group

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

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

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.

Monte Carlo renormalization group

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^* .

Computational Statistical Physics

Part I: Statistical Physics and Phase Transitions

Marina Marinkovic

March 30, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Machine Learning Classification

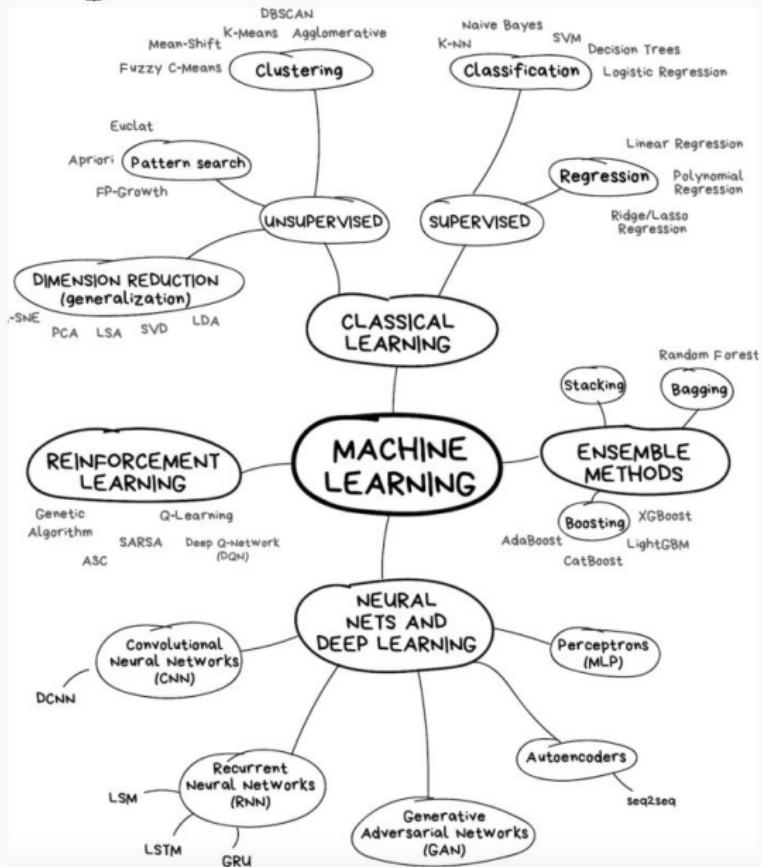


Figure 1: [Credit: https://vas3k.com/blog/machine_learning/]

Boltzmann machine

Hopfield Network

We begin our excursion to machine learning with a network consisting of neurons which are fully connected, i.e., every single neuron is connected to all other neurons. A neuron represents a node of network and is nothing but a function of I different inputs $\{x_i\}_{i \in \{1, \dots, I\}}$ which are weighted by $\{w_i\}_{i \in \{1, \dots, I\}}$ to compute and output y .

Hopfield Network

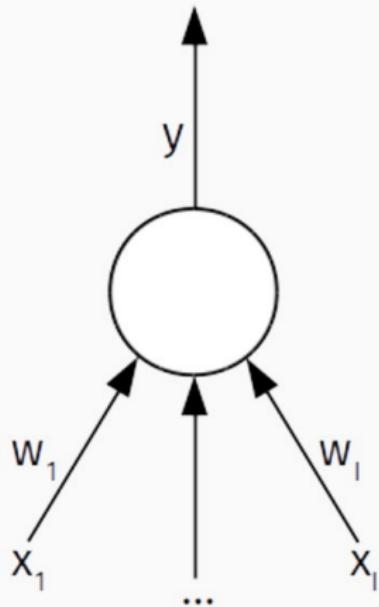


Figure 2: An illustration of a single neuron with output y , inputs $\{x_i\}_{i \in \{1, \dots, I\}}$ and weights $\{w_i\}_{i \in \{1, \dots, I\}}$.

Hopfield Network

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 $w_{ii} = 0$.

Hopfield Network

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.

Hopfield Network

The dynamics of a Hopfield network is

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

Hopfield Network

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



Figure 3: A comparison of denoising capabilities of Hopfield and Boltzmann machine models. The figure is taken from <https://arxiv.org/pdf/1803.08823.pdf>.

Boltzmann machine learning

For some applications finding a local minimum based on the deterministic update rule defined by Eq. (3) might not be sufficient. Similar to the discussion of Monte Carlo methods for Ising systems, we employ an update probability

$$p_i = \frac{1}{1 + \exp(-\Delta E_i/T)} = \sigma(2a_i/T) \quad (5)$$

to set neuron i to unity independent of its state [Ackley et al., 1985]. Here, $\sigma(x) = 1/[1 + e^{-x}]$ denotes the sigmoid function.

Boltzmann machine learning

As defined in Eq. (4), the energy difference ΔE_i is the gap between a configuration with an active neuron i and an inactive one. The parameter T acts as temperature equivalent¹.

A closer look at Eqs. (2) and (5) tells us that we are simulating a Hamiltonian system with Glauber dynamics. Due to the fulfilled detailed balance condition, we reach thermal equilibrium and find again for the probabilities of the system to be in state X or Y ²

$$\frac{p_{\text{eq}}(Y)}{p_{\text{eq}}(X)} = \exp\left(-\frac{E(Y) - E(X)}{T}\right). \quad (6)$$

¹For $T \rightarrow 0$, we recover deterministic dynamics as described by Eq. (3).

²Here set $k_B = 1$.

Boltzmann machine learning

We divide the Boltzmann machine units into visible and hidden units represented by the nonempty set V and possibly empty set H , respectively. The visible units are set by the environment whereas the hidden units are additional variables which might be necessary to model certain outputs.

Boltzmann machine learning

Let $P'(\nu)$ be the probability distribution over the visible units ν in a freely running network. It can be obtained by marginalizing over the corresponding joint probability distribution, i.e.,

$$P'(\nu) = \sum_h P(\nu, h) \tag{7}$$

where h represents a hidden unit.

Boltzmann machine learning

The goal is to come up with a method such that $P'(\nu)$ approaches the unknown environment distribution $P(\nu)$. We measure the difference between $P'(\nu)$ and $P(\nu)$ in terms of the Kullback-Leibler divergence (relative entropy)

$$G = \sum_{\nu \in V} P(\nu) \ln \left[\frac{P(\nu)}{P'(\nu)} \right]. \quad (8)$$

Boltzmann machine learning

To minimize G for the freely running network, we perform a gradient descent according to

$$\frac{\partial G}{\partial w_{ij}} = -\frac{1}{T} (p_{ij} - p'_{ij}), \quad (9)$$

where p_{ij} is the probability that two units are active on average if the environment is determining the states and p'_{ij} is the corresponding probability in a freely running network without coupling to the environment.

Boltzmann machine learning

Both probabilities are measured at thermal equilibrium. In the literature, the probabilities p_{ij} and p'_{ij} are also often defined in terms of the thermal averages $\langle x_i x_j \rangle_{\text{data}}$ and $\langle x_i x_j \rangle_{\text{model}}$ model, respectively. The weights w_{ij} of the network are then updated according to

$$\Delta w_{ij} = \epsilon (p_{ij} - p'_{ij}) = \epsilon (\langle x_i x_j \rangle_{\text{data}} - \langle x_i x_j \rangle_{\text{model}}), \quad (10)$$

where ϵ is the learning rate.

Boltzmann machine learning

The so-called restricted Boltzmann machine is not taking into account mutual connections within the set of hidden and visible units, and turned out to be more suitable for learning tasks. In the case of restricted Boltzmann machines, the weight update is given by

$$\Delta w_{ij} = \epsilon (\langle \nu_i h_j \rangle_{\text{data}} - \langle \nu_i h_j \rangle_{\text{model}}), \quad (11)$$

where ν_i and h_j represent visible and hidden units, respectively.

Boltzmann machine learning

Instead of sampling the configurations for computing $\langle \nu_i h_j \rangle_{\text{data}}$ and $\langle \nu_i h_j \rangle_{\text{model}}$ at thermal equilibrium, we could also just consider a few relaxation steps. This method is called contrastive divergence and defined by the following update rule

$$\Delta w_{ij}^{CD} = \epsilon (\langle \nu_i h_j \rangle_{\text{data}}^0 - \langle \nu_i h_j \rangle_{\text{model}}^k), \quad (12)$$

where the superscript indices indicate the number of updates.

Parallelization

Monte Carlo parallelization

Monte Carlo simulations on regular lattices are well-suited for parallelization. The following points summarize the basic ideas behind parallelization:

Domain decomposition

- Nearest-neighbor updates do not involve the whole system.
- Domain decomposition into sublattices is possible.
- The decomposed domains are distributed using MPI.
- Sublattices are extracted with logical masks.
- A periodic shift (CSHIFT) is used to obtain neighbors for periodic boundary conditions.

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

April 6, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Phase transitions

- 23.02. Introduction to statistical physics and the Ising model
- 02.03. Monte Carlo methods
- 09.03. Finite size methods and cluster algorithms
- 16.03. Histogram methods
- 23.03. Renormalization group
- 30.03. Boltzmann machines

Molecular dynamics

- 06.04. Molecular Dynamics, Verlet scheme, Leapfrog scheme
- 13.04. Optimization, Linked cell algorithm
- 20.04. ETH vacations
- 27.04. Lagrange multipliers, Rigid bodies, quaternions
- 04.05. Nosé-Hoover thermostat, stochastic method, constant pressure ensemble

Event-driven dynamics

- 11.05. Event driven, inelastic collisions, friction
- 18.05. Contact dynamics
- 25.05. *Non-equilibrium phase transitions / ab initio MD / Car-Parinello [TBC]*
- 01.06. Advanced topics

Molecular dynamics

Molecular dynamics

MD is used in a variety of fields, some examples are:

- Simulation of atoms and molecules,
- Gravitational interactions,
- Flow dynamics,
- Biopolymers,
- Granular materials,
- Dislocations, voids, quasi-particles,
- Electrons (Car-Parrinello),
- Explosions.

Molecular dynamics

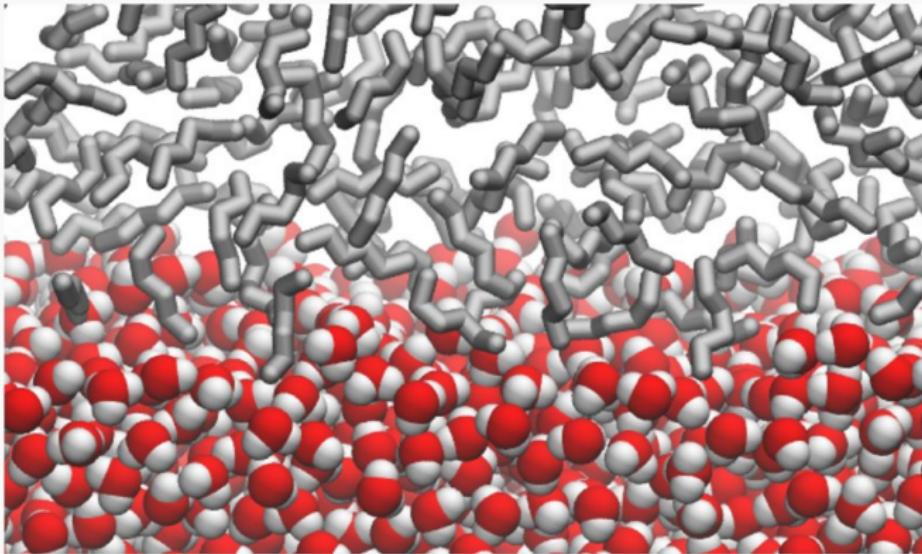


Figure 1: Water hexane interface. The figure is taken from
[https://commons.wikimedia.org/wiki/File:
Simulated_structure_of_the_water_hexane_interface.png](https://commons.wikimedia.org/wiki/File:Simulated_structure_of_the_water_hexane_interface.png)

Molecular dynamics

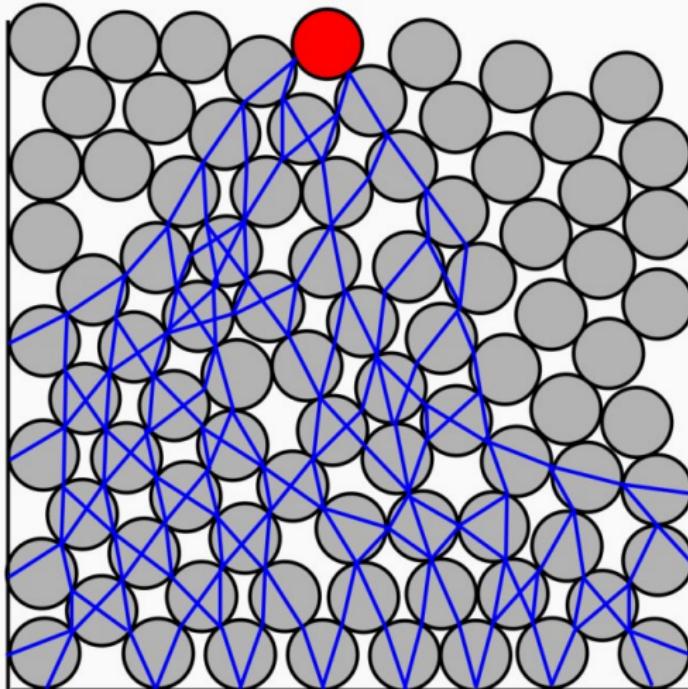


Figure 2: Granular matter. The figure is taken from https://commons.wikimedia.org/wiki/File:Stress_transmision.svg

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.



Molecular dynamics

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.



Molecular dynamics

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:

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

Molecular dynamics

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.

Molecular dynamics

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

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

Molecular dynamics

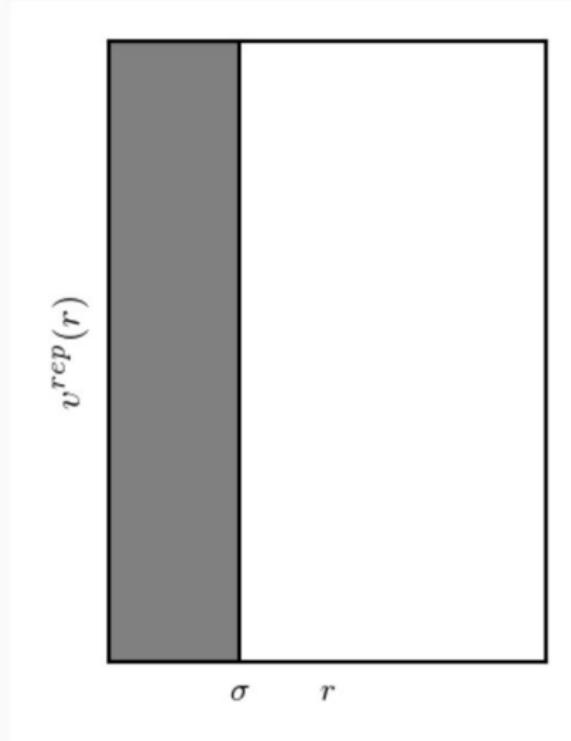


Figure 3: An example of a hard sphere potential.

Equations 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.

Equations of motion

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.

Equations of motion

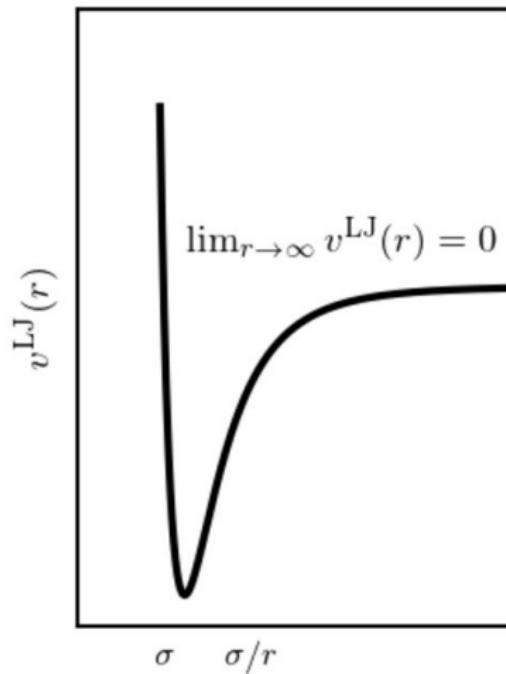


Figure 4: An example of a Lennard-Jones potential,
cf. <http://www.atomsinmotion.com/>.

Equations of motion

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 $i \in \{1, \dots, N\}$.

Equations of motion

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{v}_i . Due to $\dot{x}_i = v_i = \frac{p_i}{m}$ and $\dot{p}_i = -\nabla V(Q) = f_i$, the equations of motion are:

$$m_i \ddot{x}_i = f_i = \sum_j f_{ij}, \quad (10)$$

where f_{ij} is the force exerted by particle j on particle i .

Contact time

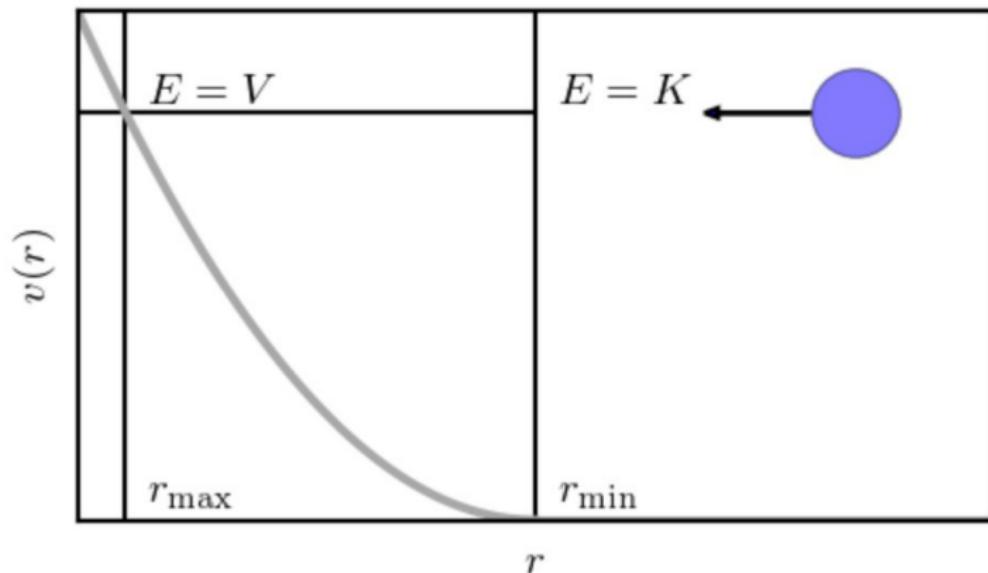


Figure 5: Derivation of the contact time.

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.

Contact time

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 [Verlet '67]

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

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

Adding the latter two expressions yields

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

Verlet method

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).

Verlet method

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:

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

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

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

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

Leapfrog Method

The analogies and differences between the Leapfrog method

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

and the forward Euler integration

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

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.

Leapfrog Method

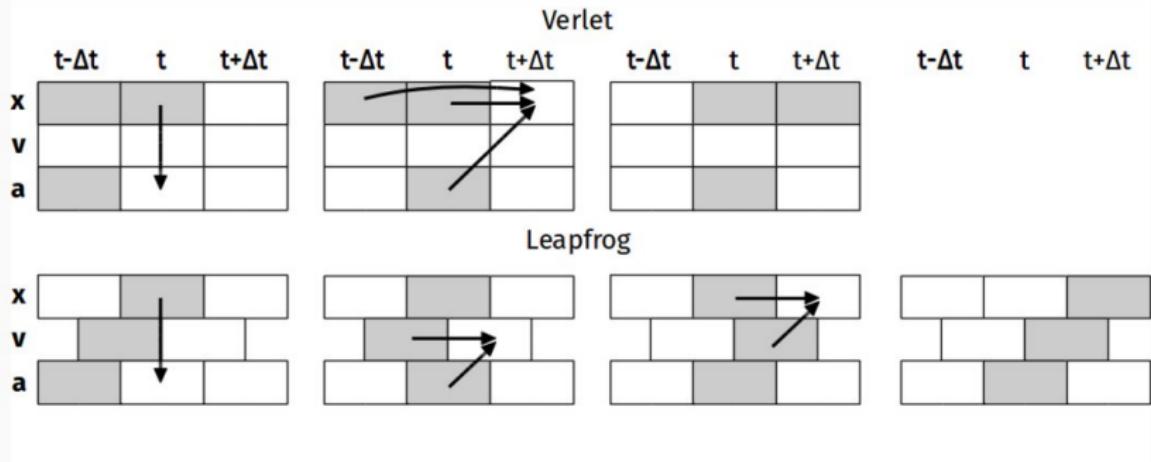


Figure 6: A comparison between Verlet and Leapfrog update schemes.

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

April 13, 2022 - updated

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Molecular dynamics

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.



Molecular dynamics

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.



Molecular dynamics

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:

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

Molecular dynamics

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.

Molecular dynamics

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

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

Molecular dynamics

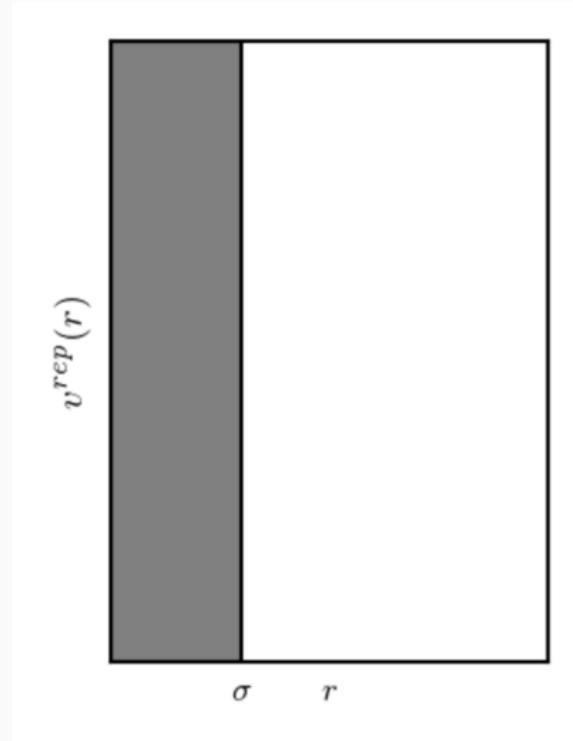


Figure 1: An example of a hard sphere potential.

Equations 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.

Equations of motion

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.

Equations of motion

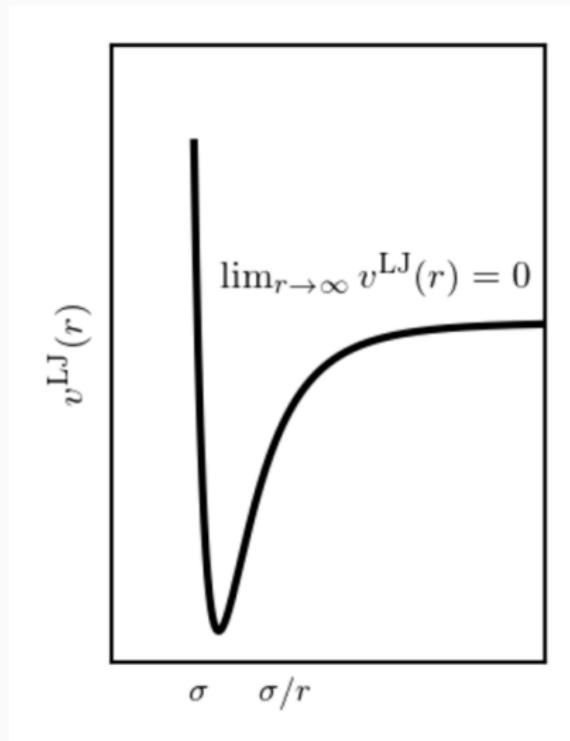


Figure 2: An example of a Lennard-Jones potential,
cf. <http://www.atomsinmotion.com/>.

Equations of motion

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 $i \in \{1, \dots, N\}$.

Equations of motion

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{v}_i . Due to $\dot{x}_i = v_i = \frac{p_i}{m}$ and $\dot{p}_i = -\nabla V(Q) = f_i$, the equations of motion are:

$$m_i \ddot{x}_i = f_i = \sum_j f_{ij}, \quad (10)$$

where f_{ij} is the force exerted by particle j on particle i .

Contact time

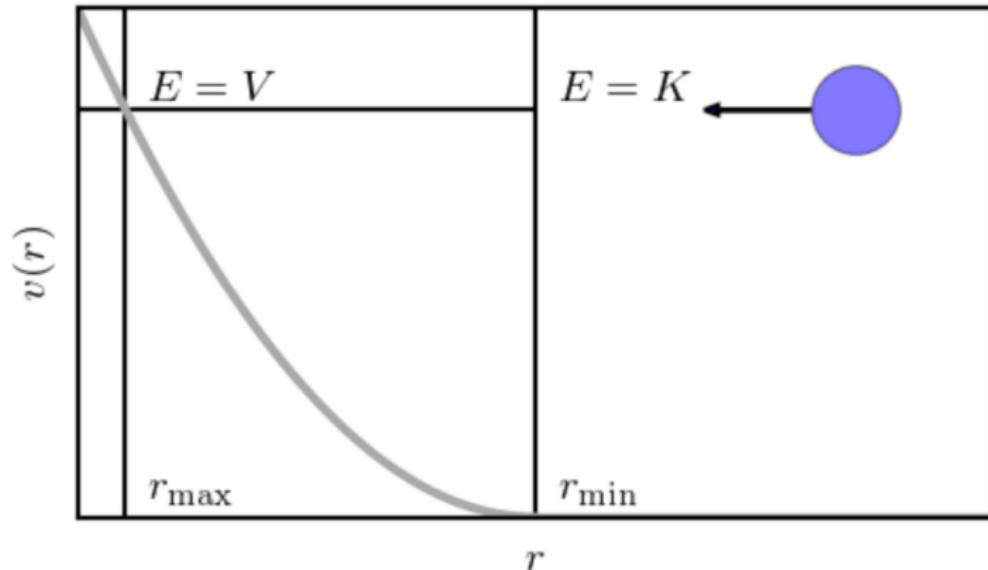


Figure 3: Derivation of the contact time.

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.

Contact time

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 [Verlet '67]

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

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

Adding the latter two expressions yields

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

Verlet method

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).

Verlet method

Some general remarks about the Verlet method:

- Two time steps need to be stored (t and $t - \Delta t$).
- Velocities can be computed with $v(t) = \frac{x(t+\Delta t) - x(t-\Delta t)}{2\Delta t}$.
- 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:

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

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

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

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

Leapfrog Method

The analogies and differences between the Leapfrog method

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

and the forward Euler integration

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

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.

Leapfrog Method

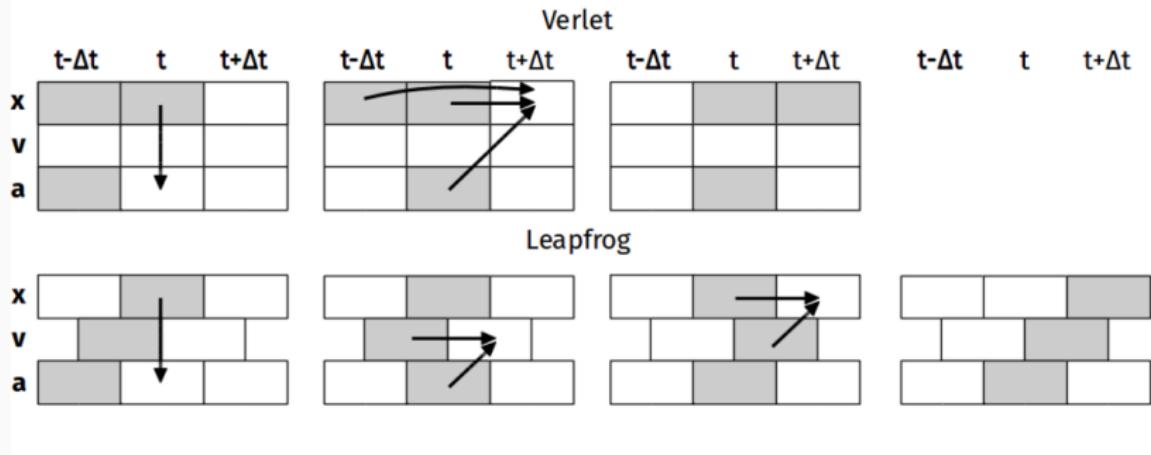


Figure 4: A comparison between Verlet and Leapfrog update schemes.

Optimization

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 $\mathbf{f}_i = f(r^{-2(n-1)}) \mathbf{r}_i$.

Optimization

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 $S = K/r_c^2$.

Optimization

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.

Verlet tables

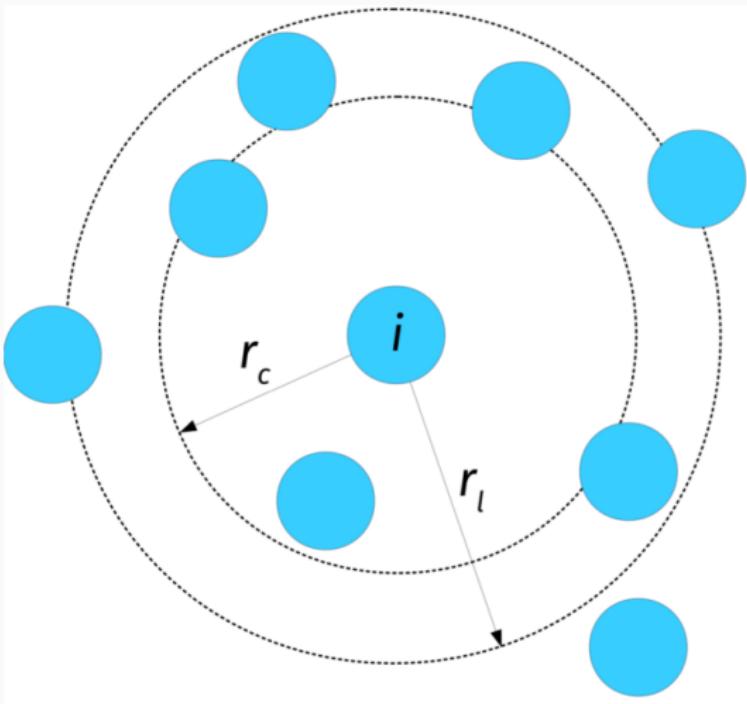


Figure 5: An illustration of the Verlet table method. Only particles within a distance of $r_l > r_c$ from particle i are considered

Verlet tables

In order to reduce the amount of necessary computations, we wish to omit the computation of particle-particle interactions when they are negligible. Therefore, we only consider particles in a certain range $r_l > r_c$.

For every particle, we then store the coordinates of the neighbouring particles in a list which is referred to as *Verlet table*. As the particles move with time, the table has to be updated after

$$n = \frac{r_l - r_c}{\Delta t v_{\max}} \quad (27)$$

time steps, where v_{\max} denotes the maximal velocity. Updating the whole list is still an operation of complexity $\mathcal{O}(N^2)$.

Linked-Cell Method

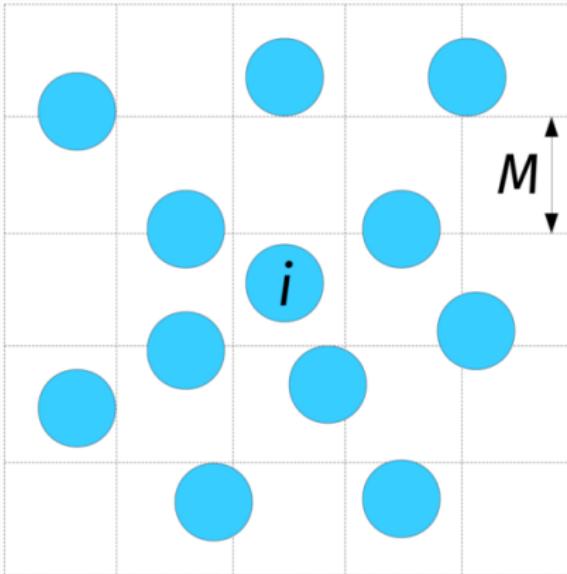


Figure 6: An illustration of the linked-cell method. A grid with grid spacing $M(\frac{r_c}{2} < M < r_c)$ is placed on top of the MD simulation geometry. Only interactions between particles in a certain cell neighborhood have to be considered [D. Knuth, The Art of Computer Programming, '68].

Linked-Cell Method

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 $\text{FIRST}[j]$. If cell j is empty, then $\text{FIRST}[j] = 0$. 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 $\text{LIST}[i] = 0$.

Linked-Cell Method

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]];  
}
```

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).

Dynamics of composed particles

Dynamics of composed particles

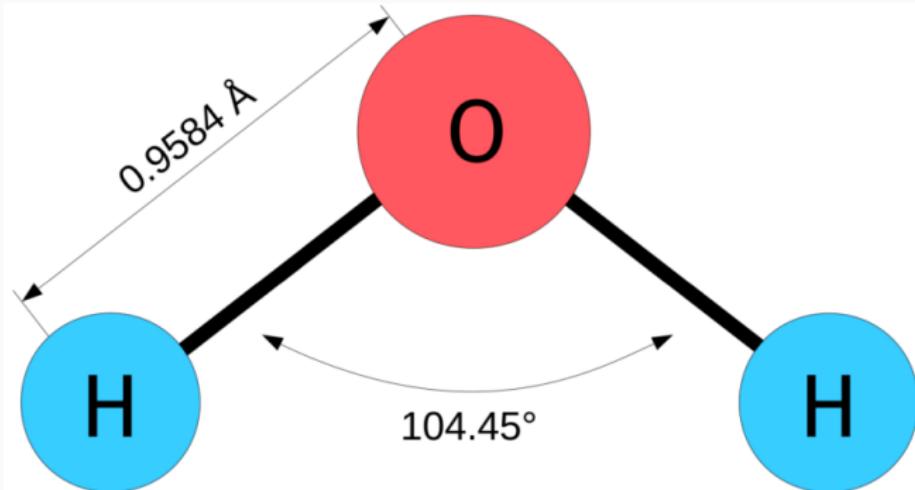


Figure 7: A water molecule as a composed particle system consisting of two hydrogen and one oxygen atom.

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.

Lagrange multipliers

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 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

$$\chi_{12} = r_{12}^2 - d_{12}^2, \tag{29}$$

$$\chi_{23} = r_{23}^2 - d_{23}^2, \tag{30}$$

are zero if the particles have the desired distance.

Lagrange multipliers

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

Lagrange multipliers

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

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

Lagrange multipliers

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

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

Lagrange multipliers

Then we correct the value using the constraints according to

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

Lagrange multipliers

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

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

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

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

Lagrange multipliers

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 .

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

April 27, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Dynamics of composed particles

Dynamics of composed particles

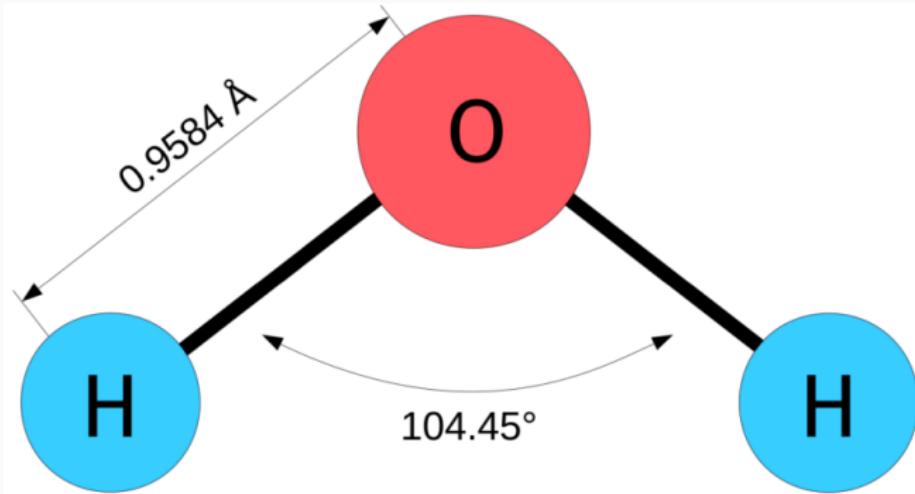


Figure 1: A water molecule as a composed particle system consisting of two hydrogen and one oxygen atom.

Rigid bodies

Rigid bodies

Systems whose n constituents of mass m_i are located at fixed positions \mathbf{x}_i are referred to as rigid bodies. The motion of such objects is described by translations of the center of mass and rotations around it. The center of mass is defined as

$$\mathbf{x}_{\text{cm}} = \frac{1}{M} \sum_{i=1}^n \mathbf{x}_i m_i \quad \text{with} \quad M = \sum_{i=1}^n m_i. \quad (1)$$

Rigid bodies

The equation of motion of the center of mass and the corresponding torque are given by

$$M\ddot{\mathbf{x}}_{cm} = \sum_{i=1}^n \mathbf{f}_i = \mathbf{f}_{cm} \quad \text{and} \quad \mathbf{M} = \sum_{i=1}^n \mathbf{d}_i \wedge \mathbf{f}_i, \quad (2)$$

where $\mathbf{d}_i = \mathbf{x}_i - \mathbf{x}_{cm}$. In two dimensions, the rotation axis always points in the direction of the normal vector of the plane.

Therefore, there exist only three degrees of freedom: two translational and one rotational. In three dimensions, there are six degrees of freedom: three translational and three rotational.

Rigid Bodies 2D

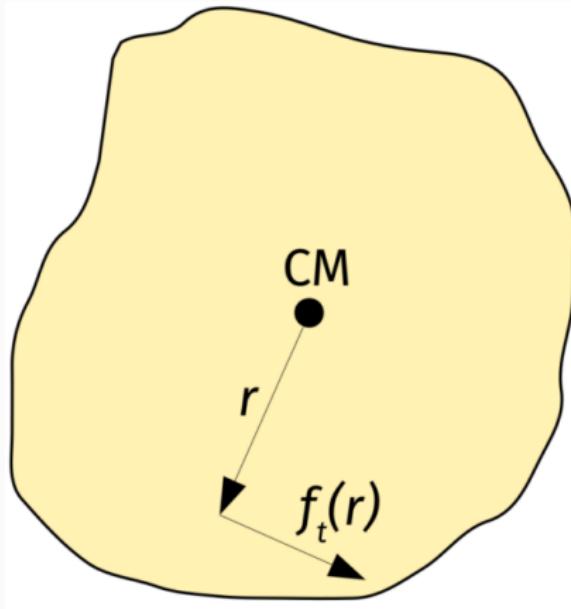


Figure 2: An example of a rigid body in two dimensions. The black dot shows the center of mass (CM), and $f_t(r)$ represents the tangential force component.

Rigid Bodies 2D

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

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

Rigid Bodies 2D

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

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

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

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

Rigid Bodies 3D

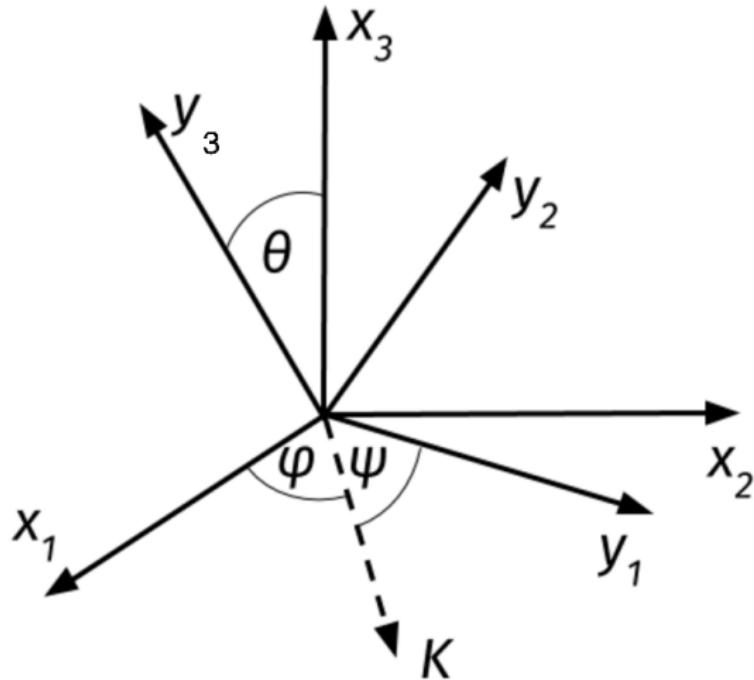


Figure 3: Euler angle parameterization of a rotation matrix.

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 $R^T R = R R^T = 1$.

Rigid Bodies 3D

Furthermore, we define $\Omega = R^T \dot{R}$ 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 $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$.

Rigid Bodies 3D

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

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

Rigid Bodies 3D

The components of the inertia tensor are defined as

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

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.

Rigid Bodies 3D

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

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

Rigid Bodies 3D

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.,

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

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

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

Rigid Bodies 3D

The angular velocities are then integrated according to

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

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

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

Rigid Bodies 3D

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

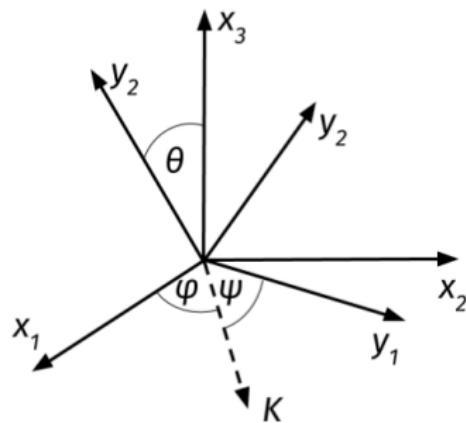


Figure 4: Euler angle parameterization of a rotation matrix.

Euler angles

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

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

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

Euler angles

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

Denis J. Evans, a professor in Canberra, AU, came up with a trick to optimize the computation of rotational velocities [Evans,'77] and replace Euler angles in EoM with Quaternions.

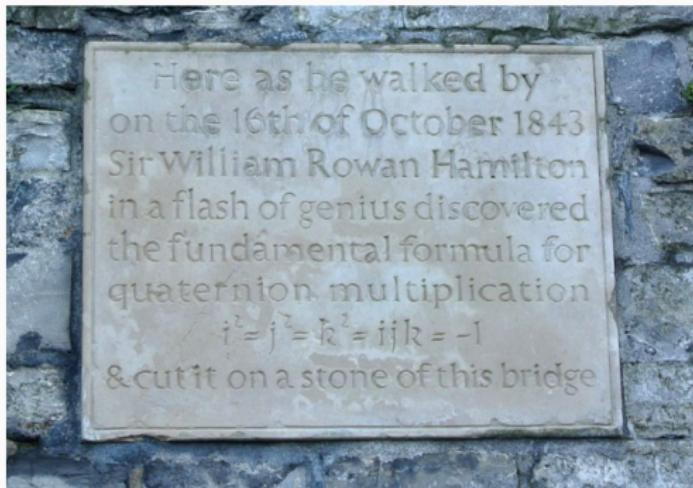


Figure 5: Stone at Brougham Bridge in Dublin where Hamilton came up with the multiplication rule for quaternion. Source:
<https://commons.wikimedia.org/>.

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.

Quaternions

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

$$R = \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix}. \quad (28)$$

Quaternions

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

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

Quaternions

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

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

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

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

Quaternions

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).

Quaternions

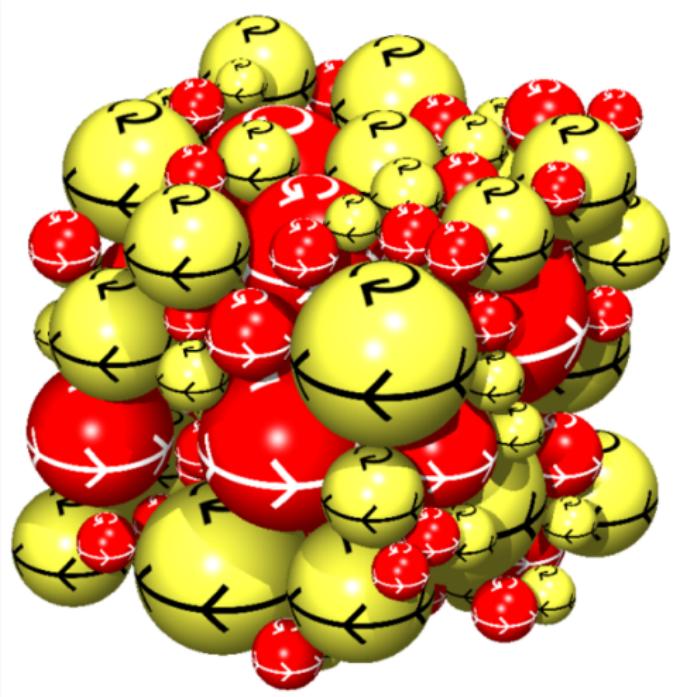


Figure 6: Rotating spheres in a sphere assembly as an example of rigid body dynamics [Stager et al., 2016]

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

May 3, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Canonical ensemble

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).

Canonical Ensemble

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} .

Canonical Ensemble

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 .

Velocity rescaling

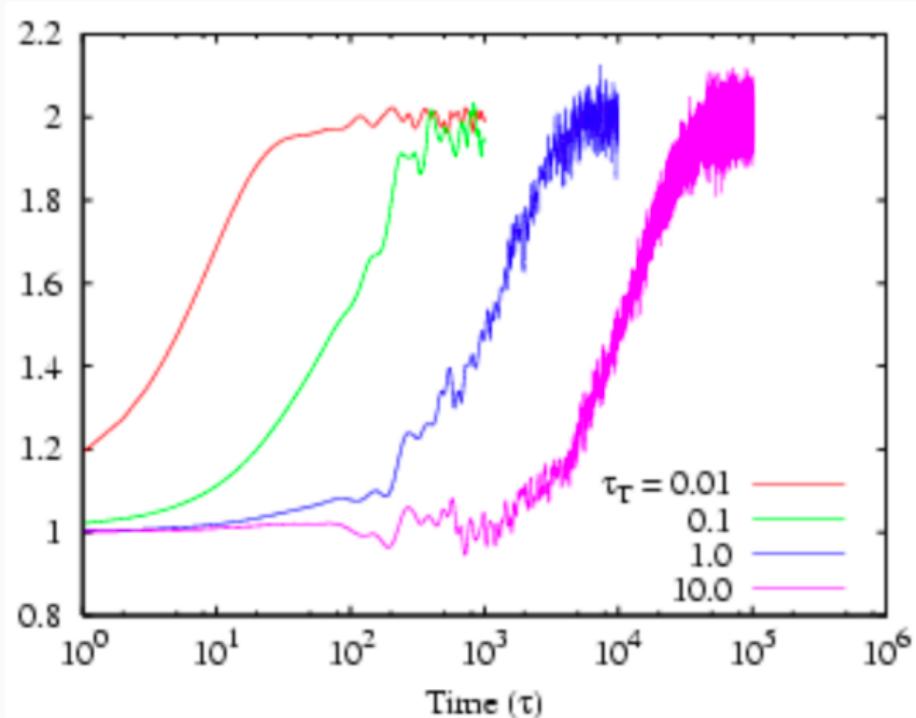
This method is very easy to implement. However, the problem is that we change the physics and in particular the time and the resulting velocity distribution deviates from the canonical one.

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.

Velocity rescaling



Constraint method

Another possibility to adjust the system's temperature is to add a friction term to the equation of motion, i.e.,

$$\dot{\mathbf{p}}_i = \mathbf{f}_i - \xi \mathbf{p}_i, \quad (8)$$

where $\mathbf{p}_i = m_i \dot{\mathbf{x}}_i$. Various definitions of the friction coefficient ξ are possible. Hoover's original proposal is based on the following constant temperature condition:

$$\dot{\mathcal{T}} \sim \frac{d}{dt} \left(\sum_{i=1}^N \mathbf{p}_i^2 \right) \sim \sum_{i=1}^N \dot{\mathbf{p}}_i \mathbf{p}_i = 0. \quad (9)$$

Constraint method

By combining Eqs. (52) and (53), we find

$$\xi = \frac{\sum_{i=1}^N \mathbf{f}_i \mathbf{p}_i}{\sum_{i=1}^N |\mathbf{p}_i|^2}. \quad (10)$$

This method makes it necessary to already start at the desired temperature. Another possibility is to determine the friction coefficient according to (Berendsen)

$$\xi = \gamma \left(1 - \frac{T}{\bar{T}} \right). \quad (11)$$

or (Hoover)

$$\dot{\xi} = \frac{fk_B}{Q} (\mathcal{T} - T). \quad (12)$$

The parameters γ and Q determine the temperature adaption rate, and f is the number of degrees of freedom.

Nosé-Hoover thermostat [S. Nosé '81, S. Nosé '84]

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

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

Nosé-Hoover thermostat

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:

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

Nosé-Hoover thermostat

The Hamiltonian is thus

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

with $p_s = Q\dot{s}$ being the momentum corresponding to s . The velocities are

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

Nosé-Hoover thermostat

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

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

Nosé-Hoover thermostat

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

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

and

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

Nosé-Hoover thermostat

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

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

and for the forces

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

Nosé-Hoover thermostat

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

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

Nosé-Hoover thermostat

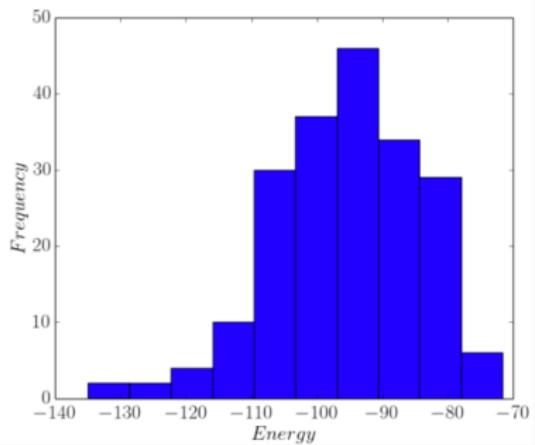
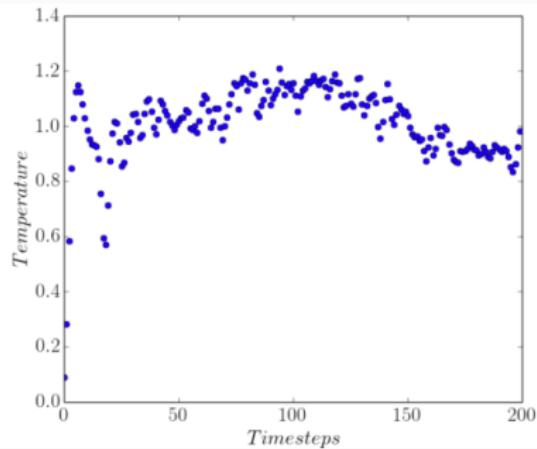
The first term in Eq. (70) denotes the **measured** kinetic energy whereas the second one corresponds to the **desired** kinetic energy. The quantity Q represents the coupling to the heat bath and the higher the value of Q , the stronger the system reacts to temperature fluctuations. For $Q \rightarrow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

Nosé-Hoover thermostat



Nosé-Hoover thermostat

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

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

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

Nosé-Hoover thermostat

Integrating Eq. (73) over s yields

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

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.

Stochastic method

This method, proposed by Andersen in 1980, is a combination of the velocity rescaling and Monte Carlo. At temperature T , one expects the velocity distribution to recover the Maxwell-Boltzmann distribution:

$$P(\mathbf{p}) = \frac{1}{(\pi k_B T)^{\frac{3}{2}}} e^{-\frac{\mathbf{p}^2}{k_B T}}. \quad (31)$$

Every n time steps, the simulation will be suspended and particles are selected uniformly at random and given a new momentum according to (75). If n is too small, one has pure Monte Carlo and loses the real time scale, e.g., the long time tail of the velocity correlation. If n is too large the coupling to the heat bath is too weak, equilibration is slow and one will essentially work microcanonically.

Constant pressure

Another important situation is the one of constant pressure.

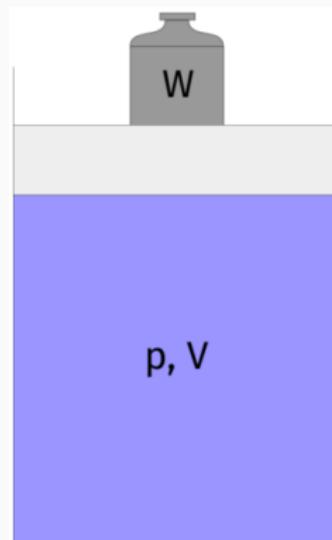


Figure 1: A weight of mass W exerts a pressure p on the system with volume V .

Constant pressure

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

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

Constant pressure

We now distinguish between particle-particle and particle-wall interactions, $\mathbf{f}_i^{\text{part}}$ and $\mathbf{f}_i^{\text{ext}}$, respectively. We then get:

$$\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot [\nabla_{\mathbf{x}_i} V(\mathbf{x})] \right\rangle = -\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot (\mathbf{f}_i^{\text{ext}} + \mathbf{f}_i^{\text{part}}) \right\rangle \quad (32)$$

$$= -\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot (\mathbf{f}_i^{\text{ext}}) \right\rangle - \underbrace{\frac{1}{3} \left\langle \sum_{i=1}^N \mathbf{x}_i \cdot (\mathbf{f}_i^{\text{part}}) \right\rangle}_{w \equiv \text{virial}} \quad (33)$$

Constant pressure

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

$$\mathcal{P}V \equiv Nk_B T + \langle w \rangle \quad (36)$$

Constant pressure

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.

Solving Nosé-Hoover – modified velocity Verlet

[Exercise 10] Coupling of the system of particles to a heat bath at fixed temperature T cannot be solved using neither (velocity) Verlet discussed in class, nor Leapfrog algorithm.

One way to solve the system of equations by introducing a small modification of the velocity Verlet algorithm.

Event-driven Molecular Dynamics

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.

Elastic Collisions

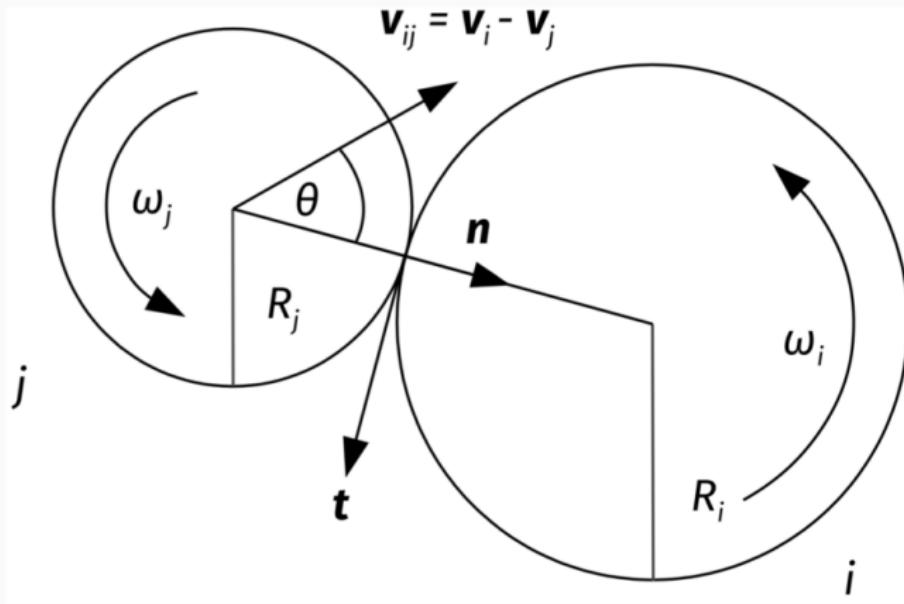


Figure 2: Two particles collide elastically.

Elastic Collisions

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

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

Elastic Collisions

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.

Lubachevsky method

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.

Lubachevsky method

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

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

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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.
- **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.

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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 perfect slip

We now approximate a collision by neglecting the tangential exchange of momentum – i.e. we assume a perfect slip. Only linear momentum and no angular momentum is exchanged. The conservation of momentum leads to

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} + \frac{\Delta \mathbf{p}}{m_i}, \quad (44)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} - \frac{\Delta \mathbf{p}}{m_j}, \quad (45)$$

and energy conservation:

$$\frac{1}{2}m_i (\mathbf{v}_i^{\text{before}})^2 + \frac{1}{2}m_j (\mathbf{v}_j^{\text{before}})^2 = \frac{1}{2}m_i (\mathbf{v}_i^{\text{after}})^2 + \frac{1}{2}m_j (\mathbf{v}_j^{\text{after}})^2. \quad (46)$$

Collision with perfect slip

The exchanged momentum is

$$\Delta \mathbf{p} = -2m_{\text{eff}} \left[(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}}) \cdot \mathbf{n} \right] \mathbf{n} \quad (47)$$

with $m_{\text{eff}} = \frac{m_i m_j}{m_i + m_j}$ being the *effective mass* and $\mathbf{n} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$. If $m_i = m_j$, the velocity updates are

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} - v_{ij}^n \cdot \mathbf{n}, \quad (48)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} + v_{ij}^n \cdot \mathbf{n}, \quad (49)$$

with $v_{ij}^n = (\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}}) \cdot \mathbf{n}$.

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.

Collision with rotation

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.

Collision with rotation

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

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

Collision with rotation

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

$$\begin{aligned} (1 - e_t) \mathbf{u}_{ij}^t &= \mathbf{u}_{ij}^t - \mathbf{u}_{ij}' \\ &= - [(\mathbf{v}'_i - \mathbf{v}_i - \mathbf{v}'_j + \mathbf{v}_j) - R (\boldsymbol{\omega}'_i - \boldsymbol{\omega}_i + \boldsymbol{\omega}'_j - \boldsymbol{\omega}_j) \wedge \mathbf{n}]. \end{aligned} \quad (57)$$

Collision with rotation

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

$$\mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} = (1 - e_t) \mathbf{u}_{ij}^t \quad (58)$$

$$= - \left[2 \left(\mathbf{v}_i^{t'} - \mathbf{v}_i^t \right) + 2q \left(\mathbf{v}_i^{t'} - \mathbf{v}_i^t \right) \right] \quad (59)$$

and finally

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

Collision with rotation

Analogously, we find for the remaining quantities

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

And the updated velocities are

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

Inelastic collisions

The kinetic energy of interacting and colliding particles is not constant due to friction, plastic deformation or thermal dissipation. We account for energy dissipation effects in an effective manner by introducing the *restitution coefficient*. The restitution coefficient is defined as the ratio of the energy before and after the interaction event, and it describes multiple physical effects, i.e.,

$$r = \frac{E^{\text{after}}}{E^{\text{before}}} = \left(\frac{v^{\text{after}}}{v^{\text{before}}} \right)^2, \quad (63)$$

where E^{after} and E^{before} are the energies before and after the interaction event, and v^{after} and v^{before} are the corresponding velocities.

Inelastic collisions

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

$$e_n = \sqrt{r_n} = \frac{v_n^{\text{after}}}{v_n^{\text{before}}}, \quad (64)$$

$$e_t = \sqrt{r_t} = \frac{v_t^{\text{after}}}{v_t^{\text{before}}}. \quad (65)$$

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

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

May 13, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Event-driven Molecular Dynamics

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.

Elastic Collisions

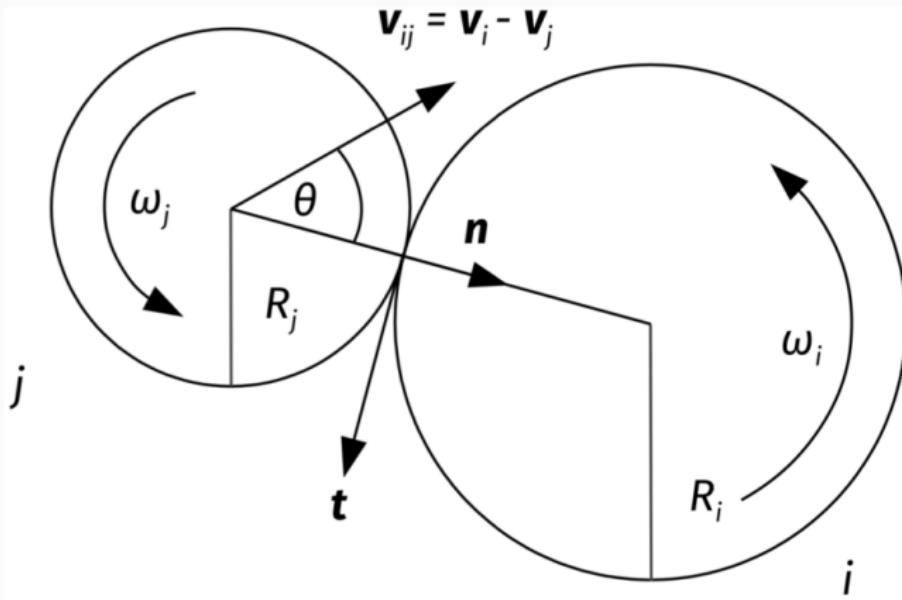


Figure 1: Two particles collide elastically.

Elastic Collisions

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

$$|\mathbf{r}_{ij}(t_{ij})| = |R_i + R_j| \quad (1)$$

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

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

Elastic Collisions

We should bear in mind that Eq. (2) 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.,

$$t_c = \min_{ij} (t_{ij}). \quad (3)$$

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

$$\mathbf{r}_i(t_0 + t_c) = \mathbf{r}_i(t_0) + \mathbf{v}_i(t_0)t_c \quad \text{and} \quad \phi_i(t_0 + t_c) = \phi_i(t_0) + \omega_i(t_0)t_c. \quad (4)$$

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.

Lubachevsky method

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.,

$$t^{(i)} = \min_j (t_{ij}). \quad (5)$$

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.

Lubachevsky method

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

$$t_c = \min_i \left(t^{(i)} \right). \quad (6)$$

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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.
- **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.

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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 perfect slip

We now approximate a collision by neglecting the tangential exchange of momentum – i.e. we assume a perfect slip. Only linear momentum and no angular momentum is exchanged. The conservation of momentum leads to

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} + \frac{\Delta \mathbf{p}}{m_i}, \quad (7)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} - \frac{\Delta \mathbf{p}}{m_j}, \quad (8)$$

and energy conservation:

$$\frac{1}{2}m_i (\mathbf{v}_i^{\text{before}})^2 + \frac{1}{2}m_j (\mathbf{v}_j^{\text{before}})^2 = \frac{1}{2}m_i (\mathbf{v}_i^{\text{after}})^2 + \frac{1}{2}m_j (\mathbf{v}_j^{\text{after}})^2. \quad (9)$$

Collision with perfect slip

The exchanged momentum is

$$\Delta \mathbf{p} = -2m_{\text{eff}} \left[(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}}) \cdot \mathbf{n} \right] \mathbf{n} \quad (10)$$

with $m_{\text{eff}} = \frac{m_i m_j}{m_i + m_j}$ being the *effective mass* and $\mathbf{n} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$. If $m_i = m_j$, the velocity updates are

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} - v_{ij}^n \cdot \mathbf{n}, \quad (11)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} + v_{ij}^n \cdot \mathbf{n}, \quad (12)$$

with $v_{ij}^n = (\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}}) \cdot \mathbf{n}$.

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

$$I \frac{d\omega_i}{dt} = \mathbf{r} \wedge \mathbf{f}_i, \quad (13)$$

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

$$\begin{aligned} I(\omega'_i - \omega_i) &= -Rm\mathbf{n} \wedge (\mathbf{v}'_i - \mathbf{v}_i), \\ I(\omega'_j - \omega_j) &= Rm\mathbf{n} \wedge (\mathbf{v}'_j - \mathbf{v}_j), \end{aligned} \quad (14)$$

with the primed velocities representing the ones after the collision.

Collision with rotation

Together with the conservation of momentum

$$\mathbf{v}'_i + \mathbf{v}'_j = \mathbf{v}_i + \mathbf{v}_j, \quad (15)$$

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

$$\omega'_i - \omega_i = \omega'_j - \omega_j = -\frac{Rm}{I} (\mathbf{v}'_i - \mathbf{v}_i) \wedge \mathbf{n}. \quad (16)$$

The relative velocity between particles i and j is

$$\mathbf{u}_{ij} = \mathbf{v}_i - \mathbf{v}_j - R(\omega_i + \omega_j) \wedge \mathbf{n}. \quad (17)$$

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

Collision with rotation

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

$$\begin{aligned}\mathbf{u}_{ij}^n &= (\mathbf{u}_{ij} \mathbf{n}) \mathbf{n}, \\ \mathbf{u}_{ij}^t &= \mathbf{u}_{ij} \wedge \mathbf{n} = [(\mathbf{v}_i - \mathbf{v}_j) - R(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j)] \wedge \mathbf{n}.\end{aligned}\tag{18}$$

Collision with rotation

General slips are described by

$$\mathbf{u}_{ij}^{t'} = e_t \mathbf{u}_{ij}^t, \quad (19)$$

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

$$\begin{aligned} (1 - e_t) \mathbf{u}_{ij}^t &= \mathbf{u}_{ij}^t - \mathbf{u}_{ij}' \\ &= - [(\mathbf{v}'_i - \mathbf{v}_i - \mathbf{v}'_j + \mathbf{v}_j) - R (\boldsymbol{\omega}'_i - \boldsymbol{\omega}_i + \boldsymbol{\omega}'_j - \boldsymbol{\omega}_j) \wedge \mathbf{n}] . \end{aligned} \quad (20)$$

Collision with rotation

Combining the previous equation with Eq. (16), we obtain an expression without angular velocities

$$\mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} = (1 - e_t) \mathbf{u}_{ij}^t \quad (21)$$

$$= - \left[2 \left(\mathbf{v}_i^{t'} - \mathbf{v}_i^t \right) + 2q \left(\mathbf{v}_i^{t'} - \mathbf{v}_i^t \right) \right] \quad (22)$$

and finally

$$\mathbf{v}_i^{t'} = \mathbf{v}_i^t - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)} \quad \text{with} \quad q = \frac{mR^2}{I}. \quad (23)$$

Collision with rotation

Analogously, we find for the remaining quantities

$$\begin{aligned}\mathbf{v}_j^{t'} &= \mathbf{v}_j^t + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \boldsymbol{\omega}_i' &= \boldsymbol{\omega}_i - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}, \\ \boldsymbol{\omega}_j' &= \boldsymbol{\omega}_j - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}.\end{aligned}\tag{24}$$

And the updated velocities are

$$\begin{aligned}\mathbf{v}_i' &= \mathbf{v}_i - \mathbf{u}_{ij}^n - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \mathbf{v}_j' &= \mathbf{v}_j + \mathbf{u}_{ij}^n + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}.\end{aligned}\tag{25}$$

Inelastic collisions

The kinetic energy of interacting and colliding particles is not constant due to friction, plastic deformation or thermal dissipation. We account for energy dissipation effects in an effective manner by introducing the *restitution coefficient*. The restitution coefficient is defined as the ratio of the energy before and after the interaction event, and it describes multiple physical effects, i.e.,

$$r = \frac{E^{\text{after}}}{E^{\text{before}}} = \left(\frac{v^{\text{after}}}{v^{\text{before}}} \right)^2, \quad (26)$$

where E^{after} and E^{before} are the energies before and after the interaction event, and v^{after} and v^{before} are the corresponding velocities.

Inelastic collisions

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.

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

May 18, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

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.

Elastic Collisions

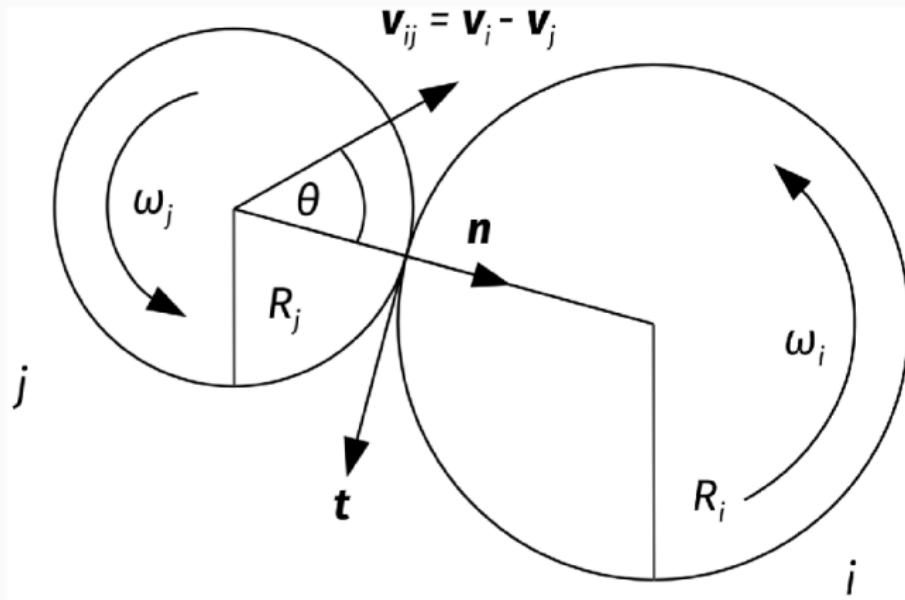


Figure 1: Two particles collide elastically.

Elastic Collisions

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

$$|\mathbf{r}_{ij}(t_{ij})| = |R_i + R_j| \quad (1)$$

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

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

Elastic Collisions

We should bear in mind that Eq. (2) 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.,

$$t_c = \min_{ij} (t_{ij}). \quad (3)$$

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

$$\mathbf{r}_i(t_0 + t_c) = \mathbf{r}_i(t_0) + \mathbf{v}_i(t_0)t_c \quad \text{and} \quad \phi_i(t_0 + t_c) = \phi_i(t_0) + \omega_i(t_0)t_c. \quad (4)$$

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.

Lubachevsky method

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.,

$$t^{(i)} = \min_j (t_{ij}). \quad (5)$$

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.

Lubachevsky method

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

$$t_c = \min_i \left(t^{(i)} \right). \quad (6)$$

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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.
- **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.

Lubachevsky method

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

- The vector part[m] points to particle i which is at position m in the stack. (Sometimes also a vector 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.
- 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 part[m] has to be reordered.

Lubachevsky method

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 perfect slip

We now approximate a collision by neglecting the tangential exchange of momentum – i.e. we assume a perfect slip. Only linear momentum and no angular momentum is exchanged. The conservation of momentum leads to

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} + \frac{\Delta \mathbf{p}}{m_i}, \quad (7)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} - \frac{\Delta \mathbf{p}}{m_j}, \quad (8)$$

and energy conservation:

$$\frac{1}{2}m_i (\mathbf{v}_i^{\text{before}})^2 + \frac{1}{2}m_j (\mathbf{v}_j^{\text{before}})^2 = \frac{1}{2}m_i (\mathbf{v}_i^{\text{after}})^2 + \frac{1}{2}m_j (\mathbf{v}_j^{\text{after}})^2. \quad (9)$$

Collision with perfect slip

The exchanged momentum is

$$\Delta \mathbf{p} = -2m_{\text{eff}} \left[(\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}}) \cdot \mathbf{n} \right] \mathbf{n} \quad (10)$$

with $m_{\text{eff}} = \frac{m_i m_j}{m_i + m_j}$ being the *effective mass* and $\mathbf{n} = \mathbf{r}_{ij}/|\mathbf{r}_{ij}|$. If $m_i = m_j$, the velocity updates are

$$\mathbf{v}_i^{\text{after}} = \mathbf{v}_i^{\text{before}} - v_{ij}^n \cdot \mathbf{n}, \quad (11)$$

$$\mathbf{v}_j^{\text{after}} = \mathbf{v}_j^{\text{before}} + v_{ij}^n \cdot \mathbf{n}, \quad (12)$$

with $v_{ij}^n = (\mathbf{v}_i^{\text{before}} - \mathbf{v}_j^{\text{before}}) \cdot \mathbf{n}$.

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

$$I \frac{d\omega_i}{dt} = \mathbf{r} \wedge \mathbf{f}_i, \quad (13)$$

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

$$\begin{aligned} I(\omega'_i - \omega_i) &= -Rm\mathbf{n} \wedge (\mathbf{v}'_i - \mathbf{v}_i), \\ I(\omega'_j - \omega_j) &= Rm\mathbf{n} \wedge (\mathbf{v}'_j - \mathbf{v}_j), \end{aligned} \quad (14)$$

with the primed velocities representing the ones after the collision.

Collision with rotation

Together with the conservation of momentum

$$\mathbf{v}'_i + \mathbf{v}'_j = \mathbf{v}_i + \mathbf{v}_j, \quad (15)$$

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

$$\omega'_i - \omega_i = \omega'_j - \omega_j = -\frac{Rm}{I} (\mathbf{v}'_i - \mathbf{v}_i) \wedge \mathbf{n}. \quad (16)$$

The relative velocity between particles i and j is

$$\mathbf{u}_{ij} = \mathbf{v}_i - \mathbf{v}_j - R(\omega_i + \omega_j) \wedge \mathbf{n}. \quad (17)$$

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

Collision with rotation

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

$$\begin{aligned}\mathbf{u}_{ij}^n &= (\mathbf{u}_{ij} \mathbf{n}) \mathbf{n}, \\ \mathbf{u}_{ij}^t &= \mathbf{u}_{ij} \wedge \mathbf{n} = [(\mathbf{v}_i - \mathbf{v}_j) - R(\boldsymbol{\omega}_i + \boldsymbol{\omega}_j)] \wedge \mathbf{n}.\end{aligned}\tag{18}$$

Collision with rotation

General slips are described by

$$\mathbf{u}_{ij}^{t'} = e_t \mathbf{u}_{ij}^t, \quad (19)$$

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

$$\begin{aligned} (1 - e_t) \mathbf{u}_{ij}^t &= \mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} \\ &= - [(\mathbf{v}'_i - \mathbf{v}_i - \mathbf{v}'_j + \mathbf{v}_j) - R (\boldsymbol{\omega}'_i - \boldsymbol{\omega}_i + \boldsymbol{\omega}'_j - \boldsymbol{\omega}_j) \wedge \mathbf{n}]. \end{aligned} \quad (20)$$

Collision with rotation

Combining the previous equation with Eq. (16), we obtain an expression without angular velocities

$$\mathbf{u}_{ij}^t - \mathbf{u}_{ij}^{t'} = (1 - e_t) \mathbf{u}_{ij}^t \quad (21)$$

$$= - \left[2 \left(\mathbf{v}_i^{t'} - \mathbf{v}_i^t \right) + 2q \left(\mathbf{v}_i^{t'} - \mathbf{v}_i^t \right) \right] \quad (22)$$

and finally

$$\mathbf{v}_i^{t'} = \mathbf{v}_i^t - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)} \quad \text{with} \quad q = \frac{mR^2}{I}. \quad (23)$$

Collision with rotation

Analogously, we find for the remaining quantities

$$\begin{aligned}\mathbf{v}_j^{t'} &= \mathbf{v}_j^t + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \boldsymbol{\omega}_i' &= \boldsymbol{\omega}_i - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}, \\ \boldsymbol{\omega}_j' &= \boldsymbol{\omega}_j - \frac{(1 - e_t) \mathbf{u}_{ij}^t \wedge \mathbf{n}}{2R(1 + q^{-1})}.\end{aligned}\tag{24}$$

And the updated velocities are

$$\begin{aligned}\mathbf{v}_i' &= \mathbf{v}_i - \mathbf{u}_{ij}^n - \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}, \\ \mathbf{v}_j' &= \mathbf{v}_j + \mathbf{u}_{ij}^n + \frac{(1 - e_t) \mathbf{u}_{ij}^t}{2(1 + q)}.\end{aligned}\tag{25}$$

Inelastic collisions

The kinetic energy of interacting and colliding particles is not constant due to friction, plastic deformation or thermal dissipation. We account for energy dissipation effects in an effective manner by introducing the *restitution coefficient*. The restitution coefficient is defined as the ratio of the energy before and after the interaction event, and it describes multiple physical effects, i.e.,

$$r = \frac{E^{\text{after}}}{E^{\text{before}}} = \left(\frac{v^{\text{after}}}{v^{\text{before}}} \right)^2, \quad (26)$$

where E^{after} and E^{before} are the energies before and after the interaction event, and v^{after} and v^{before} are the corresponding velocities.

Inelastic collisions

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.

Inelastic collisions

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

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

Inelastic collisions

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

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

In the case of perfect slip, the momentum exchange is

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

Inelastic collisions

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

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

These equations describe inelastic collisions of rotating particles.

Inelastic collisions

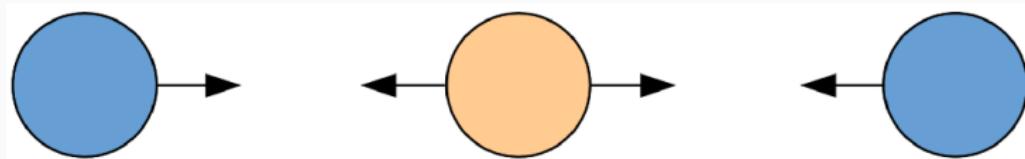


Figure 2: The orange particle at the center is bouncing between the blue particles which approach each other.

Inelastic collisions



Figure 3: A bouncing ball.

Inelastic collisions

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.

¹https://en.wikipedia.org/wiki/Zeno's_paradoxes

Inelastic collisions

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

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

Inelastic collisions

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.,

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

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

Inelastic collisions

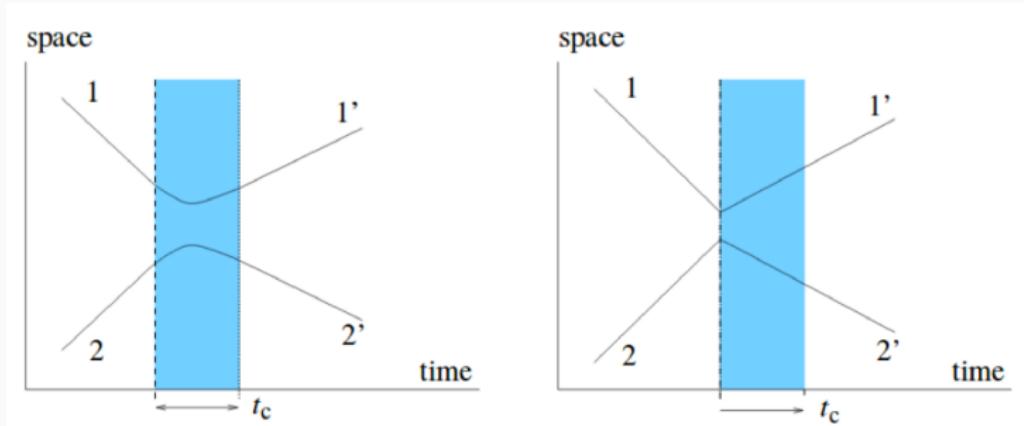


Figure 4: Trajectories of soft (left) and hard (right) particles. The figure is taken from [S. Luding, S. McNamara, *Granul. Matter* (1998).].

Inelastic collisions

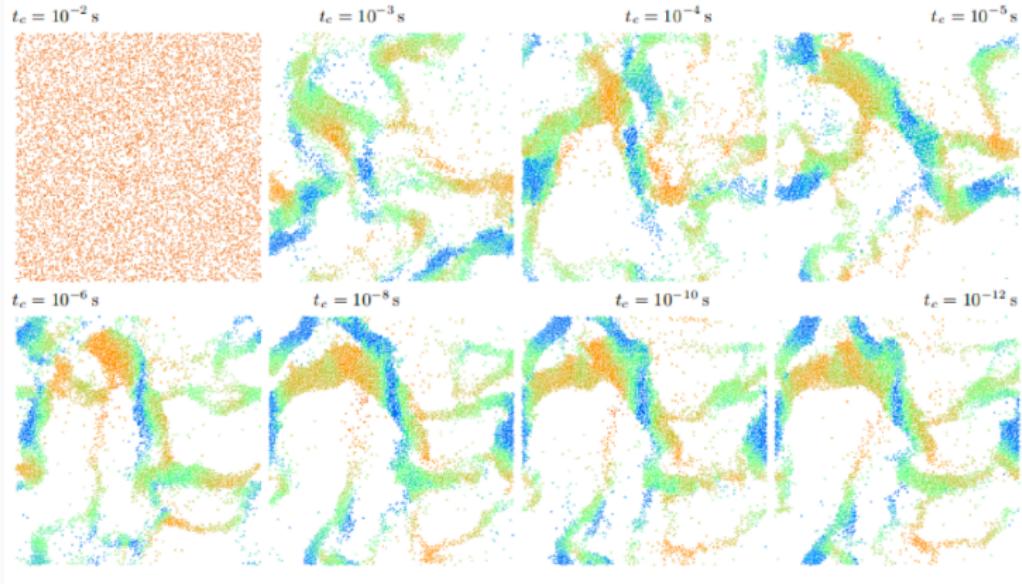


Figure 5: Examples of different contact times t_c . The figure is taken from [S. Luding, S. McNamara, *Granul. Matter* (1998).].

Contact dynamics

Contact Dynamics

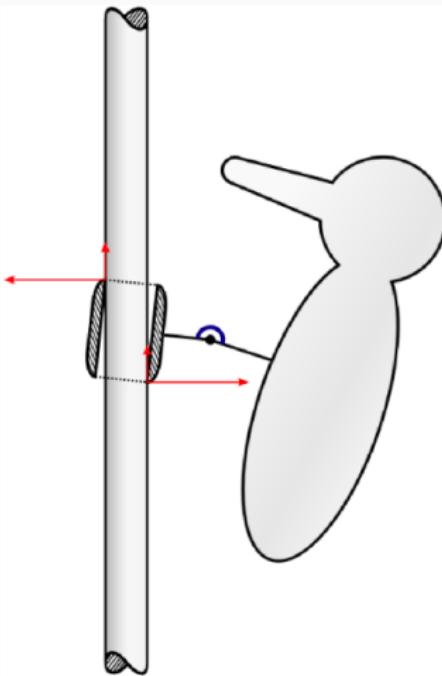


Figure 6: A woodpecker toy as a benchmark example for contact dynamics problems.

Contact Dynamics

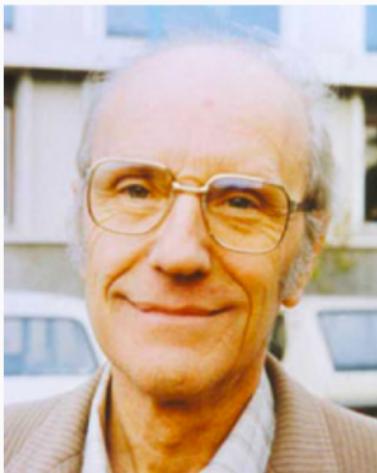


Figure 7: Per Lötstedt and Jean-Jacques Moreau contributed to development of contact dynamics.

Contact Dynamics

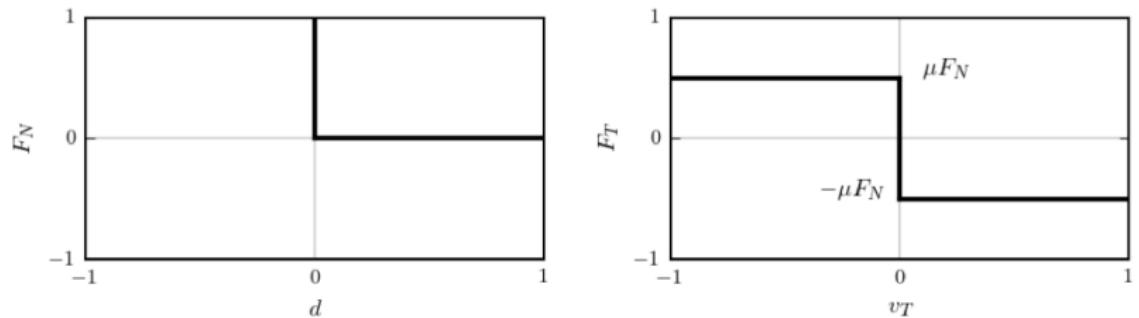


Figure 8: Signorini (left) and Coulomb (right) graphs.

One-dimensional Contact

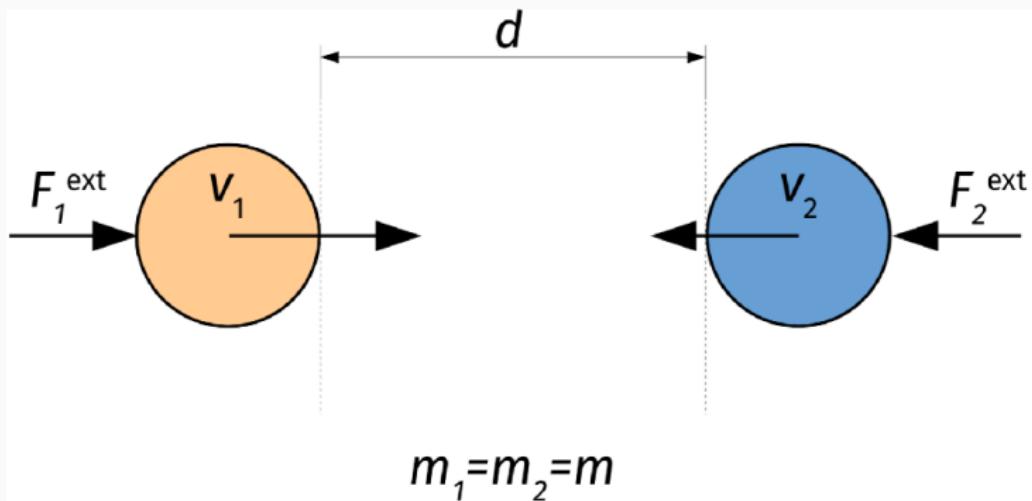


Figure 9: Illustration of a one-dimensional contact.

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

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

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

One-dimensional Contact

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

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

One-dimensional Contact

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.

One-dimensional Contact

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

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

One-dimensional Contact

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.

Three-dimensional 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

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

Three-dimensional Contact

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

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

Three-dimensional Contact

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

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

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.

Computational Statistical Physics

Part II: Interacting particles and molecular dynamics

Marina Marinkovic

May 25, 2022

ETH Zürich

Institute for Theoretical Physics

HIT G 41.5

Wolfgang-Pauli-Strasse 27
8093 Zürich



402-0812-00L
FS 2022

Particles in fluids

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

Particles in Fluids

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

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

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

Particles in Fluids

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

Particles in Fluids

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:

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

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

Particles in Fluids

In the Stokes limit for $\text{Re} \ll 1$, 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 $\text{Re} = 0$. In the case of turbulent flow for $\text{Re} \gg 1$, the drag force is (Newton's law)

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

Particles in Fluids

The general drag law is

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

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.

Particles in Fluids

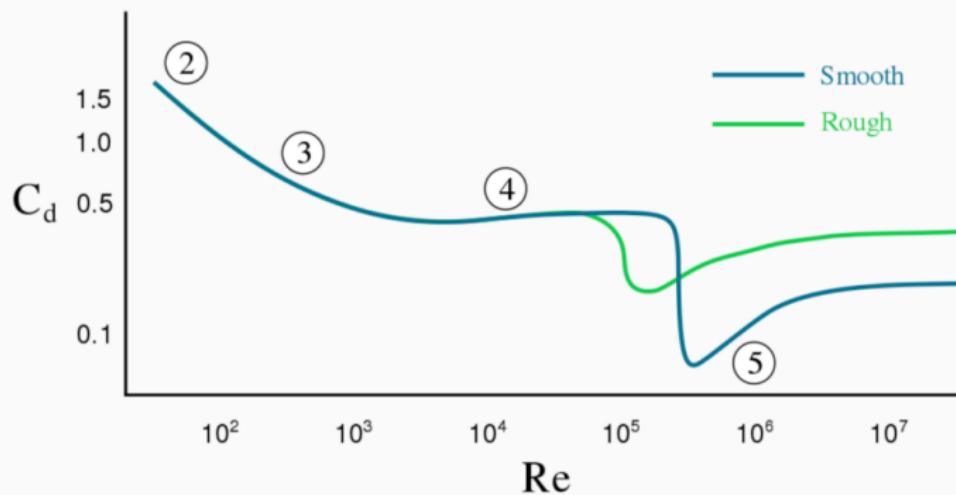


Figure 1: Dependence of the drag coefficient C_D on the Reynold's number.

Particles in Fluids

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.

Particles in Fluids

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

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

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

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

Particles in Fluids

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

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

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

Stokesian Dynamics [Brady, Bossis 1985]

In a method to study Stokesian dynamics ($\text{Re} \ll 1$), we start from the Stokes equation

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla p + \mu \Delta \mathbf{u} \quad (13)$$

The Green's function of the Stokes equation is the Stokeslet

$$G_{\alpha\beta}^S(\mathbf{r}) = \frac{1}{8\pi\eta} \left(\frac{\delta_{\alpha\beta}}{r} + \frac{r_\alpha r_\beta}{r^3} \right). \quad (14)$$

Stokesian Dynamics [Brady, Bossis 1985]

A general solution for the velocities fields of N particles is then

$$\mathbf{u}(\mathbf{x}) = - \sum_{i=1}^N \int_{\Gamma_i} G^S \Theta \mathbf{n} \, d\Gamma_i \quad (15)$$

The drag force on a surface element ijk is determined according to

$$\mathbf{f}_{ijk} = \Theta^{ijk} \mathbf{n}. \quad (16)$$

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

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

Lattice Boltzmann Method

One possible choice of the weights in two dimensions is

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

Lattice Boltzmann Method

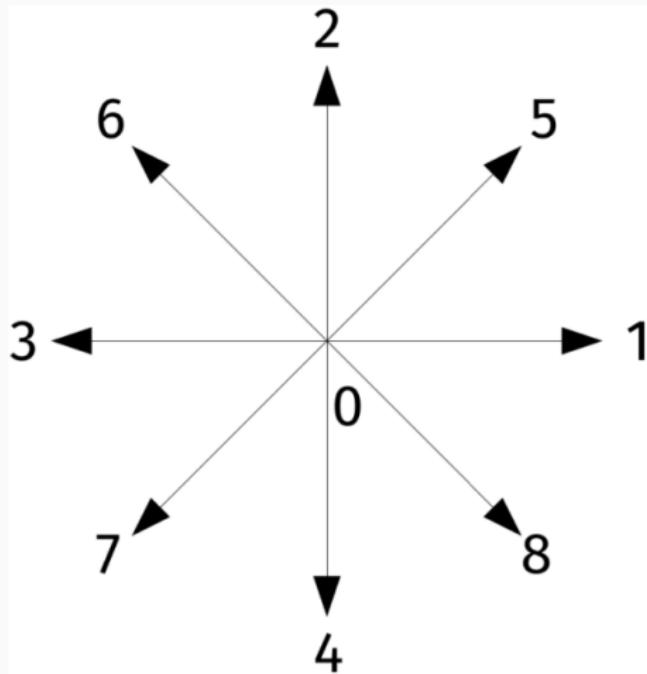


Figure 2: Lattice Boltzmann weights in 2 dimensions

Stochastic rotation dynamics

Stochastic Rotation Dynamics (SRD) is a particle-based fluid modeling approach [Malevanets, Kapral, 99]. This technique is also known as Multi-particle Collision Dynamics (MPC). In this method, we discretize the space into cells and model the fluid as a system composed of N particles with mass m and coordinates x_i and v_i . The particle positions and velocities are updated according to

$$\mathbf{x}_i' = \mathbf{x}_i + \Delta t \mathbf{v}_i, \quad (20)$$

$$\mathbf{v}_i' = \mathbf{u} + \Omega(\mathbf{v}_i - \mathbf{u}) + \mathbf{g} \quad (21)$$

where $\mathbf{u} = \langle \mathbf{v} \rangle$ is the mean velocity of particles in the respective cell and Ω is the rotation matrix.

Stochastic rotation dynamics

The rotation matrix is given by

$$\Omega = \begin{pmatrix} \cos(\alpha) & \pm \sin(\alpha) & 0 \\ \mp \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (22)$$

The collective fluid particle interaction is modeled by rotations of local particle velocities. In this model, Brownian motion is intrinsic. These very simple dynamics recovers hydrodynamics correctly.

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.

Direct simulation Monte Carlo

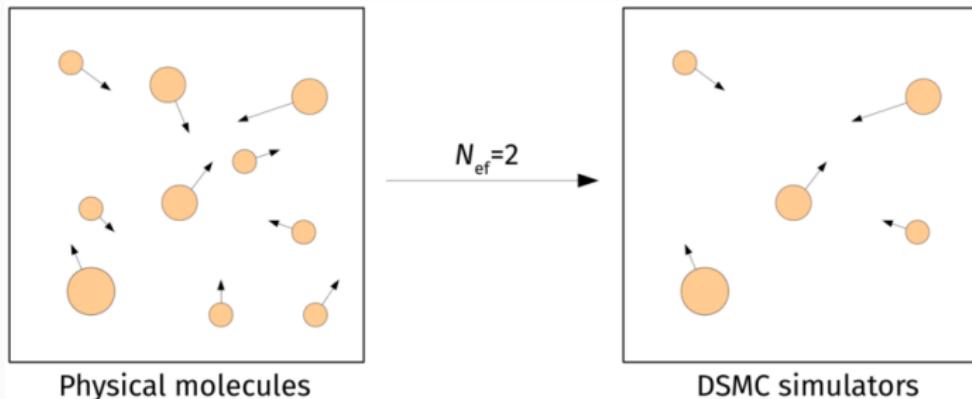


Figure 3: In DSMC N_{ef} simulators represent one physical particle.

Direct simulation Monte Carlo

Collisions 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.

Direct simulation Monte Carlo

The material surface may be treated with a thermal wall, which resets the velocity of a particle as a biased-Maxwellian distribution

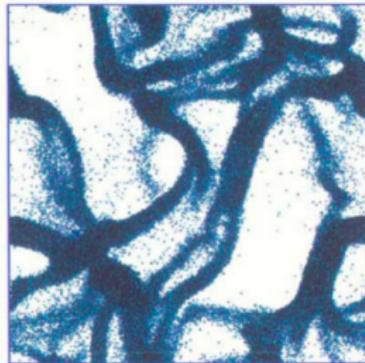
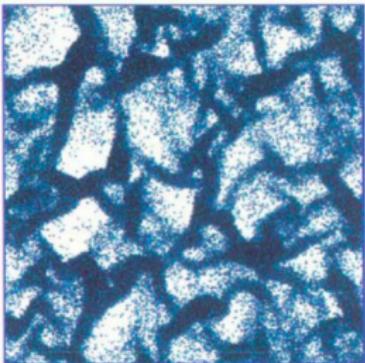
$$P_{v_x}(v_x) = \pm \frac{m}{k_B T_W} v_x e^{-\frac{mv_x^2}{2k_B T_W}} \quad (24)$$

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

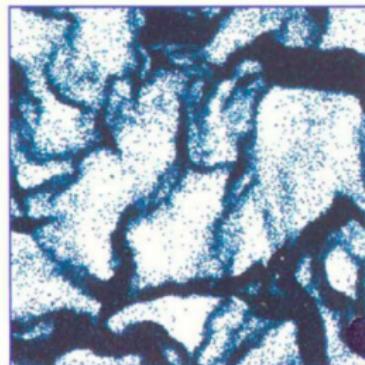
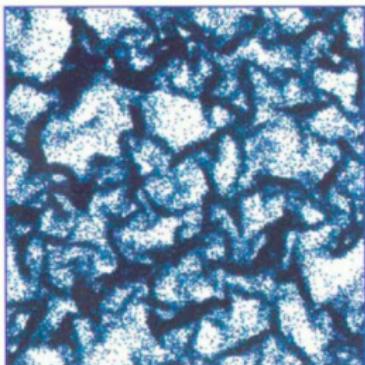
$$P_{v_z}(v_z) = \sqrt{\frac{m}{2\pi k_B T_W}} e^{-\frac{mv_z^2}{2k_B T_W}} \quad (26)$$

Direct simulation Monte Carlo

DSMC:



ED:



Dissipative Particle Dynamics [Hoogerbrugge, Koelman, '92]

Another particle-based fluid simulation approach is the so-called Dissipative Particle Dynamics (DPD). The particle interactions are described by

$$\mathbf{F}_i = \sum_{i \neq j} (\mathbf{f}_{ij}^C + \mathbf{f}_{ij}^R + \mathbf{f}_{ij}^D), \quad (27)$$

where \mathbf{f}_{ij}^C represents the conservative forces (e.g., momentum transfer), \mathbf{f}_{ij}^R a random force and \mathbf{f}_{ij}^D the dissipative forces, proportional to the velocity of the particles. The weights of the random and dissipative forces must be chosen such that thermal equilibrium is reached

Smoothed Particle Hydrodynamics

Another important technique in the field of computational fluid dynamics is Smoothed Particle Hydrodynamics (SPH). This method uses smooth kernel functions W to represent properties of particles in a weighted sense. Instead of localized positions and velocities, the particle characteristics are smoothed over a smoothing length h . An arbitrary quantity A is then given by

$$A(r) = \int_{\Omega} W(|r - r'|, h) A(r') dr' \approx \sum_j \frac{m_j}{\rho_j} W(|r - r_j|, h) A_j.$$

Smoothed Particle Hydrodynamics

In this method, no spatial discretization is necessary and even complex geometries can be interpolated and simulated with SPH. This makes this method broadly applicable in many different fields where fluids interact with complex structures. Examples of kernel functions include Gaussians or quadratic functions

$$W(r, h) = \frac{3}{2\pi h^2} \left(\frac{1}{4}q^2 - q + 1 \right) \quad (28)$$

with $q = \frac{h}{r}$ and $r = |r_a - r_b|$. Another advantage of this method is that kernels may be changed without much effort for a given simulation framework.