Definition

Computational Statistical Physics

Frühjahrssemester 2014

Giuseppe Accaputo

`www.accaputo.ch`

Computational Statistical Physics

Definition

Markov chain

Computational Statistical Physics

Formula

Total probability of a Markov chain

Computational Statistical Physics

Formula

Master equation

Computational Statistical Physics

Properties

Properties of the total probability $W(X \to Y)$ of a Markov chain

Computational Statistical Physics

Formula

Detailed balance

Computational Statistical Physics

Definition

Single flip Metropolis algorithm (Ising Model)

Computational Statistical Physics

Definition

Optimizations for the single flip Metropolis algorithm (3D Ising Model)

Computational Statistical Physics

Definition

Critical temperature T_c

Computational Statistical Physics

Definition

Spontaneous magnetization

Computational Statistical Physics

Mathematical system that undergoes transition from one state to another on a state space. It is a random process and it is memoryless, i.e. the next state depends only on the current state and not on the sequence of events

A master equation is a set of first-order differential equations describing the time evolution of (usually) the probability of a system to occupy each one of a discrete set of states with regard to a continuous time variable t:

$$\frac{\mathrm{d}p(X,t)}{\mathrm{d}t} = \sum_Y p(Y)W(Y \to X) - \sum_Y p(X)W(X \to Y)$$

Start in configuration X and propose new configuration Y with probability $T(X \to Y)$. The proposed configuration Y will be accepted with probability $A(X \to Y)$

$$W(X \to Y) = T(X \to Y) \cdot A(X \to Y)$$

At equilibrium, each elementary process should be equilibrated by its reverse process.

$$P_{\mathrm{eq}}(Y)W(Y \to X) = P_{\mathrm{eq}}(X)W(X \to Y)$$

In stationary state one should have equilibrium distribution (Boltzmann): $\frac{\mathrm{d}P(X,t)}{\mathrm{d}t} = 0 \Leftrightarrow P_{\mathrm{st}}(X) = P_{\mathrm{eq}}(X) \implies \sum_Y P_{\mathrm{eq}}(Y)W(Y \to X) = \sum_Y P_{\mathrm{eq}}(X)W(X \to Y) \implies$ Detailed balance is a sufficient condition for this equation to hold

- Ergodicity: $\forall X, Y : W(X \to Y) > 0$ (each configuration is reachable)
- Normality: $\sum_Y W(X \to Y) = 1$
- Homogeneity: $\sum_Y p_{\mathrm{st}}(Y)W(Y \to X) = p_{\mathrm{st}}(X)$

1. Store flip probabilities (in 2D there are 2, in 3D there are 3)

   $P(k) = \exp\{-4\beta Jk\} \qquad k = 1/2\sigma_i h_i = 1, 2, 3$ in 3D

2. Multi-spin coding: One word has 64 bits. Possible energy values in 3D: $E = 0 \ldots 6$. Use 3 bits to store energy value of site i. $64/3 = 21$ sites per word. Update 21 sites simultaneously and reduce memory requirement by a factor 21.

1. Choose a random site i with spin σ_i
2. Calculate $\Delta E = E(Y) - E(X) = 2J\sigma_i h_i$
   - $h_i = \sum_{\text{near. neighb. of}\sigma_i} \sigma_j$ is the local field at site i
3. If $\Delta E \leq 0$ then flip spin, i.e. $\sigma_i \to -\sigma_i$
4. If $\Delta E > 0$ flip with probability $\exp\{-\beta \Delta E\}$

Spontaneous magnetization is called the magnetization in the absence of an external magnetic field. This means that at low enough temperatures a given magnetic moment can influence the alignment of spins by neihgbor-to-neighbor interactions. At the critical temperature T_c the spontaneous magnetization vanishes.

The critical temperature T_c denotes the highest temperature for which there can be non-zero magnetization. At this point, the system undergoes and order-to-disorder transition, called a *phase transition*. For example, the spontaneous magnetization vanishes at temperatures higher than the critical temperature. For the 3D Ising model we have $T_c = 4.51$

Formula *Magnetization* Computational Statistical Physics	Formula *Magnetic susceptibility* Computational Statistical Physics
Formula *Specific heat* Computational Statistical Physics	Definition *Magnetic susceptibility χ* Computational Statistical Physics
Formula *Magnetic susceptibility χ (Monte Carlo simulations)* Computational Statistical Physics	Definition *Specific heat C* Computational Statistical Physics
Formula *Specific heat C (Monte Carlo simulations)* Computational Statistical Physics	Definition *Correlation length (Ising model)* Computational Statistical Physics
Formula *Correlation function and correlation length* Computational Statistical Physics	Formula *Correlation length at the critical temperature T_c* Computational Statistical Physics

$\chi = \frac{1}{N}\left(\frac{\partial M}{\partial H}\right) = \frac{k_B T}{N}\frac{\partial^2 \log Z}{\partial^2 H}$	$k_B T \frac{1}{Z}\frac{\partial Z}{\partial H} = k_B T \frac{\partial \log Z}{\partial H}$
The magnetic susceptibility χ is a parameter that shows how much an extensive parameter changes when an intensive paramater increases, thus χ tells us how much the magnetization changes by increasing the temperature. At the critical temperature T_c we observe that $\chi \to \infty$	$C_V = \frac{\partial E}{\partial T}$
The specific heat C tells us how much the energy changes with increasing temperature. At the critical temperature T_c we observe a divergence of the specific heat.	$\chi = \beta\left[\langle M^2\rangle - \langle M\rangle^2\right]$
The correlation length is the typical size of the clusters that emerge at lower temperatures ($T < T_c$) and shrink in size at higher temperatures.	$C = \beta^2\left[\langle E^2\rangle - \langle E\rangle^2\right]$
The correlation length diverges at T_c as: $\xi \propto \lvert T - T_c\rvert^{-\nu}$ with the critical exponent ν. At T_c we have for large R: $C(R) \propto R^{2-d-\eta}$	A correlation function describes the influence of particles on one another and is a measure of the order in a system. The correlation function is defined as follows: $C(R) = \langle\sigma(0)\sigma(R)\rangle$ where R is the radius. For $T \neq T_c$ and for large R we have: $C(R) \propto M^2 + a\exp\{-R/\xi\}$ where ξ is the correlation length

Formula *Non-linear correlation function* Computational Statistical Physics	Formula *Non-linear correlation time* Computational Statistical Physics
Formula *Critical slowing down (non-linear correlation time)* Computational Statistical Physics	Formula *Linear correlation function* Computational Statistical Physics
Formula *Linear correlation time* Computational Statistical Physics	Formula *Critical slowing down (linear correlation time)* Computational Statistical Physics
Definition *How to generate decorrelated configurations* Computational Statistical Physics	Formula *Criical exponents of the correlation-times (linear and non-linear)* Computational Statistical Physics
Definition *Critical exponent* Computational Statistical Physics	Definition *Critical exponents for specific heat, spontaneous magnetization, susceptibility, and correlation length* Computational Statistical Physics

The non-linear correlation time describes the relaxation *towards* equilibrium and is defined as:

$$\tau_A^{\mathrm{nl}} = \int_0^\infty \Phi_A^{\mathrm{nl}}(t)\mathrm{d}t$$

Example:

$$\Phi_A^{\mathrm{nl}}(t) = \exp\{-t/\tau_A^{\mathrm{nl}}\}$$

$-\tau_A^{\mathrm{nl}}$ is the slope of the line resulting from plotting the time t against $\log\{\Phi_A^{\mathrm{nl}}(t)\}$ for $T \neq T_c$

Suppose that the configuration at t_0 is not at equilibrium, then define the non-linear-correlation function as follows:

$$\Phi_A^{\mathrm{nl}}(t) = \frac{\langle A(t)\rangle - \langle A(\infty)\rangle)}{\langle A(t_0)\rangle - \langle A(\infty)\rangle}$$

$\langle A(\infty)\rangle$ is supposed to be at equilibrium.

With two quantities A and B in equilibrium define the linear time correlation function as follows:

$$\Phi_{AB}(t) = \frac{\langle A(t_0)B(t)\rangle - \langle A\rangle\langle B\rangle}{\langle AB\rangle - \langle A\rangle\langle B\rangle}$$

Near the critical temperature, the relaxation time becomes very large and can be shown to diverge for an infinite system: $\tau \sim \xi^z \sim |T - T_c|^{\nu z}$ This phenomenon is called critical slowing down.

$$\tau_A^{\mathrm{nl}} \propto |T - T_c|^{z_A^{\mathrm{nl}}}$$

z_A^{nl} is the non-linear dynamical critical exponent

$$\tau_{AB} \propto |T - T_c|^{z_{AB}}$$

z_{AB} is the linear dynamical critical exponent

The linear correlation time describes the relaxation *in* equilibrium and is defined as:

$$\tau_A^{\mathrm{nl}} = \int_0^\infty \Phi_{AB}(t)\mathrm{d}t$$

Example:

$$\Phi_{AB}(t) = \exp\{-t/\tau_{AB}\}$$

$$\tau^{\mathrm{nl}}(T_c) = L^{z^{\mathrm{nl}}/\nu}$$
$$\tau(T_c) = L^{z/\nu}$$

First, throw away $n_0 = c\tau^{\mathrm{nl}}(T)$ configurations to reach equilibrium. Then only take $n_{\mathrm{eq}} = c\tau(T)$ configuration to have decorrelated samples. Use $c \approx 3$ as a safe value.

$$C_V \propto |T - T_c|^{-\alpha}$$
$$M_S \propto |T - T_c|^{\beta}$$
$$\chi \propto |T - T_c|^{-\gamma}$$
$$\xi \propto |T - T_c|^{-\nu}$$

Critical exponents describe the behaviour of physical quantities near continues phase transitions (e.g. second-order phase transitions). We want to describe the behaviour of a physical quantity F in terms of a power law around the critical temperature.

Definition

Finite size scaling

Computational Statistical Physics

Formula

Thermal average

Computational Statistical Physics

Definition

Fluctuation-dissipation theorem

Computational Statistical Physics

Derivation

Derive fluctuation-dissipation theorem for the susceptibility (Part 1 of 2)

Computational Statistical Physics

Derivation

Derive fluctuation-dissipation theorem for the susceptibility (Part 2 of 2)

Computational Statistical Physics

Definition

Kawasaki dynamics with Glauber spin flip

Computational Statistical Physics

Definition

Idea behind Creutz's demon algorithm

Computational Statistical Physics

Definition

Creutz's demon algorithm for the Ising model

Computational Statistical Physics

Formula

Binder cumulant

Computational Statistical Physics

Plot

Plot of Binder cumulant for system sizes $N = 4, 6, 8$

Computational Statistical Physics

$$\langle Q(T)\rangle = \frac{1}{Z_T}\sum_X Q(X)\exp\{-E(X)/(k_BT)\}$$

Finite size scaling is a method to find the values of critical exponents by observing how measured quantities vary for different lattice sizes.

$$\chi(T) = \left.\frac{\partial\langle M(T,H)\rangle}{\partial H}\right|_{H=0}$$
$$= \left.\frac{\partial}{\partial H}\frac{\sum_X\sum_{i=1}^N \sigma_i \exp\{H_0+\beta H\sum_{i=1}^N\sigma_i\}}{\sum_X\exp\{H_0+\beta H\sum_{i=1}^N\sigma_i\}}\right|_{H=0}$$
with $H_0 = \beta J\sum_{i,j:nn}^N \sigma_i\sigma_j$ and $\beta = \frac{1}{k_BT}$

The fluctuation-dissipation theorem states that the linear response of a given system to an external perturbation is expressed in terms of fluctuation properties of the system in thermal equilibrium.
Dissipation: energy is transformed from some initial form to some final form; the capacity of the final form to do mechanical work is less than that of the initial form.

1. Choose any $A-B$ bond
2. Calculate ΔE for $A-B \to B-A$
3. Flip with probability $p = \exp\{-\beta\Delta E\}/(1+\exp\{-\beta\Delta E\})$ (Glauber)

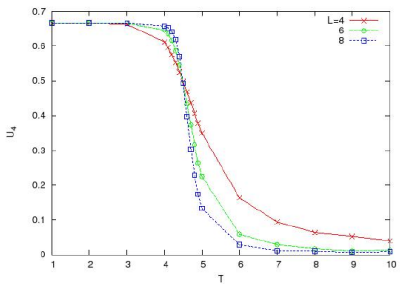
Glauber's spin-flip probability fulfills detailed balance.

$$\chi(T) = \left.\frac{\beta\sum_X\left(\sum_{i=1}^N\sigma_i\right)^2\exp\{H_0+\beta H\sum_{i=1}^N\sigma_i\}}{Z_T(H)}\right|_{H=0}$$
$$\left.-\frac{\beta\left(\sum_X\sum_{i=1}^N\sigma_i\exp\{H_0+\beta H\sum_{i=1}^N\sigma_i\}\right)^2}{(Z_T(H))^2}\right|_{H=0}$$
$$= \beta\left(\langle M(T)^2\rangle - \langle M(T)\rangle^2\right) \implies \chi(T)\geq 0$$

1. Equilibrate system to reach a given target energy E_{target}
2. Flip random spin
3. Calculate ΔE
4. If $\Delta E < 0$ then the flip is accepted, i.e. $E_{\text{demon}} = |\Delta E| + E_{\text{demon}}$
5. If $\Delta E > 0$ then flip is only accepted if $E_{\text{demon}} > \Delta E$. Else the flip is rejected

Creutz's demon algorithm is used to simulate a microcanonical ensemble using a Monte Carlo simulation. The energy is kept constant thanks to the existence of an energy demon. The algorithm itself does not use random numbers

One can see that by increasing the system size we get a clear cut at $T = 4.51 = T_c$



The Binder cumulant is used to approximate T_c.

$$U_L \equiv 1 - \frac{\langle M^4\rangle_L}{3\langle M^2\rangle_L^2} \underset{L\to\infty}{\longrightarrow} \begin{cases} 0 \text{ for } T > T_c \\ 2/3 \text{ for } T < T_c \end{cases}$$

Definition

First-order phase transitions (Potts model, Ising model)

Computational Statistical Physics

Definition

Second-order phase transitions

Computational Statistical Physics

Definition

The Potts model

Computational Statistical Physics

Definition

Swendsen-Wang algorithm (3D Ising model)

Computational Statistical Physics

Definition

Wolff algorithm (3D Ising model)

Computational Statistical Physics

Formula

Verlet method to calculate $\vec{x}(t + \Delta t), \vec{v}(t)$

Computational Statistical Physics

Formula

Leap frog method to calculate $\vec{x}(t + \Delta t), \vec{v}(t + 1/2\Delta t)$

Computational Statistical Physics

Definition

Verlet tables

Computational Statistical Physics

Definition

Linked cell method

Computational Statistical Physics

Definition

Particle-Mesh algorithm

Computational Statistical Physics

Second-order phase transitions (continuous phase transitions) are characterized by a divergent susceptibility, an infinite correlation length, and a power-law decay of correlations near criticality.

For $T < T_c$ the Ising model has at $H = 0$ a phase transition (order-to-disorder transition) in H of the first order, i.e. the system has a jump in magnetization ΔM and latent heat ΔE

One Swendsen-Wang step consists of the following: For every spin $\sigma_{x,y,z}$ check if it has already been discovered, i.e. is already part of a cluster. If yes, go to the next spin; else start a breadth-first-search from the current spin $\sigma_{x,y,z}$ during which a cluster is built. Add an aligned nearest-neighbor (NN) of $\sigma_{x,y,z}$ with probability $p_{\text{add}} = 1 - \exp\{-2 \cdot J \cdot \beta\}$ to the cluster. After having checked all aligned NNs, flip the current spin $\sigma_{x,y,z}$ with probability $p_{\text{flip}} = 0.5$.

Model consisting of q states, i.e. $\sigma_i = 1, \ldots, q$ with the Hamiltonian H

$$H = E = -J \sum_{i,j=nn} \delta_{\sigma_i,\sigma_j} - H_1 \sum_i \delta_{\sigma_i 1}$$

The Potts model with $q = 2$ corresponds to the Ising model.

$$\ddot{\vec{x}}_i(t) = 1/m_i \sum_j \vec{f}_{ij}(t), \qquad \vec{f}_{ij} = -\nabla V(r_{ij}(t))$$

$$\vec{x}_i(t + \Delta t) = 2\vec{x}_i(t) - \vec{x}_i(t - \Delta t) + \Delta t^2 \ \ddot{\vec{x}}_i(t)$$

$$\vec{v}_i(t) = \frac{\vec{x}_i(t + \Delta t) - \vec{x}_i(t - \Delta t)}{2\Delta t}$$

Only store two time steps (t and $t - \Delta t$). Error is $O(\Delta t^4)$, i.e. third-order algorithm

One Wolff step consists of the following: Pick a random spin $\sigma_{x,y,z}$ and start to construct a single cluster from there. At the beginning $\sigma_{x,y,z}$ is added to the cluster and is also flipped. From there now construct the cluster recursively by adding aligned nearest-neighbors (NNs) to the cluster with probability $p_{\text{add}} = 1 - \exp\{-2 \cdot J \cdot \beta\}$ and flipping each one of the spins after having added it to the cluster. This algorithm is an improvement over the Swendsen-Wang algorithm because it has larger probability of flipping bigger clusters.

Define around each particle i a neighborhood of radius $r_l > 2r_c$. Vector *list* contains all neighborhoods. Vector *point[i]* contains the index of the first particle in the neighborhood of i. Particles in the neighborhood of i are: *list[point[i]], ..., list[point[i+1]-1]*. Update the Verlet table every $n = \frac{r_l - 2r_c}{\Delta t v_{\max}}$; the algorithm runs in $O(N^2)$.

$$\ddot{\vec{x}}_i(t) = 1/m_i \sum_j \vec{f}_{ij}(t), \qquad \vec{f}_{ij} = -\nabla V(r_{ij}(t))$$

$$\vec{v}_i(t + 1/2\Delta t) = \vec{v}_i(t - 1/2\Delta t) + \Delta t \ \ddot{\vec{x}}(t)$$

$$\vec{x}_i(t + \Delta t) = \vec{x}_i(t) + \Delta t \vec{v}_i(t + 1/2\Delta t)$$

Put a fine mesh on top of the system. Distribute charges onto the mesh points. Calculate electrostatic potential by solving the Poission equation on the mesh using FFT. Calculate force on each particle by numerically differentiating the potential and interpolating back from the mesh to the particle position.

Divide domain in M^d cells of length $r_l > 2r_c$. On average we only need to test $N \cdot 3^d N/M^d$ particles. Vector *cells* contains for each cell a list of all particles within the cell. The algorithm is $O(N)$.

Definition *Particle-Mesh algorithm types* Computational Statistical Physics	Definition *Criteria for a good particle-mesh scheme* Computational Statistical Physics
Definition *Particle-particle particle mesh (P^3M) algorithm* Computational Statistical Physics	Definition *Constraint method with Lagrange multipliers to implement fixed distances inside the water molecule* Computational Statistical Physics
Definition *Constraint method: steps to calculate new position* Computational Statistical Physics	Definition *Rigid bodies: equation of motion for rotation* Computational Statistical Physics
Definition *Evolution of rotation angle in 2D using Verlet algorithm* Computational Statistical Physics	Definition *Evolution of rotation angle in 3D* Computational Statistical Physics
Definition *Quaternions* Computational Statistical Physics	Definition *Nosé-Hoover thermostat: idea* Computational Statistical Physics

1. Errors should vanish at large particle distances
2. Momentum conservation: $\vec{F}_{ij} = -\vec{F}_{ji}$

1. Nearest Grid Point: Put particle on nearest grid point and also evaluate its force at the nearest grid point
2. Cloud In Cell: Assign the charge to the 2nd nearest grid points and also interpolate from these 2^d grid points.

Water molecule H_2O consists of 3 atoms: a_1 is the upper-left H atom, a_2 is the O atom, and a_3 is bottom-right H atom. Constraints that bonds have length d_{12} and d_{23}:

$\chi_{12} = r_{12}^2 - d_{12}^2 = 0$
$\chi_{23} = r_{23}^2 - d_{23}^2 = 0$
with $r_{ij} = \|\vec{r}_{ij}\|$
$\vec{g}_k = \frac{1}{2}\lambda_{12}\vec{\nabla}\vec{x}_k\chi_{12} + \frac{1}{2}\lambda_{23}\vec{\nabla}\vec{x}_k\chi_{23} \quad \lambda_{ij}$ Lagrange mult.

Split force into short and long range part: $\vec{F} = \vec{F}_s + \vec{F}_l$. $\vec{F}_l$ is small and smooth at short distances and is calculated using the particle-mesh algorithm. $\vec{F}_s$ is calculated exactly by solving Newton's equation. Adaptive P^3M: refine the mesh in the regions where the density of masses is dense (e.g. cluster massing under gravity, $F_s \sim O(N^2)$; else when mass distribution is homogeneous: $F_l \sim O(N \log N), F_s \sim O(N)$

$$I\dot{\omega} = T$$

1. $m_i\ddot{\vec{x}}_i = \vec{f}_i + \vec{g}_i$
2. Introduce Lagrange multipliers λ_{ij} in $\vec{g}_i$
3. Execute Verlet algorithm in two steps (one with $\vec{f}_i$, another with $\vec{g}_i$)
4. Obtain λ_{ij} by inserting expressions into the constraint condition
5. Solve resulting coupled quadratic equations and use resulting λ_{ij} to calculate $\vec{x}_i(t + \Delta t)$

Tensor of inertia: $\overleftrightarrow{I} = \sum_{i=1}^{n} m_i(d_i^T \otimes d_i - d_i^2 \overleftrightarrow{1})$. Its eigenvectors span a body-fixed coord. system with origin in the center of mass. Transform from laboratory-fixed $(\cdot^l)$ to body-fixed $(\cdot^b)$ system with $\overleftrightarrow{A}$: $\vec{e}^b = \overleftrightarrow{A} \cdot \vec{e}^l$. Use $\vec{T}^b = \overleftrightarrow{A}\,\vec{T}^l$ and $\vec{\omega}^l(t + \Delta t) = \overleftrightarrow{A}^T\,\vec{\omega}^b(t + \Delta t)$. For rotations ($\overleftrightarrow{A}$ is a combination of rotations), use Euler angles ϕ, θ, ψ.

$$\gamma(t + \Delta t) = 2\gamma(t) - \gamma(t - \Delta t) + \Delta t^2 \frac{T(t)}{I}$$

We considered constant energy and constant volume, i.e. we worked in the microcanonical ensemble. Most commonly, however, experiments are performed at constant temperature, i.e. in the canonical ensemble. We couple the system to a heat bath to maintain a constant temperature.

$$Q = (q_0, q_1, q_2, q_3) \text{ with } q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$$

Formula *Nosé-Hoover thermostat: potential and kinetic energy* Computational Statistical Physics	Formula *Nosé-Hoover thermostat: new Hamiltonian and new equations of motion in the new time t'* Computational Statistical Physics
Definition *Nosé-Hoover thermostat: selection of Q* Computational Statistical Physics	Definition *Important fact about the Nosé-Hoover thermostat* Computational Statistical Physics
Definition *Event driven simulations* Computational Statistical Physics	Definition *Event driven molecular dynamics* Computational Statistical Physics
Definition *Event driven simulations: Lubachevsky's Tricks* Computational Statistical Physics	Definition *Event driven simulations: Steps* Computational Statistical Physics
Definition *Inelastic collisions* Computational Statistical Physics	Definition *Inelasticity* Computational Statistical Physics

$$H_{\text{new}} = \sum_{i=1}^{N} \frac{\vec{p_i}^2}{2m_i s^2} + \frac{1}{2}Q\dot{s}^2 + V(\vec{x_1}, \dots \vec{x_N}) + V(s)$$

$$Q\ddot{s} = \sum_{i=1}^{N} m_i s\, \dot{\vec{x}}_i^2 - \frac{1}{s}(m+1)k_B T$$

set $\xi \equiv \frac{\dot{s}}{s}$ to express equations in real time t

use $\frac{\mathrm{d}\log s}{\mathrm{d}t} = \xi$

$$V(s) = (m+1)k_B T \log s$$

$$K(s) = \frac{1}{2}Q\dot{s}$$

s is the new degree of freedom that describes the heat bath. It introduces a new time scale $\mathrm{d}t' = s\ \mathrm{d}t$

Hoover proved in 1985 that the Nosé-Hoover thermostat is the only method with a single friction parameter that gives the canonical distribution.

- Q is too large $\implies$ equilibration is too slow
- $Q \to \infty \implies$ recovers microcanonical molecular dynamics (but we want to simulate a canonical ensemble!)
- Q is too small $\implies$ temperature exhibits spurious oscillations. Use $\overline{\Delta T} = \sqrt{2/(Nd)}\ \overline{T}$ as the width of the temperature distribution (d is the dimension, N is the number of particles)

No forces are calculated in this method. Only binary collisions are considered, i.e. interactions between three or more particles are neglected. One needs to calculate the time t_c between two collisions and then obtain the velocities of the two particles after the collision from the velocities of the particles before the collision from a look-up table.

In event driven simulations the collisions between particles are considered as instantaneous events and between them particles do not interact. The simulation of rigid objects of finite volume (e.g. billiard balls) cannot be done in classical MD because of the hard core potential which results in infinite forces.

1. Calculate new collision times using cell-lists: $O(1)$
2. Reorder heap of collision times: $O(\log N)$
3. Move particles for smallest time available
4. Update particles involved with collision rule

Loop to calculate t_c is of order N^2. Lubachevsky trick: keep track of the time of the event and the partner particle involved in the event in a list of length N. For all particles store: position, velocity, last event time, last event partner, next event time, next event partner.

Inelasticity is described via the restitution coefficient r.

- $r = 1$: elastic collisions (perfect bounce)
- $r = 0$: plastic collisions (plastic bounce)

During inellastic collisions energy is dissipated through vibrations and eventually also small plastic deformation or heat production. Dissipation is quantified through the material dependent restitution coefficient r. One has $r = 1$ for elastic collisions and $r = 0$ in case of perfect plasticity.

Definition

Finite time singularity

Computational Statistical Physics

Definition

Aim of histogram methods

Computational Statistical Physics

Definition

Broad histogram method

Computational Statistical Physics

Definition

Broad histogram method: problem of sampling

Computational Statistical Physics

Definition

Broad histogram method: problem of histogram methods

Computational Statistical Physics

Definition

Broad histogram method: Equivalent condition to detailed balance to reach a homogeneous steady state

Computational Statistical Physics

Definition

Broad histogram method with Metropolis

Computational Statistical Physics

The aim of histogram methods is to obtain functions at one temperature from a simulation at another temperature.

$$P_T(E) = g(E)\exp\{-E/(k_BT)\}$$

$g(E)$ describes the density of states, i.e. the number of configurations that have energy E

If an inelastic sphere jumps on a plate it will perform in a finite time t_{tot} an infinity of collisions.

$$t_{\text{tot}} = \sum_{j=1}^{\infty} t_j = 2\sqrt{\frac{2h^{\text{initial}}}{g}} \sum \sqrt{r^j}$$
$$= 2\sqrt{\frac{2h^{\text{initial}}}{g}} \left(\frac{1}{1-\sqrt{r}} - 1\right)$$

Distribution of average energy $\langle E\rangle$ gets sharper with increasing size. Choosing configurations equally distributed over energy would be very ineffective, since $P(E)$ is mostly zero and only peaks at $E = \langle E\rangle_T$.

Use a non biased random walk along the E axis. Assume $\Delta E > 0$. Then, if $E \to E - \Delta E$, accept the move. Else, if $E \to E + \Delta E$ accept move with probability $N_{\text{down}}/N_{\text{up}}$ (removes bias, forcing probabilities of increasing and decreasing the energy to be equal). The region already visited along the E axis increases its width proportionally to $\Delta E\sqrt{t}$, like a random walk, where t is the number of performed movements, i.e. the length of the Markovian sequence of states.

Following the non biased random walk dynamics of the broad histogram method, the probability for the energy to jump from E to $E+\Delta E$ is the same as that of jumping back from $E + \Delta E$ to E, which can be mathematically stated as

$$\langle N_{\text{down}}(E + \Delta E)\rangle \cdot g(E + \Delta E) = \langle N_{\text{up}}(E)\rangle \cdot g(E)$$

Calculate $Q(T')$ using $Q(T) = 1/Z_T \sum_E Q(E)P_T(E)$ with $Z_T = \sum_E P_T(E)$. Use $P_{T'} = g(E)\exp\{-E/(k_BT')\} = P_T(E)\exp\{-E/(k_BT') + E/(k_BT)\}$. P_T is the canonical Boltzmann probability distribution. Values of a quantity $Q(T)$ are sampled close to the maximum of $P_T(E)$ which for large systems is very peaked. If T and T' are not too close the overlap between the distribution is very small so that very few configuratoins are sampled around the maximum uf $T' \implies$ bad statistics

1. Choose a new configuration by flipping randomly a spin
2. If $E \to E - \Delta E$, then accept
3. If $E \to E + \Delta E$, then accept with probability $\frac{N_{\text{down}}(E+\Delta E)}{N_{\text{up}}(E)}$