



Monte Carlo

Monte Carlo method is a common name for a wide variety of stochastic techniques. These techniques are based on the use of random numbers and probability statistics to investigate problems in areas as diverse as economics, nuclear physics, and flow of traffic. In general, to call something a "Monte Carlo" method, all you need to do is use random numbers to examine your problem. Materials science related examples include

- "classical" Monte Carlo: samples are drawn from a probability distribution, often the classical Boltzmann distribution, to obtain thermodynamic properties or minimum-energy structures;
- "quantum" Monte Carlo: random walks are used to compute quantum-mechanical energies and wavefunctions, often to solve electronic structure problems, using Schrödinger's equation as a formal starting point;
- "volumetric" Monte Carlo: random number generators are used to generate volumes per atom or to perform other types of geometrical analysis;
- "kinetic" Monte Carlo: simulate processes using scaling arguments to establish time scales or by introducing stochastic effects into molecular dynamics.

In this course we will consider so-called classical Metropolis Monte Carlo and, later, the basics and a few examples of kinetic Monte Carlo.

Metropolis Monte Carlo – generates configurations according to the desired statistical-mechanics distribution. There is no time, the method cannot be used to study evolution of the system. Equilibrium properties can be studied.

Kinetic Monte Carlo – can address kinetics. The main idea behind KMC is to use transition rates that depend on the energy barrier between the states, with time increments formulated so that they relate to the microscopic kinetics of the system.

The Metropolis Monte Carlo was developed in present form by Metropolis, Ulam, Neumann during their work on Manhattan project (study of neutron diffusion). The name "Monte Carlo" was first used in a paper published by Metropolis and Ulam in 1949 and is attributed by some sources to the fact that S. Ulam's uncle went each year to Monte Carlo for gambling. Others attribute the name to S. Ulam's own interest in poker.

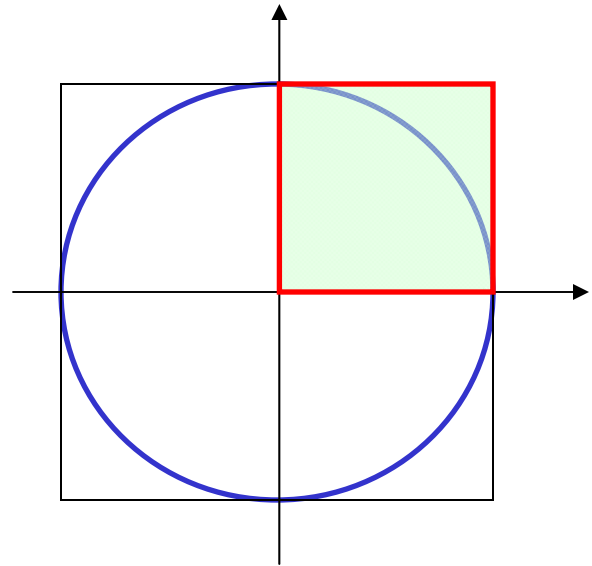


Monte Carlo Calculation of π

Let us illustrate the use of the MC technique as a method of integration and let us begin with a simple geometric MC experiment which calculates the value of π based on a "hit and miss" integration.

```
x=(random #)  
y=(random #)  
R= (x2 + y2)1/2  
if R < 1 then hits=hits+1.0
```

$$\pi = \frac{4N_{\text{hit}}}{N_{\text{shot}}}$$





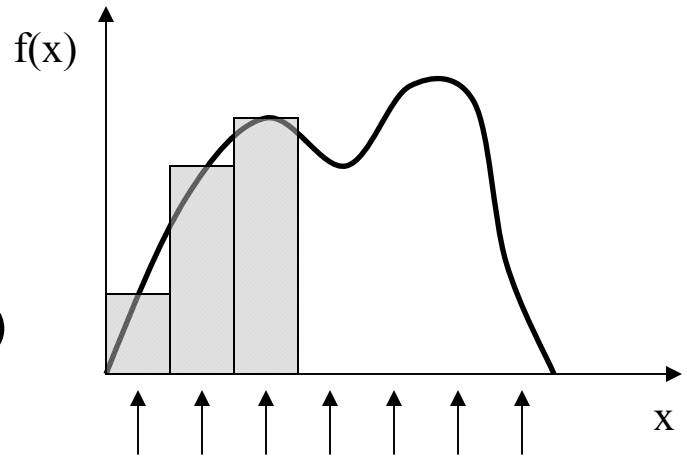
One-dimensional Integrals

Conventional approaches: rectangle rule, trapezoid rule, Simpson's rule ...

$$I = \int_a^b f(x) dx$$

Quadrature formula

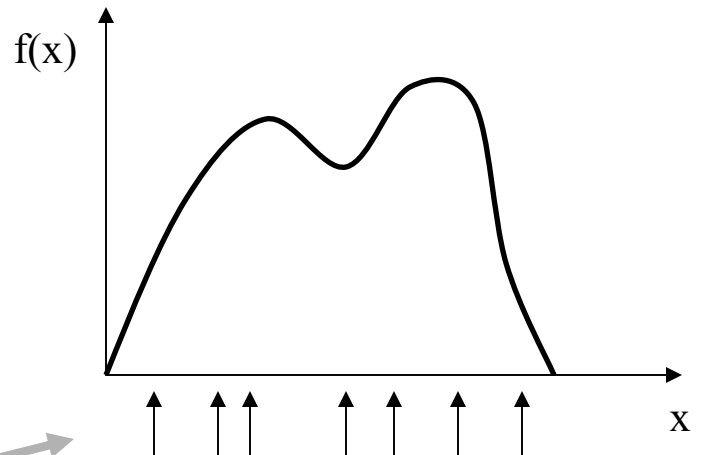
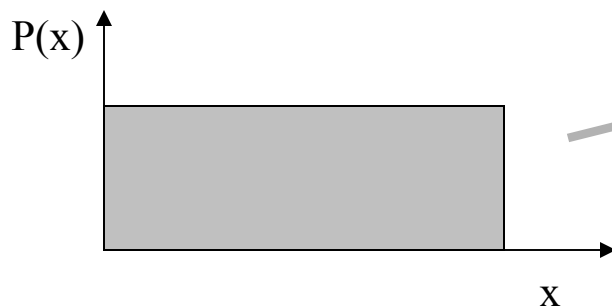
$$I \approx \Delta x \sum_{i=1}^n f(x_i) = \frac{b-a}{n} \sum_{i=1}^n f(x_i)$$



n uniformly separated points

Monte Carlo Integration: same quadrature formula, different selection of points

$$I \approx \frac{b-a}{n} \sum_{i=1}^n f(x_i)$$



n points selected at random from uniform distribution $P(x)$



Errors in Monte Carlo vs. methodical sampling

Methodical approach: $\delta I \sim \Delta x^2 \sim n^{-2}$

Monte Carlo approach: $\delta I \sim n^{-1/2}$ – error is purely statistical

For one-dimensional integrals, Monte Carlo error vanishes slower with increasing number of points, n , and methodical approach is more efficient.

As dimension of the integral, d , increases, Monte Carlo becomes more efficient.

Methodical approach: $\delta I \sim \Delta x^2 \sim n^{-2/d}$ e.g., in 3D to have the same Δx as in 1D we need n^3 points.

Monte Carlo approach: $\delta I \sim n^{-1/2}$ Monte Carlo “wins” at $d > 4$

Why are we interested in integration?

Monte Carlo simulation = evaluation of statistical-mechanics integrals.

Ensemble average of quantity $A(\mathbf{r}, \mathbf{p})$ can be calculated for a given distribution function. For NVT ensemble we have distribution function $\rho(\mathbf{r}, \mathbf{p})$,

$$\rho(\vec{r}^N, \vec{p}^N) = \frac{1}{Z} \exp\left(-\frac{E(\vec{r}^N, \vec{p}^N)}{kT}\right) d\vec{r}^N d\vec{p}^N$$

$$\langle A(\vec{r}^N, \vec{p}^N) \rangle = \frac{1}{Z} \int A(\vec{r}^N, \vec{p}^N) \exp\left(-\frac{E(\vec{r}^N, \vec{p}^N)}{kT}\right) d\vec{r}^N d\vec{p}^N$$

Energy can always be expressed as a sum of kinetic and potential contributions. The contribution of the kinetic part is trivial and we can consider integral in only configurational $3N$ dimensional space, where Z is configurational integral.

$$\langle A(\vec{r}^N) \rangle = \frac{1}{Z} \int A(\vec{r}^N) \exp\left(-\frac{U(\vec{r}^N)}{kT}\right) d\vec{r}^N \quad Z = \int \exp\left(-\frac{U(\vec{r}^N)}{kT}\right) d\vec{r}^N$$



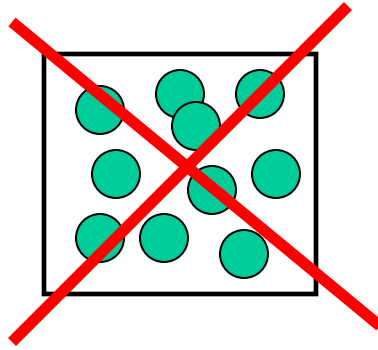
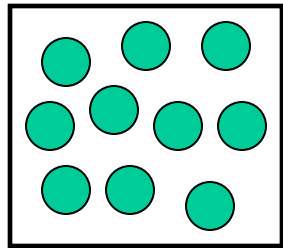
Monte Carlo simulation

(evaluation of statistical-mechanics integrals)

$$\langle A(\vec{r}^N) \rangle = \frac{1}{Z} \int A(\vec{r}^N) \exp\left(-\frac{U(\vec{r}^N)}{kT}\right) d\vec{r}^N \quad Z = \int \exp\left(-\frac{U(\vec{r}^N)}{kT}\right) d\vec{r}^N$$

Statistical-mechanics integrals typically have significant contributions only from very small fractions of the $3N$ space. For example for hard spheres contributions are coming from the areas of the configurational space where there are no spheres that overlap:

$$\exp\left(-\frac{U(\vec{r}^N)}{kT}\right) \neq 0$$



Random sampling of the configurational space is highly inefficient.

We have to restrict the sampling to the areas of space contributing to the integral
– *concept of importance sampling.*

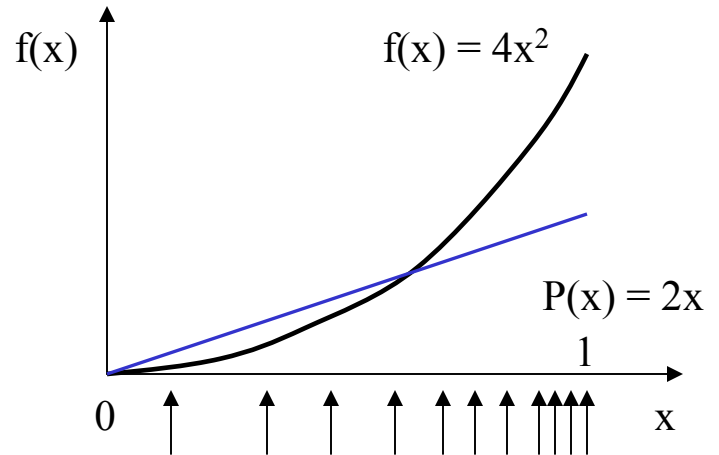


Importance sampling

We want to have more points in the region from where integral is getting its greatest contributions.

$$I = \int_0^1 4x^2 dx$$

Contribution to the integral is coming from the region near $x = 1$. We want to have more points there!



non-uniformly separated points

We can go from even distribution of random points to a non-uniform distribution, for example linear dependence from x : $P(x) = 2x$.

To evaluate the importance-sampled integral let us consider rectangle-rule quadrature with unevenly spaced points/measurements.

$$I \approx \sum_{i=1}^n f(x_i) \Delta x_i$$

$$\Delta x_i = \frac{b-a}{n} \frac{1}{P(x_i)}$$

where Δx_i is spacing between points.

$$I \approx \frac{b-a}{n} \sum_{i=1}^n \frac{f(x_i)}{P(x_i)} \quad \text{where points } x_i \text{ are chosen according to } P(x_i).$$



Importance Sampling: Metropolis Monte Carlo

$$\langle A(\vec{r}^N) \rangle = \int A(\vec{r}^N) \underbrace{\frac{\exp\left(-\frac{U(\vec{r}^N)}{kT}\right)}{Z}}_{P(\vec{r}^N)} d\vec{r}^N \quad Z = \int \exp\left(-\frac{U(\vec{r}^N)}{kT}\right) d\vec{r}^N$$

We can use importance sampling Monte Carlo to calculate ensemble average of quantity A: Average over measurements of A for configurations generated according to distribution $P(\mathbf{r}^N)$.

To generate configurations according to the desired distribution $P(\mathbf{r}^N)$ we can create random walk in the phase space, sampling it with the ensemble distribution. This can be realized in different ways. The approach that is used in famous Metropolis Monte Carlo algorithm uses random walk in the phase space with transition probability to go from state \mathbf{m} to state \mathbf{n} equal to $\mathbf{1}$ if the move is downhill in energy ($\Delta U_{nm} < \mathbf{0}$). If the move is uphill in energy ($\Delta U_{nm} > \mathbf{0}$) than the move is accepted with a probability defined by **the ratio of probabilities of initial and final states**:

$$\frac{\rho_n}{\rho_m} = \frac{\frac{1}{Z} \exp\left(-\frac{U_n}{kT}\right)}{\frac{1}{Z} \exp\left(-\frac{U_m}{kT}\right)} = \exp\left(-\frac{U_n - U_m}{kT}\right) = \exp\left(-\frac{U_{nm}}{kT}\right)$$

For a detailed discussion see Allen & Tildesley, pp.115-131



Detailed Balance & Metropolis Monte Carlo (I)

Let us set up a random walk through the configurational space (so-called Markov chain of configurations) by the introduction of a fictitious dynamics. The "time" t is a computer time (marking the number of iterations of the procedure), not real time - our statistical system is considered to be in equilibrium, and thus time is irrelevant.

$P(m,t)$ is the probability of being in configuration m at time t , $P(n,t)$ the probability of being in configuration n at time t , $W(m \rightarrow n, t)$ is the probability of going from state m to state n per unit time (transition probability). Then we have

$$P(m, t+1) = P(m, t) + \sum_n [W(n \rightarrow m)P(n, t) - W(m \rightarrow n)P(m, t)]$$

At large t , once the arbitrary initial configuration is "forgotten," want $P(m,t)$ to be $P(m)$. Clearly a sufficient (but not necessary) condition for an equilibrium (time independent) probability distribution is the so-called **detailed balance condition**:

$$W(n \rightarrow m)P(n, t) = W(m \rightarrow n)P(m, t)$$

This can be applied to any probability distribution, but if we choose the Boltzmann distribution we have

$$\frac{W(m \rightarrow n)}{W(n \rightarrow m)} = \frac{P(n)}{P(m)} = \frac{\frac{1}{Z} \exp\left(-\frac{U_n}{kT}\right)}{\frac{1}{Z} \exp\left(-\frac{U_m}{kT}\right)} = \exp\left(-\frac{U_{nm}}{kT}\right)$$

$$\text{where } U_{nm} = U_n - U_m$$

Z does not appear in this expression, and it only involves quantities that we know, T , or can easily calculate, U .



Detailed Balance & Metropolis Monte Carlo (II)

There are many possible choices of the \mathbf{W} which will satisfy detailed balance. Each choice would provide a dynamic method of generating an arbitrary probability distribution. Let us make sure that Metropolis algorithm satisfies the detailed balance condition.

$$W(m \rightarrow n) = \exp\left(-\frac{U_{nm}}{kT}\right) \quad (U_{nm} > 0)$$
$$W(m \rightarrow n) = 1 \quad (U_{nm} \leq 0)$$

if $U(n) > U(m)$

$$\frac{W(m \rightarrow n)}{W(n \rightarrow m)} = \frac{\exp\left(-\frac{U_{nm}}{kT}\right)}{1} = \exp\left(-\frac{U_{nm}}{kT}\right)$$

if $U(n) < U(m)$

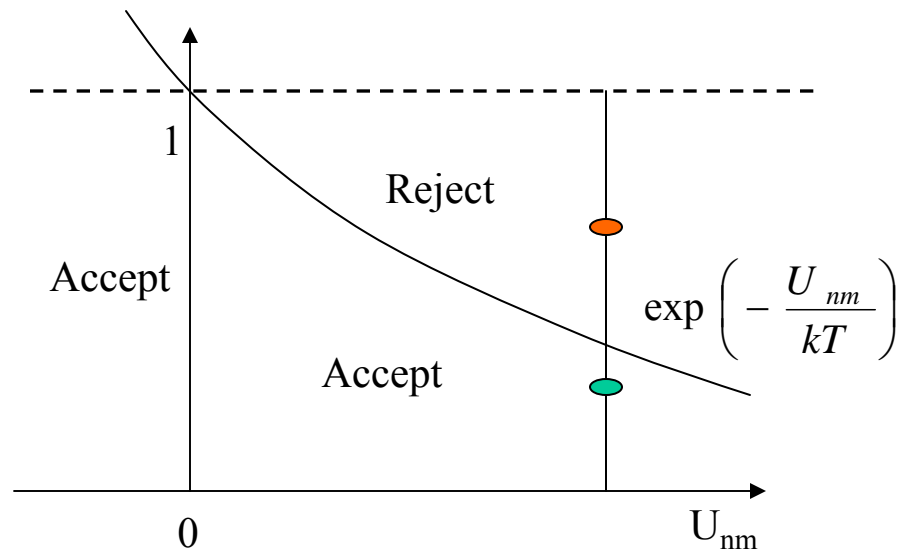
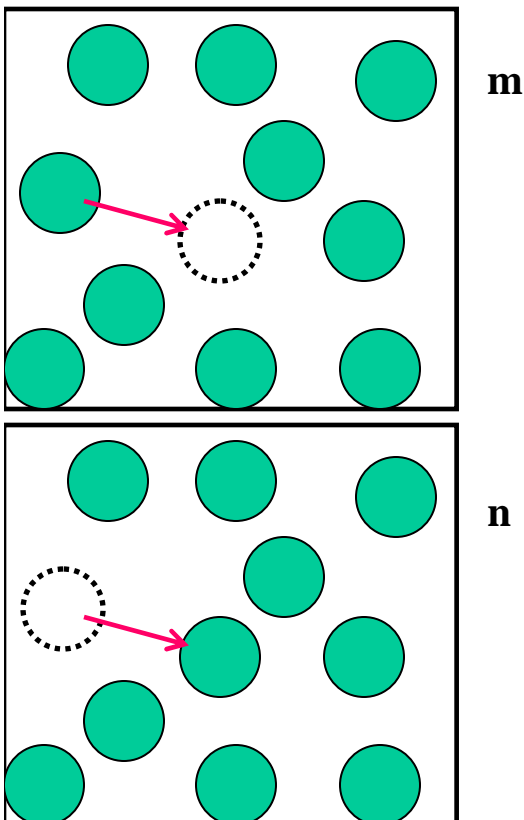
$$\frac{W(m \rightarrow n)}{W(n \rightarrow m)} = \frac{1}{\exp\left(-\frac{U_{mn}}{kT}\right)} = \exp\left(-\frac{U_{nm}}{kT}\right)$$

Thus, Metropolis Monte Carlo algorithm generate a new configuration n from a previous configuration m so that the transition probability $W(m \rightarrow n)$ satisfies the detailed balance condition.



Metropolis Monte Carlo algorithm

1. Choose the initial configuration, calculate energy
2. Make a “move” (for example, pick the random displacement). Calculate the energy for new “trail” configuration.
3. Decide whether to accept the move:
 - if $U_{nm} = U_n - U_m < 0$, then accept the new configuration,
 - if $U_{nm} = U_n - U_m > 0$, then
 - calculate $W(m \rightarrow n) = \exp\left(-\frac{U_{nm}}{kT}\right)$
 - draw a random number R from 0 to 1.
 - if $W(m \rightarrow n) > R$ then accept the new configuration,
 - otherwise, stay at the same place.
5. Repeat from step 2, accumulating sums for averages (if atom is retained at its old position, the old configuration is recounted as a new state in the random walk).





Metropolis Monte Carlo algorithm: implementation

1. Choose the initial configuration of the system, calculate energy
2. Loop through all the particles

For each particle pick a random displacement $d = (\text{random \#} - 0.5) * d_{\text{max}}$ for x, y and z coordinates. Here d_{max} is the maximum displacement, and random # is from 0 to 1.

Calculate the energy change ΔU due to the displacement.

Decide whether to accept the move based on the Metropolis criterion:

if $\Delta U < 0$, then accept the new configuration,

if $\Delta U > 0$, then calculate $W = \exp\left(-\frac{\Delta U}{kT}\right)$

draw a random number R from 0 to 1

if $W > R$ then accept the new configuration, otherwise, keep the old one

Accumulate sums for averages (if atom is retained at its old position, the old configuration is recounted as a new state).

Pick the next particle...

3. If # of MC cycles is $<$ then maximum # of cycles, Go back to step 2.

Choosing the maximum allowed displacement (size of the trail move) d_{max} :

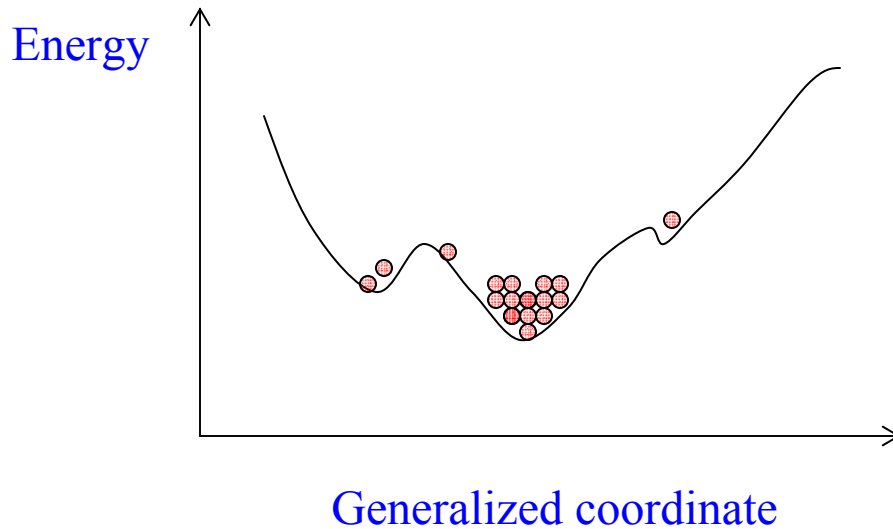
if d_{max} is too small then a large fraction of moves is accepted but the phase space is sampled slowly (consecutive states are highly correlated).

if d_{max} is too large then most of the trail moves is rejected and there is little movement through the phase space.

Usually d_{max} is chosen so that $\sim 50\%$ of the trail moves is accepted.



Metropolis Monte Carlo algorithm: method for finding the equilibrium state



What if we start our simulation away from the equilibrium (in a region of the configurational space that is unlikely to be sampled at equilibrium)?

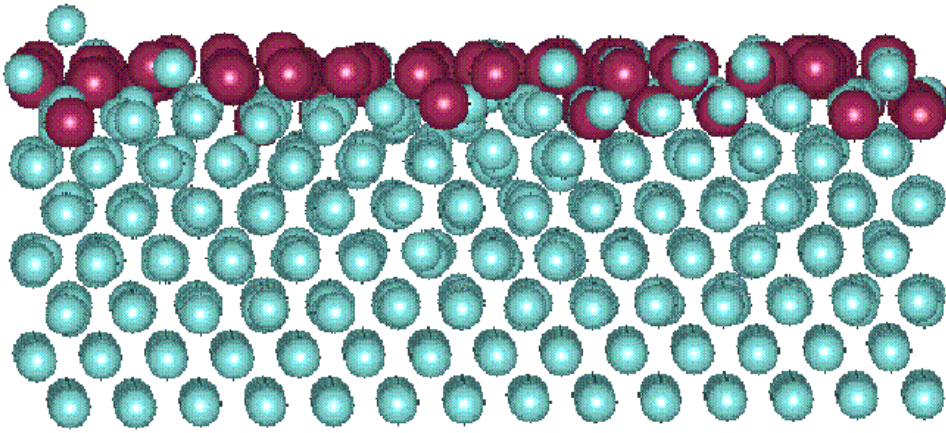
The random walk in MC should bring us to the equilibrium.

We can use this method to find equilibrium structure and composition!



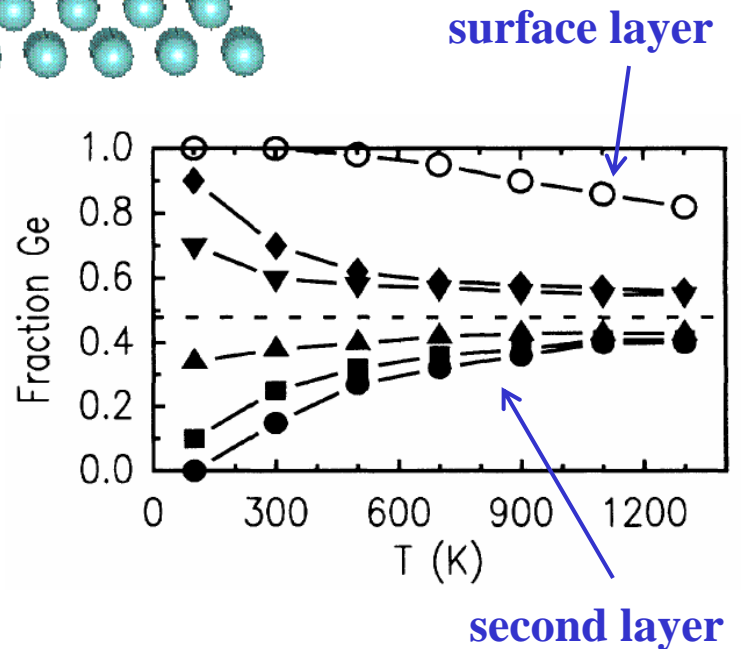
Metropolis Monte Carlo algorithm: Examples

Oscillatory surface segregation of Si-Ge alloys



Kelires & Tersoff,
Phys. Rev. Lett. **63**, 1164, 1989

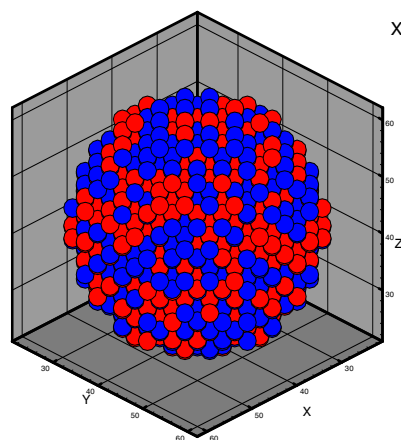
Two types of Monte Carlo moves
– small atomic displacements and
interchange of atom type are used
to investigate surface segregation
vs temperature in Si-Ge alloys.



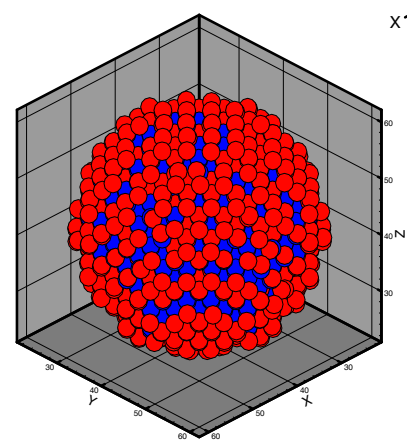
Compositional ordering in
Si-Ge clusters.

Metropolis MC simulation
by Adam DiNardo

Si - blue, Ge - red



initial



final

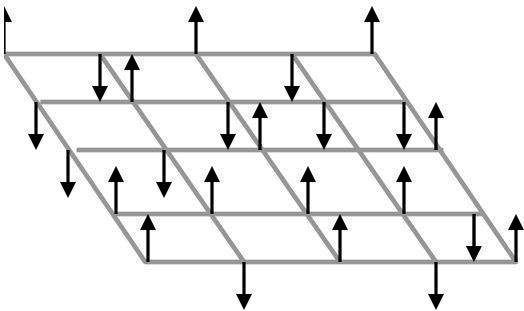


Lattice Monte Carlo. Ising Model.

Models in which particles belong to the lattice sites and interact locally can be a good representation of a number of real systems (magnetic materials, binary alloys, etc.). There are several modifications of lattice MC:

In the original Ising model the internal energy of a system (e.g. magnetic material) is calculated as the sum of pair interaction energies between the atoms or molecules, which are attached to the nodes of a regular lattice. In terms of **magnetic material**, we have a discrete set of degrees of freedom interacting with each other and an external magnetic field. The spin variable S_i is defined at each lattice site i and can have one of two states: “spin up”, $S_i = +1$ and “spin down”, $S_i = -1$. For classical description of ferromagnetic ordering, the energy can be written as the sum of spin-spin interaction energy and interaction of each spin with an external magnetic field H :

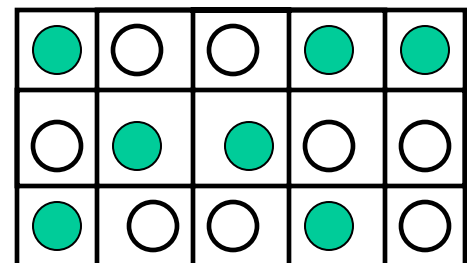
$$E = -J \sum_{(ij)} S_i S_j - H \sum_i S_i$$



where J is the effective interaction energy ($J > 0$ for ferromagnetic, $J < 0$ for antiferromagnetic). Average magnetization $m = \sum S_i / N$ is zero at $T > T_c$, and has some non-zero value $m(T)$ at $T < T_c$, where T_c is the Curie temperature.

For a binary alloy, one can consider “spin up” = “A atom,” “spin down” = “B atom”, and “the magnetic field H ” = “the chemical potential difference $\Delta\mu$ ” Energy of interaction ϵ_{12} is 0 for AA and BB and ϵ for AB. That is

$\epsilon_{12} = \epsilon (1 - S_1 S_2) / 2$. Difference between the numbers of particles $N_A - N_B = \sum S_i$ One can perform simulation (MC moves = switch particle types) to find equilibrium composition.





Monte Carlo algorithm for Ising model

1. Choose the initial configuration of the system, calculate energy

2. Loop through the lattice sites

Pick the lattice site i to swap.

Calculate the energy change ΔU due to the swapping ($S_i = -S_i$).

Decide whether to accept the swap based on the Metropolis criterion:

if $\Delta U < 0$, then accept the new configuration,

if $\Delta U > 0$, then calculate $W = \exp\left(-\frac{\Delta U}{kT}\right)$

draw a random number R from 0 to 1

if $W > R$ then accept the swap, otherwise, keep the old configuration

Accumulate sums for average magnetization (magnetic spin system) or composition (binary alloy).

Go to the next lattice site...

3. If # of MC cycles is $<$ then maximum # of cycles, Go back to step 2.

Multistate Potts lattice model: Ising spin model has been generalized to the kinetic multistate Potts lattice model. Potts model replaces the Boolean spin variable S_i with only two states (up & down) by a generalized variable S_i that can assume one out of a discrete set of possible states. Only dissimilar neighbors contribute to the interaction energy:

$$E = -J \sum_{ij} (\delta_{S_i S_j} - 1), \quad S_i = 1, 2, \dots, q$$

The delta-function is 1 for pairs of neighboring sites with identical spins and corresponding interaction energy is zero.

The spectrum of possible spins allows one to represent many different domains, for example different crystal orientation. Each orientation can be associated with state variables such as lattice energy, surface energy, dislocation density, etc. For example, Potts model is often used in simulations of microstructure development.



Metropolis Monte Carlo simulation: summary

MC simulation gives properties via ensemble averaging. Random numbers are used to generate large number of states of the system (members of an ensemble) with distribution characteristic for the ensemble.

Metropolis Monte Carlo – a method to generate a **sequence of random states** of the system with probability distribution characteristic for the ensemble.

Simulations discussed until now are performed at **constant T without any special efforts** (different from MD). No need to ensure that energy is conserved after a trail move.

MC can be easily adapted to the calculation of averages in any ensemble.

For example, simulation of isothermal – isobaric ensemble involves random volume change (volume is added to the set of independent variables that characterize the state of the system); simulation of grand-canonical ensemble have “moves” that insert/delete atoms/molecules.

Trail moves: We did not make any assumptions regarding the nature of the trail moves. Trails must ensure that all relevant elements of ensemble are sampled. Depending on the nature of the simulated system we can have atom displacements, molecule displacements and rotations... Significant increase in efficiency of algorithm can be achieved by the introduction of clever trail moves.

Any given probability distribution can be sampled in different ways, using different strategies to sample configurational space.

MC cannot address kinetics or dynamical properties (diffusion, mean square displacements, velocity autocorrelation functions, vibrational spectra etc.)