Computational Physics – Lecture 6: Monte Carlo methods I

Kristel Michielsen
Institute for Advanced Simulation
Jülich Supercomputing Centre
Forschungszentrum Jülich
k.michielsen@fz-juelich.de
http://www.fz-juelich.de/ias/jsc/qip

Contents

- Monte Carlo method
  - Definition
  - Application
- Simple sampling Monte Carlo methods
  - Estimation of the value of π
  - Numerical integration
    - Rectangular rule
    - Midpoint rule
    - Trapezoidal rule
    - Simpson’s rule
  - Monte Carlo integration
    - Hit-or-miss method
    - Crude (sample-mean) method
    - Statistical errors
  - Simulation of radioactive decay
  - Percolation
  - Ground state finding of an Hamiltonian
    - Ising model

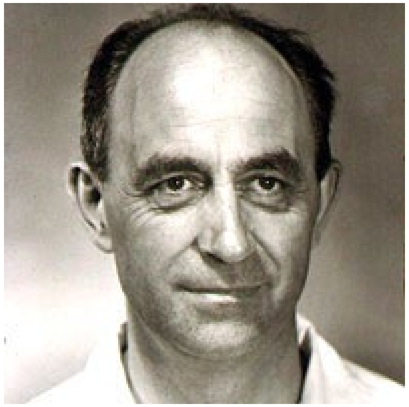
Some pictures are taken from the internet for illustrational and educational purposes

Monte Carlo method

- Broad class of computational algorithms that rely on repeated random sampling to obtain numerical results
- Name refers to gambling casino in Monte Carlo (Monaco), because of the resemblance of the technique to the act of playing and recording results in a gambling game

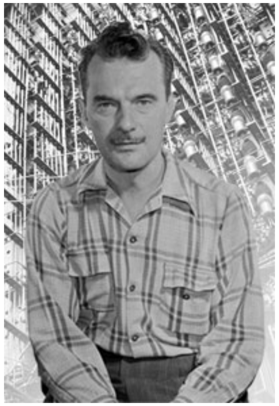
Monte Carlo method

- Designers of the modern Monte Carlo method in the late 1940’s are Enrico Fermi, Stanislav Ulam, John von Neumann, Nicholas Metropolis, and others [N.G. Cooper (ed.), *From Cardinals to Chaos*, (Cambridge University Press, Cambridge 1989), N. Metropolis, "The beginning of the Monte Carlo method“, Los Alamos Science (1987 Special Issue dedicated to Stanislaw Ulam), 125–130]


Enrico Fermi
1901 - 1954

John von Neumann
1903 - 1957

Stanislav Ulam
1909 - 1984


Nicholas Metropolis
1915 - 1999

Monte Carlo method

- Principle:
  - Consider a domain X of possible events x and a distribution $p(x)$ over it
  - Generate events x randomly from $p(x)$
  - Perform a deterministic computation (evaluation) on the events
  - Accumulate the results
- Characteristics:
  - Stochastic evolution, depending on the sequence of random numbers that has been generated
  - Statistical results: A second, different sequence of random numbers will not give the same results as obtained by a first sequence of random numbers but will give results which agree with those obtained from the first sequence of random numbers within some statistical error.

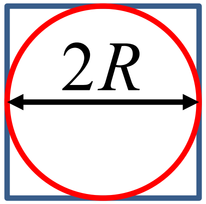
Monte Carlo method

- Stochastic method which is useful when it is difficult or impossible to obtain a closed-form expression, or difficult to apply a deterministic algorithm
- Applications in physics:
  - Simple sampling Monte Carlo methods
    - Numerical integration
    - Percolation
    - Fluid flow
    - …
  - Importance sampling Monte Carlo methods
    - (Quantum) statistical physics

Simple sampling Monte Carlo:
Estimation of the value of π

- Draw a square with an inscribed circle

$$\frac{A_c}{A_s} = \frac{\pi R^2}{(2R)^2} = \frac{\pi}{4} \Rightarrow \pi = 4\frac{A_c}{A_s}$$



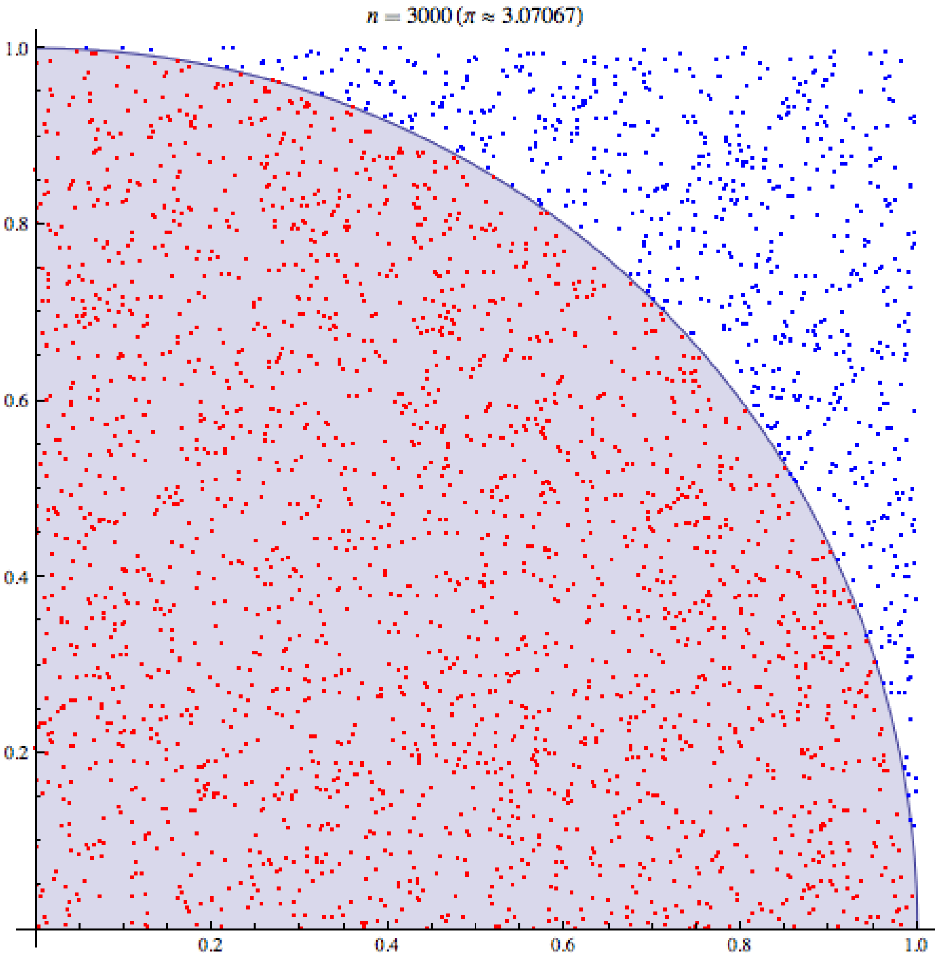
- Select uniformly distributed random points within the square
- Count the number of points N_c lying within the circle (distance between point and center of circle is $\leq R$) and the total number of points N
- To compute π use: $\pi = 4\frac{A_c}{A_s} \approx 4\frac{N_c}{N}$

N has to be sufficiently large!

Simple sampling Monte Carlo:

Estimation of the value of π

Hit-or-miss Monte Carlo



n =30000 random points → estimate for π is within 0.07% of the actual value.

Caithin Jo Ramsey, Wikipedia

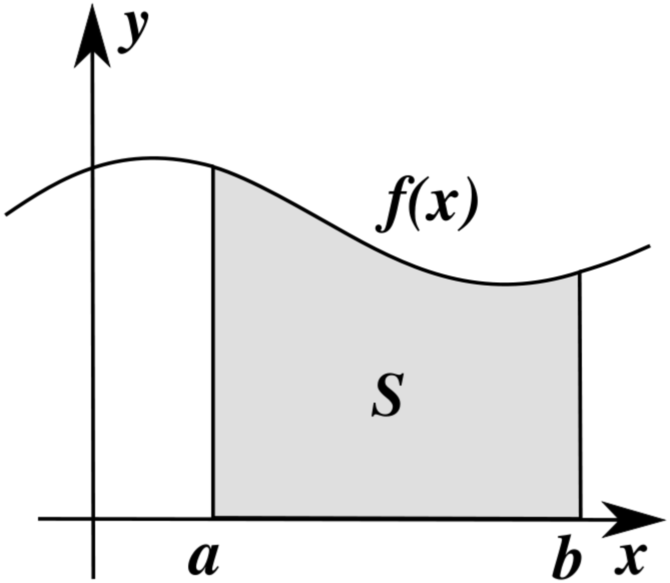
Simple sampling Monte Carlo:
Numerical and Monte Carlo integration

- Numerical (deterministic) and Monte Carlo (stochastic) integration: evaluate integrals that cannot be solved analytically
- First consider elementary algorithms for one-dimensional (1D) integrals of the form

  $I = \int_a^b f(x)dx$ where $f(x)$ can be evaluated at all points in $[a,b]$

Simple sampling Monte Carlo:
Numerical and Monte Carlo integration

- Task is to compute an approximate solution to the definite integral to a given degree of accuracy
→ find numerical approximations for the value S



Simple sampling Monte Carlo:
Numerical integration

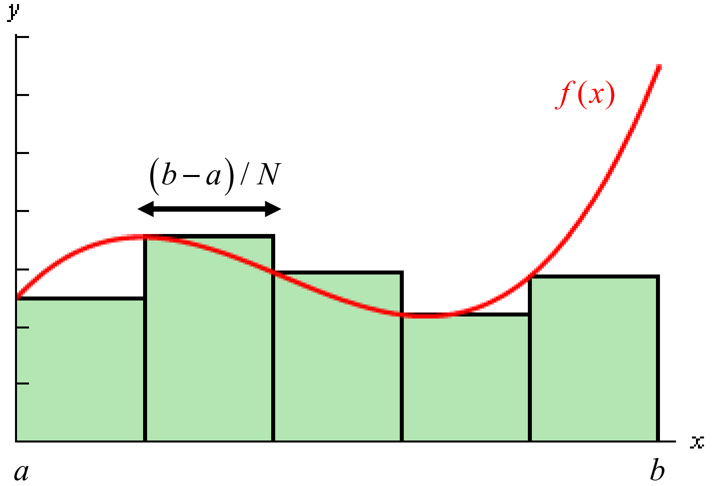
- Divide the domain $b-a$ in N equally sized intervals $(b-a)/N$ and apply the composite
  - Discrete sum (rectangular) rule

$$I \approx I_N = \frac{b-a}{N} \sum_{n=0}^{N-1} f\left(a + n\frac{b-a}{N}\right) + O\left(\frac{1}{N}\right)$$

$$I = \lim_{N \to \infty} I_N$$

- Error: $O(N^{-1})$
- # function evaluations:

  $M = N$



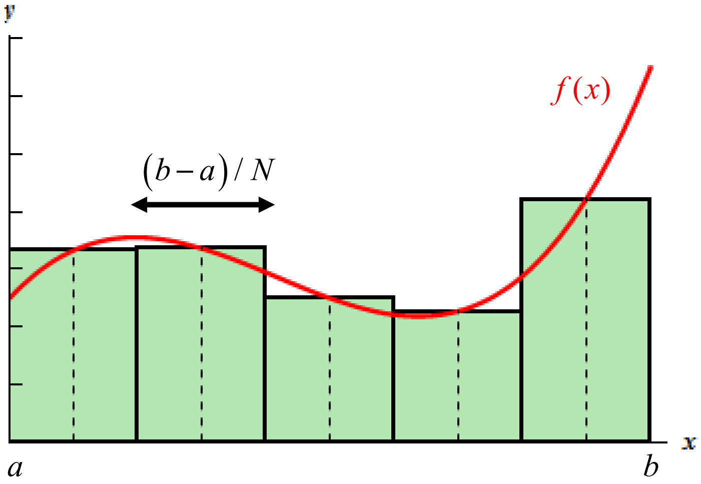
Simple sampling Monte Carlo: Numerical integration

– Midpoint rule

$$I \approx I_N = \frac{b-a}{N}\sum_{n=0}^{N-1} f\left(a + \frac{2n+1}{2}\frac{b-a}{N}\right) + O\left(\frac{1}{N^2}\right)$$

$$I = \lim_{N\to\infty} I_N$$

- Error: $O(N^{-2})$
- # function evaluations: $M = N$



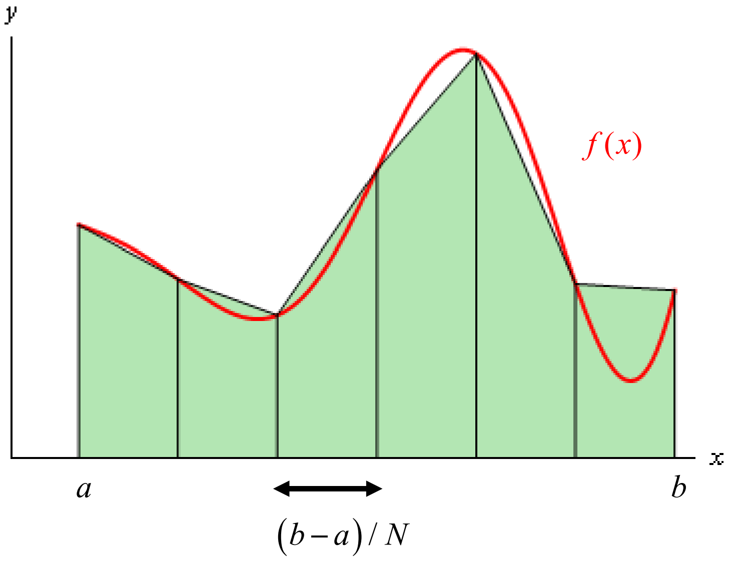
Simple sampling Monte Carlo: Numerical integration

– Trapezoidal rule

$$I \approx I_N = \frac{b-a}{N}\left(\frac{1}{2}f(a)+\sum_{n=1}^{N-1} f\left(a+n\frac{b-a}{N}\right)+\frac{1}{2}f(b)\right)+O\left(\frac{1}{N^2}\right)$$

$$I = \lim_{N\to\infty} I_N$$

- Error: $O(N^{-2})$
- # function evaluations: $M = N+1$



Simple sampling Monte Carlo: Numerical integration

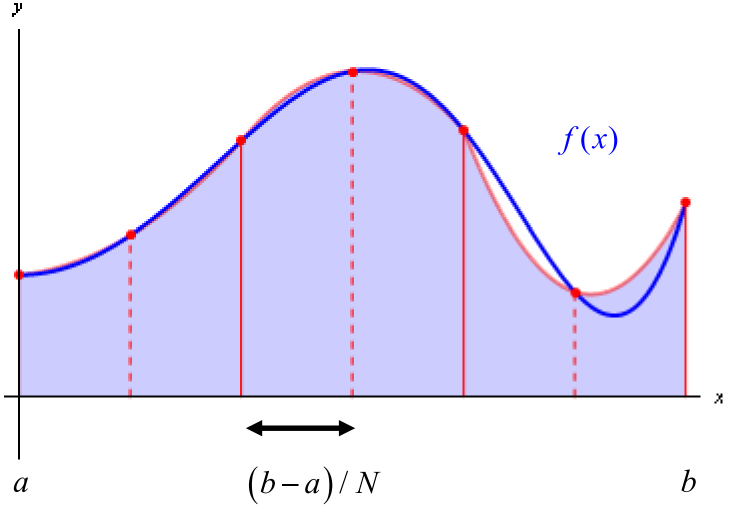
– Simpson's rule

$$I \approx I_N = \frac{b-a}{3N}\left(f(a) + \sum_{n=1}^{N-1} \left(3-(-1)^n\right) f\left(a + n\frac{b-a}{N}\right) + f(b)\right) + O\left(\frac{1}{N^4}\right)$$

with N even

$$I = \lim_{N\to\infty} I_N$$

- Error: $O(N^{-4})$
- # function evaluations: $M = N + 1$



Simple sampling Monte Carlo: Numerical integration

- What about d dimensional integrals?
  - Algorithms for multi-dimensional integrals can be carried out "dimension-by-dimension" using one of the 1D formula's
  - Nested numerical integration, e.g. $d=2$

$$I=\int_a^b dx_1\int_c^d dx_2 f\left(x_1,x_2\right)\rightarrow I_N=\int_a^b dx_1 F\left(x_1\right)$$

    - Evaluate the inner integral using one of the integration rules to obtain a function $F\left(x_1\right)$, then integrate this function over x_1 using the same integration rule

Simple sampling Monte Carlo: Numerical integration

- Overall error: same as for 1D integral, but # function evaluations: $M \approx N^d$

→Error in nested d-dimensional

- – Trapezoidal rule: $O(N^{-2}) = O(M^{-2/d})$
- – Simpson's rule: $O(N^{-4}) = O(M^{-4/d})$

→the number of function evaluations needed to achieve a given tolerance grows exponentially with d ("curse of dimensionality"): an order $-n$ method in 1 dimension is an order $-n/d$ method in d dimensions

- E.g. statistical mechanics: model with N particles has $6N$ integrals ($3N$ position + $3N$ momentum)

Extremely inefficient!

Simple sampling Monte Carlo: Monte Carlo integration

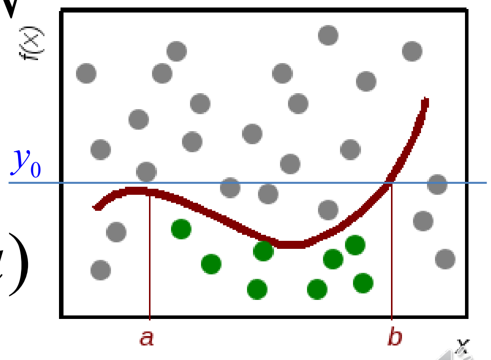
- Evaluation of definite integrals using random numbers
- Particularly useful for higher dimensional integrals
- Simplest case: Determine the integral of $f(x)$ over some fixed interval

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

Simple sampling Monte Carlo: Monte Carlo integration

- Hit-or-miss (acceptance-rejection) method:
  - Draw a box extending from a to b and from 0 to y_0 where $y_0 > f(x)$ throughout this interval
  - Generate uniformly distributed random numbers within the box
  - Count the number of points N_0 lying below the curve $f(x)$ and the total number of points N
  - Calculate

$$\frac{I}{A_{\text{box}}} = \frac{I}{y_0(b-a)} \approx \frac{N_0}{N} \Rightarrow I \approx \frac{N_0}{N} y_0(b-a)$$

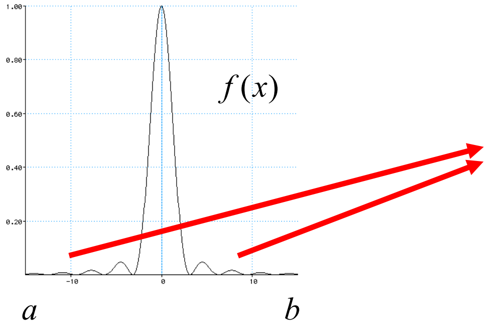


Simple sampling Monte Carlo: Monte Carlo integration

- Crude (sample-mean) method:
  - Choose N values of x randomly and evaluate $f(x)$ at each value so that

$$I \approx I_N = \frac{(b-a)}{N} \sum_{n=1}^{N} f(x_n)$$

- Simple sampling Monte Carlo integration is not so efficient for sharply peaked functions



In these two regions sampling becomes time inefficient

Simple sampling Monte Carlo: Statistical error

- Monte Carlo data are reported in the form

$$E\left[X\right]=\overline{x}\pm\frac{s}{\sqrt{N}}$$

- Graphically:
  - $\overline{x}$ → symbol (e.g. circle)
  - $\frac{s}{\sqrt{N}}$ → bar of length $s/2\sqrt{N}$ with $\overline{x}$ in the centre of the "error bar"

Simple sampling Monte Carlo: Numerical integration

- Error: $O\left(N^{-1/2}\right)$ independent of d ! (central limit theorem)

$$\left|I - I_N\right| \propto \frac{1}{\sqrt{N}} \qquad I = \lim_{N\to\infty} I_N$$

Statistical error

→If d is small, then Monte Carlo has much larger integration errors than numerical integration

→Monte Carlo is as good as Simpson's rule for $d = 8$

- Note: MC sampling of sharply peaked functions may have a large sampling error since the variance may be larger

Simple sampling Monte Carlo: Simulation of radioactive decay

- Consider a sample of N nuclei with decay rate $dN/dt = -\lambda N$ where λ is the decay constant.
- The resulting time dependence of the number of undecayed nuclei is $N = N_0 e^{-\lambda t}$, where N_0 is the initial number of nuclei.
- Monte Carlo approach:
  - For a given nucleus, the probability P that a nucleus decays in time dt is $P = \lambda dt$ → calculate for each nucleus whether it decays or not for some small dt

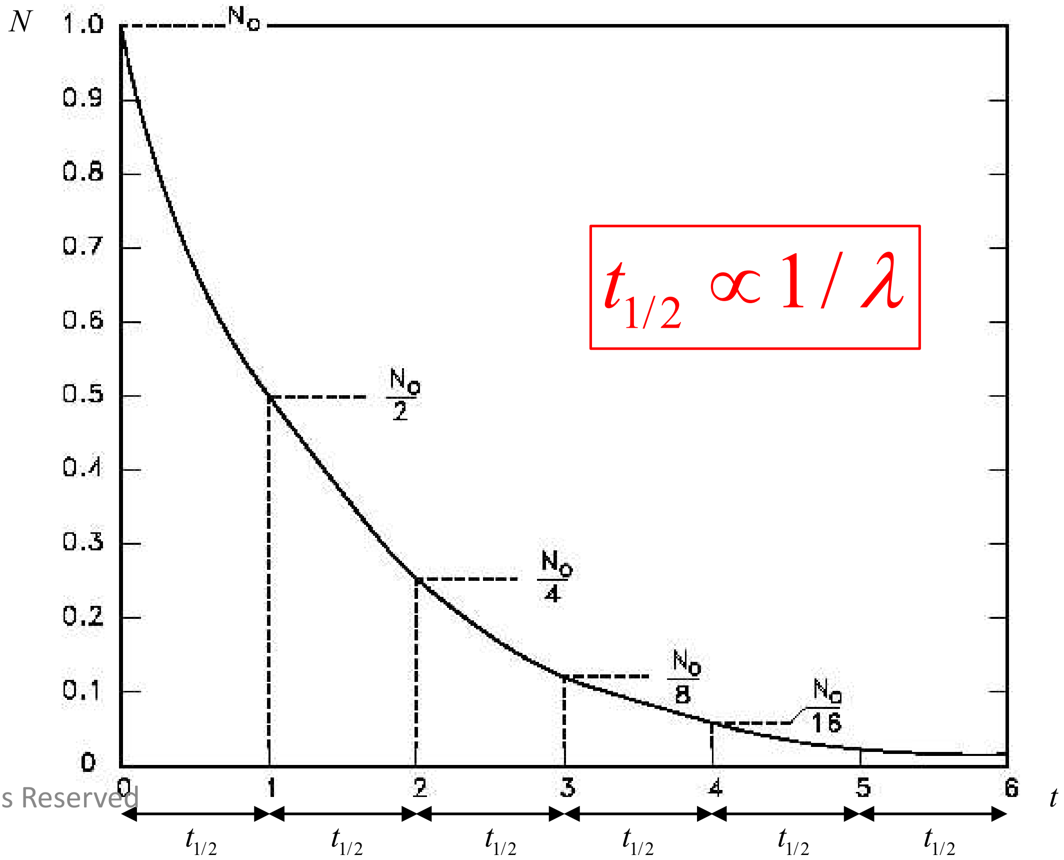
Simple sampling Monte Carlo:
Simulation of radioactive decay

- Pseudocode:

```
Input: $N_0, \lambda, dt, M$ ( $M$: number of time steps)
Loop1: $i = 1, M$      (loop over time)
$N_i = N_{i-1}$
Loop2: $j = 1, N_i$      (loop over undecayed nuclei)
Generate a random number $0 \le r \le 1$
If $r < \lambda * dt$ then $N_i = N_i - 1$
End loop2
Write $i * dt$ and $N_i$ to a file
End loop1
Plot $N_k$ versus $k * dt$ for $k = 0, M$
Plot $N = N_0 * \exp(-\lambda * dt)$
```

Simple sampling Monte Carlo:
Simulation of radioactive decay

- Half-life $t_{1/2}$: time taken for the radioactive activity of the sample to reduce by a factor of two



$$\boxed{t_{1/2} \propto 1/\lambda}$$

Simple sampling Monte Carlo: Simulation of radioactive decay

- Remarks:
  - Time discretization must be such that a reasonable number of decays occur in each time step
    - if dt/M is too small too much CPU time is required
    - if dt/M is too large many decays occur during a given time interval and there is very poor time resolution
  - Whole process may be repeated many times to get a series of independent experiments
    - Mean value of N and an error bar may be determined for each value of time.
    - Since each experiment is independent of the others, measurements for each value of time are uncorrelated even though there may be correlations between different times for a single experiment.

Simple sampling Monte Carlo: Percolation

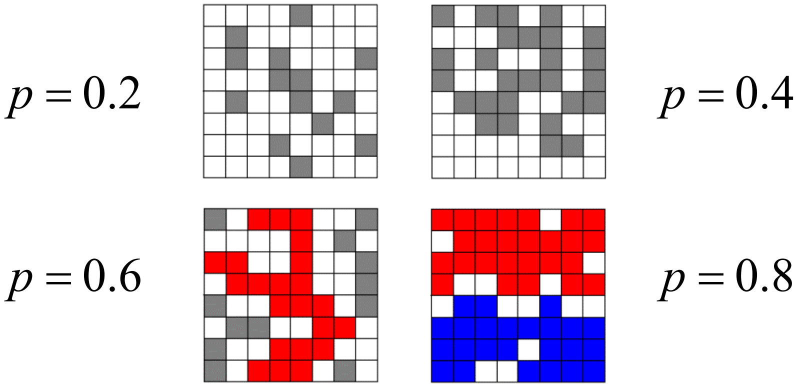
D. Staufer, Introduction to percolation Theory (Taylor and Francis, London, 1985)

- Introduction
  - Geometric problem which has played a significant role in statistical mechanics
  - Percolation processes: processes in which objects are added randomly until an unbroken path spans the entire system (or vice versa)
- Percolation model: Site percolation
  - Consider a 2D square lattice (simplest kind of percolation problem)
  - Initially the lattice is empty

Simple sampling Monte Carlo: Percolation

- Sites are randomly occupied with probability p
- Clusters of occupied nearest neighbor sites are formed
  - Smallest possible cluster = 1 site
- Percolation occurs when a cluster extends from one side of the system to the other one

  → spanning or infinite cluster

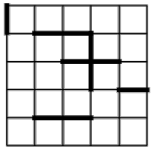


Simple sampling Monte Carlo: Percolation

- Critical probability
  - For each value of p the probability P_{span} of having a spanning cluster can be determined by generating many realizations of the lattice and counting the number of times a spanning cluster appears
  - In the limit of an infinite lattice there exists a critical probability p_c such that
    - For $p < p_c$, $P_{\text{span}} = 0$
    - For $p > p_c$, $P_{\text{span}} = 1$

2D square lattice: $p_c = 0.59$

- Bond percolation: bonds are randomly occupied with probability p



2D square lattice: $p_c = 0.5$

Simple sampling Monte Carlo: Percolation

- Percolation transition
  - Define $M(p)$: fraction of occupied sites in the lattice which belong to the spanning cluster.
  - Small p (relatively sparsely occupied lattices): $M(p) \simeq 0$
  - $p \nearrow$: p eventually reaches $p = p_c$, the percolation threshold for which $M(p) > 0$
  - $p \nearrow \nearrow \rightarrow M(p) \nearrow$
  - Near the percolation threshold ($p \rightarrow p_c^-$): $M(p) \sim (p - p_c)^{\beta}$ with β the critical exponent. M is called the order parameter.

Simple sampling Monte Carlo: Percolation

  - Percolation transition = geometric phase transition
  - Geometric phase transitions have a close analogy to real phase transitions which occur in various physical systems

- Implementation of Monte Carlo method
  - Start with an empty square lattice with linear dimension L
  - Go through the lattice, site by site, and fill each site with probability p. Count the number of filled sites.

Simple sampling Monte Carlo: Percolation

- At the end, the actual concentration of filled sites might be different from p. Randomly fill or empty a few sites so that the desired value of p is obtained.
- Clusters are found by searching for nearest neighbor occupied sites.
- Test for percolation by checking if any of the clusters reaches from one side of the lattice to the opposite side.
- Repeat the whole process many times to get an estimate of the probability of finding a spanning cluster.

Simple sampling Monte Carlo: Percolation

– Cluster identification: Hoshen-Kopelman algorithm

(J. Hoshen and R. Kopelman, Phys. Rev. B 14,3438 (1976))

Example:

-1	-1	0	-1	-1	-1	0
-1	0	0	0	-1	-1	-1
0	-1	-1	0	0	-1	-1
-1	-1	-1	-1	-1	-1	-1
0	0	-1	0	-1	-1	-1
-1	-1	0	0	-1	-1	-1
-1	0	-1	-1	0	-1	0

7	7	0	9	3	3	0
7	0	0	0	8	3	3
0	6	3	0	0	3	3
6	6	5	5	3	3	3
0	0	5	0	4	3	3
1	1	0	0	4	3	3
1	0	2	2	0	3	0

$np(1)=1\,, np(2)=2\,, np(3)=3$

$np(4)=3\,, np(5)=3\,, np(6)=3$

$np(7)=7\,, np(8)=3\,, np(9)=3$

7	7		3	3	3	
7				3	3	3
	3	3			3	3
3	3	3	3	3	3	3
		3		3	3	3
1	1			3	3	3
1		2	2		3	

cluster 3 = spanning cluster

Simple sampling Monte Carlo: Percolation

- An (un)occupied site has initially value -1 (0)
- Start cluster labeling from left bottom corner and move in rows
- If a site is occupied, check the occupancy of its nearest neighbors in the previous row and column
    - neither neighbor occupied → assign the next available proper cluster label m and set $np(m) = m$
    - 1 nearest neighbor site occupied with label m_1 → proper cluster label $m = m_1$
    - 2 nearest neighbors occupied with labels m_1 and m_2 → proper cluster label $m = \min(np(m_1), np(m_2))$ and set $np(\max(np(m_1), np(m_2))) = m$

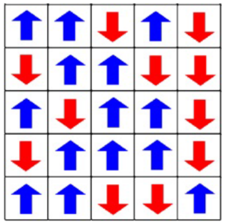
Simple sampling Monte Carlo: Percolation

    - Here it becomes clear that the array np distinguishes proper and improper labels and provides their connections
        - » Note that instantly relabeling of the occupied neighbor with the highest cluster value would be inefficient

- – At the end, relabel all improper cluster labels using the values in the array np
- – Spanning cluster if two sides of the lattice contain a site with the same cluster number
- – Efficient algorithm (although not the most general and efficient): it only sweeps once through the whole lattice

Simple sampling Monte Carlo:
Find the ground state of a Hamiltonian

- If the ground state (minimum energy state) of a system is not known, then a simple Monte Carlo simulation can be used to find states of low energy, and hopefully that of lowest energy.
- Example: Ising model



$$H = -J\sum_{<n,m>} S_n S_m - h\sum_{n=1}^{N} S_n \text{ where } S_n = \pm 1$$

  - Mathematical model of ferromagnetism in statistical physics
  - S_n represent magnetic dipole moments of atomic spins

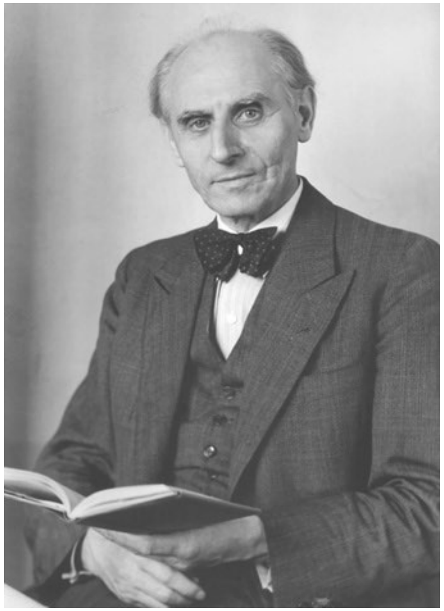
Simple sampling Monte Carlo: Ising model

- J : spin-spin interaction
- h : external magnetic field
- Minus signs in Hamiltonian: convention
- If, for all pairs i, j
  - $J_{i,j} > 0$: ferromagnetic interaction
  - $J_{i,j} < 0$: antiferromagnetic interaction
  - $J_{i,j} = 0$: the spins are noninteracting

  otherwise the system is called nonferromagnetic

Simple sampling Monte Carlo: Ising model

- Ising model allows the identification of phase transitions, as a simplified model of reality
- Invented by Wilhelm Lenz (1920), who gave it as a problem to his student Ernst Ising (1925)



Wilhelm Lenz
1888 - 1957

Ernst Ising
1900 - 1998

Simple sampling Monte Carlo: Find the ground state of a Hamiltonian

- Select a randomly chosen spin configuration
- Proceed through the lattice and determine the change in energy of the system if the spin is overturned
  - If the energy is lower, then turn the spin
  - If the energy is higher, then leave the spin unchanged
  - Move to the next lattice point
- Sweep through the system repeatedly until no spin-flip can be made anymore → the system is in the ground state or in a metastable state
- Repeat the process starting from different initial spin configurations to test whether a lower energy state can be found.
- For systems with complex energy landscapes there may be many energy minima of approximately the same depth and then a more sophisticated strategy will have to be chosen.

Simple sampling Monte Carlo method

→

Importance sampling Monte Carlo method