Random Number Generators

Of central importance for stochastic algorithms

- Monte Carlo
- Brownian Dynamics (Integration of Stochastic Differential Equations)

Actually: Pseudo-Random Numbers

i.e. generated through some <u>algorithm</u>

Typical example: take a <u>SEED</u>, for example an integer number and perform algebraic manipulations

<u>Consequence</u>: typically an integer word is 4 Byte = 32 bit

$\square\ \square \cdots \square\ \square$ → a total of $2^{31} - 1 \approx 2 \cdot 10^9$

sign 2^{30} $\quad\quad$ $2^1\ 2^0$

different numbers → <u>maximum theoretically possible</u>

~~Period~~ of a generator working with 32 Bit Integers!

<u>Remarks</u>: - In practice the period depends on the quality of the algorithm and often is much shorter

- even 10^9 is a small number
- using longer integers [for instance dround48()] does not help significantly
- there are more complex, so-called Very Long Period generators

To test the quality of a generator one has to address the question of correlations (also of higher order) between the generated random numbers.

We will assume that all discussed generators deliver uniformly distributed random numbers in $[0,1[$

Other distributions can be obtained by transformations of these random numbers.

1) Congruential generators [Lehmer 1951]

$$x_n = [a\, x_{n-1} + c] \bmod m$$

linear Congruential Generator $LCG(a; c; m)$

for $c \equiv 0$: multiplicative LCG: $MLCG(a; m)$

When one uses a built-in generator provided on an operating system it is usually of this kind.

Examples: i) $a = 65539$ $m = 2^{29}$: RANDU

supplied by IBM (quite some time ago)

created for smallest possible serial correlations (x_n, x_{n-1})

but has long higher order correlations

ii) $a = 1664525$ $m = 2^{32}$: INMOS Transputer development System

iii) $a = 5^{15}$ $m = 2^{47}$: CDC Computers (Control Data Corporation) ; Period $\approx 10^{13}$

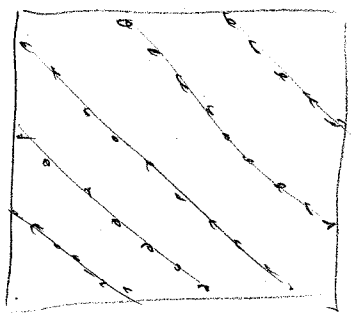
↳ classic generators but typically not used any longer : Why?

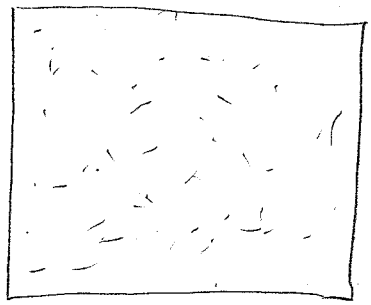
How does one measure the quality of a generator?

Obvious: Goodness of the generated distribution
↳ judge by moments and their errors
mostly easy to pass

Higher order serial correlations:

- Create a d-tuple of random numbers
- interpret this as a point in d-dimensional space
- all such generated points of most LCG's lie on a small number of hyperplanes





2] Linear Feedback Shift Register or Tausworth method
LFSR (p, q) generate sequence of binary digits

$$b_n = b_{n-p} \oplus b_{n-q}$$

↑ bitwise addition modulo 2

Use appropriate number of bit to define the next random integer.

⇒ Generalized feedback shift register GFSR($l_1, \dots, l_a$; m)

$$\omega_n = (\omega_{n-l_1} \oplus \dots \omega_{n-l_a}) \bmod m \quad ; \quad m = 2^d \; ; d \text{ length of word}$$

↑ computer word.

Special case (again from IBM laboratories): R 250

$$\omega_n = (\omega_{n-103} \otimes \omega_{n-250}) \bmod 2^d$$

↑ exclusive or

```
1 0 1 1 0 1 0 1
0 1 1 0 1 0 0 1
---------------
1 1 0 1 1 1 0 0
```

} 2 8 bit words

result of exclusive or

3] Lagged Fibonacci Generators

LF (p, q, m) : related to the GFSR

$$r_n = (r_{n-p} \odot r_{n-q}) \bmod m \qquad p > q$$

maximum period for right choice of p and m

$$(2^p - 1)(2^m - 1)$$

If $\odot$ = XOR → GFSR

Better: $\odot$ = "+" or "−" → r_n can be directly a real number between 0 and 1

is no need to divide by largest integer

advantage: division by largest integer: result depends on compiler and machine (how many bits are thrown away?)

real arithmetic with 24 bit mantissa in 4 Byte Real number gives the same result on all machines and in all languages

General point: 2] and 3] increase period by using but only one such but many

R250: works on array (vector) of 250 integers

RANMAR: works on array (vector) of 97 real numbers

period: 2^{144}

4] Variations of lagged Fibonacci generators:

add-and-carry or subtract-and-borrow

$$x_n = (x_{n-p} \pm x_{n-q} \pm c) \bmod b$$

$p > q$ are the lags, "c" is a carry bit

$c = 0$ if $x_{n-p} \pm x_{n-q} \leq b$

$c = 1$ (a one in the least significant bit position) if

$$x_{n-p} \pm x_{n-q} > 2$$

For $b = 2$: generation of random bits.

$b = 2^{24}$ generation of 32 bit floating point numbers with 24-bit mantissas.

Choice of p and q: number theory to ensure very long period

for $p = 24$ and $q = 10$ (i.e. use 24 real seeds)

$$\text{period} = \frac{1}{48}\left(2^{24}\right)^{24} \approx 2^{570} \approx 10^{171}$$

Final check of a random number generator:

use several random number generators for the same physical problem.

Example: Analysis of random number generators using Monte Carlo P.D. Coddington simulation, Int. J. Mod. Phys C 5, 547 (1994) (2d Ising model)

L.N. Fenenberg, V.A. Landau, E.J. Janke, Phys. Rev. Lett 73, 2573 (1994): Physical Tests for Random Numbers in Simulations (3-tuple correlations in GFSR)

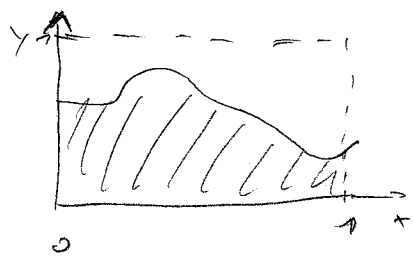
II. The Monte Carlo Method

i) Simple sampling

" Monte Carlo computations may be viewed as attempts to estimate the value of a (multiple) integral. "

Consider: $I = \int_0^1 f(x)\,dx \quad ; \; 0 \le f(x) \le 1 \;\; \forall x \in [0,1]$

$$=: \int_0^1 dy \int_0^1 dx \; g(x,y)$$



with: $g(x,y) = \begin{cases} 0 & y > f(x) \\ 1 & y \le f(x) \end{cases}$

stochastic approximation of this integral:

generate N points $z_n = (z_{nx}, z_{ny}) \; ; \; z_{nx}, z_{ny} \in [0,1[$

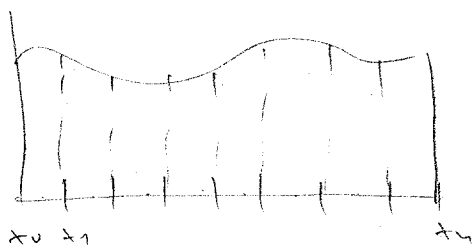
by a random number generator

$$\Rightarrow \quad I \approx \bar{g} = \frac{1}{N} \sum_{n=1}^{N} g(z_{nx}, z_{ny}) = \frac{N^*}{N}$$

with N^* the number of points with $f(z_{nx}) \le z_{ny}$.

In low dimensions: inefficient compared to numerical integration methods

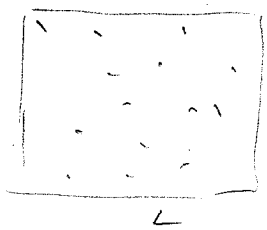
In high dimensions: One typically performs a limiting procedure in numerical integration



M: number of intervalls $\to \infty$

In d dimensions this means using M^d sample points of the function to be probed; for large d this number becomes untractable.

Consider 100 particles in 2 dim box.



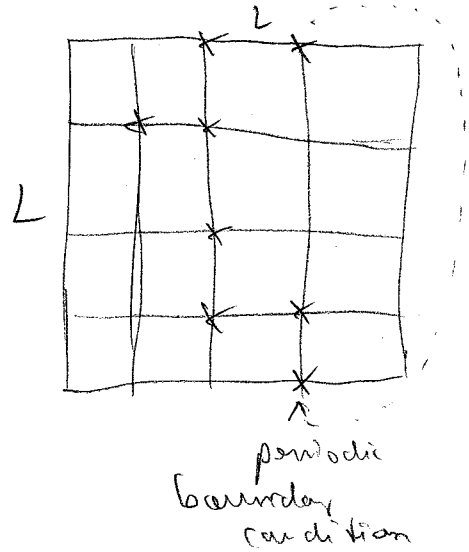
configuration space is $(L \times L)^{100} \Rightarrow d = 200$

only 10 intervals for each coordinate $\Rightarrow 10^{200}$ points

to generate a random „representative" sample of points a „Representative": works if each point has equal probability

Examples: (i) generate a starting configuration for a binary alloy of composition $A_x B_{1-x}$ on some regular lattice

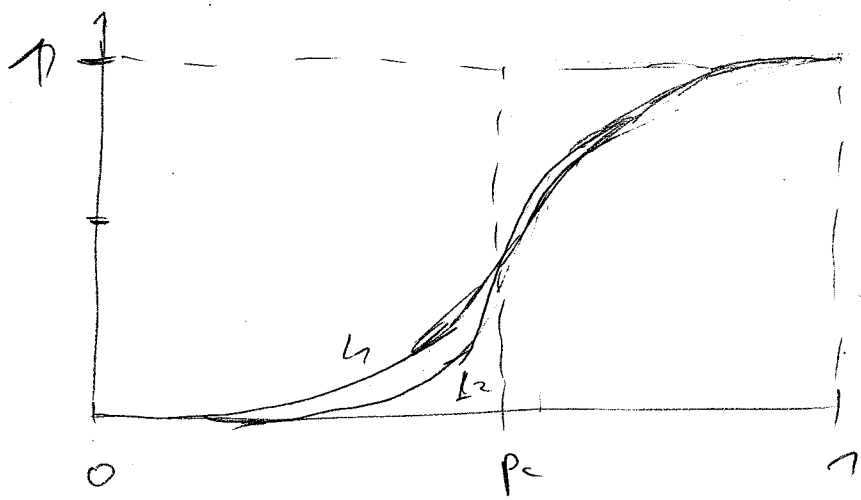
(ii) The site percolation problem.



with probability p select a lattice site to be occupied.

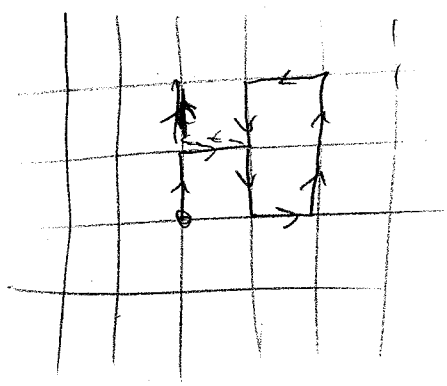
What is the probability for a spanning cluster to occur?

In the thermodynamic limit this probability jumps from 0 for $p \leq p_c$ to 1 at p_c

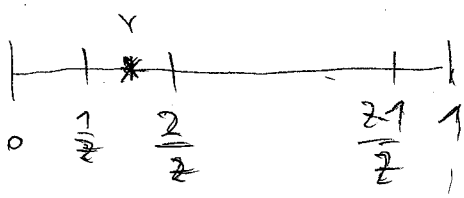


$L_2 > L_1$

(iii) The random walk on the lattice



A random walk of N steps. start at $(x,y) = (0,0)$ with probability $\frac{1}{z}$ [for 2d square lattice $z=4$] jump to one of the next neighbors



draw a random number "r" and determine in which interval it lies.

Number of configurations for a walk of N steps: $\#_{conf}^{RW} = z^N = e^{N \ln z}$

grows exponentially with the length of the walk

A typical quantity to measure for such a random walk would be

end-to-end-distance: $\langle R_e^2 \rangle = \langle (\vec{r}_N - \vec{r}_0)^2 \rangle \sim N$

$$\equiv \frac{1}{k} \sum_{k=1}^{k} (\vec{r}_N^{(k)} - \vec{r}_0^{(k)})^2$$

radius of gyration:

$$\langle R_G^2 \rangle = \left\langle \frac{1}{N} \sum_{n=0}^{N} (\vec{r}_n - \vec{r}_{cm})^2 \right\rangle \sim N$$

with $\vec{r}_{cm} = \frac{1}{N} \sum_{n=0}^{N} \vec{r}_n$

A variant of this model is the non-reversal random walk.

$$\#_{conf}^{NRRW} = z (z-1)^{N-1}$$

These lattice (or continuum) random walks are often used as model systems for the long-scale structure of polymers. Problem: these walks intersect.

Solution: Self-avoiding Random Walk: SAW

$$\#_{conf}^{SAW} \sim z_{eff}^{N} N^{\gamma - 1} \quad \text{for } N \to \infty$$

$$z_{eff} < z - 1$$

For the SAW one finds

$$\langle R_e^2 \rangle \sim N^{2\nu} \qquad \nu = \frac{3}{4} \text{ in } d = 2$$

$$\langle R_G^2 \rangle \sim N^{2\nu} \qquad \nu = 0.588 \text{ in } d = 3$$

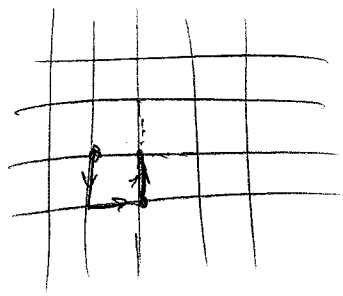
Problem with the simple sampling.

$$\frac{\#_{conf}^{SAW}}{\#_{conf}^{NRRW}} \sim \frac{1}{z}\left(\frac{z_{eff}}{z-1}\right)^N N^{\gamma-1} \xrightarrow[N\to\infty]{} 0 \qquad \text{since } z-1 > z_{eff}$$

This problem is called attrition ☹

Task: Generate NRRW of length $N=100$ on the square lattice and measure the fraction of the remaining SAW as a function of step number n. Total sample 10^6. Measure the end-to-end distance distribution

Biased Sampling: At each step select only one of the possible direction which do not lead to an overlap.



2 possibilities left.

→ but then the realizations are not equally probable.

weight of one realization:

$$\omega(N) = \prod_{n=1}^{N} \omega_n \quad ; \quad \omega_n = \frac{1}{\#_{\text{poss. dir. in step } n}}$$

Correct for this when

$$\langle R_e^2 \rangle = \left[\sum_{n=1}^{k} \omega(n)^{-1}\right]^{-1} \sum_{n=1}^{k} \omega(n)^{-1} R_e^{(n)\,2}$$

In statistical physics we typically are not faced with a uniform probability of all states.

Canonical ensemble and Gibbs measure

$$d\mu_\beta(\vec{x}) = \frac{1}{Z} e^{-\beta \mathcal{H}(\vec{x})} d\vec{x} \quad \text{(probability measure)}$$

$\vec{x}$: point in configuration space : dN coordinates (d: space dimension, N: number of particles)

$\beta = \frac{1}{k_B T}$

$d\vec{x}$: Lebesgue measure

Z : partition function $= \int_{\mathcal{C}} d\vec{x}\, e^{-\beta \mathcal{H}(\vec{x})}$

$\mathcal{C}$ = Configuration Space.

→ regions with low energy are more probable than regions with high energy.

If you want to calculate an average of an observable

$$\langle A \rangle_T = \int_{\mathcal{C}} A(\vec{x})\, d\mu_\beta(\vec{x})$$

$$= \frac{1}{Z} \int_{\mathcal{C}} A(\vec{x})\, e^{-\beta \mathcal{H}(\vec{x})} d\vec{x}$$

it would be good if one could generate the "integration grid" points in such a way that they are distributed according to $d\mu_\beta(\vec{x})$.

This is the idea of importance sampling. Suppose we can generate the sample points according to some probability density $p(\vec{x})$.

$$\Rightarrow \quad \langle A \rangle_T = \frac{\int_{\mathcal{L}} A(\vec{x})\, e^{-\beta \mathcal{H}(\vec{x})}\, d\vec{x}}{\int_{\mathcal{L}} e^{-\beta \mathcal{H}(\vec{x})}\, d\vec{x}}$$

$$\approx \frac{\sum_{i=1}^{M} A(\vec{x}_i)\, e^{-\beta \mathcal{H}(\vec{x}_i)} / p(\vec{x}_i)}{\sum_{i=1}^{M} e^{-\beta \mathcal{H}(\vec{x}_i)} / p(\vec{x}_i)}$$

Optimal choice: $p(\vec{x}_i) \sim e^{-\beta \mathcal{H}(\vec{x}_i)}$

$$\Rightarrow \quad \langle A \rangle_T \approx \frac{1}{M} \sum_{i=1}^{M} A(\vec{x}_i)$$

$\Rightarrow$ use a stochastic process to generate these points

(ii) Stochastic Processes.

<u>Def</u>: A stochastic process is a family of random variables $(X_t)_{t \in I}$ indexed by an ordered set I. ($t_0 < t_1 < \dots$)

discrete process: $I = \mathbb{N}$

continuous process $I = \mathbb{R}_0^+$

For each fixed $t_n \in I$ the random variable obeys some probability distribution

$$p_{t_n}(x_n) \equiv p(x_n, t_n)$$

To completely specify the stochastic process one needs to know all n-point distributions

$$p_n(x_n, t_n; x_{n-1}, t_{n-1}; \ldots; x_1, t_1)$$

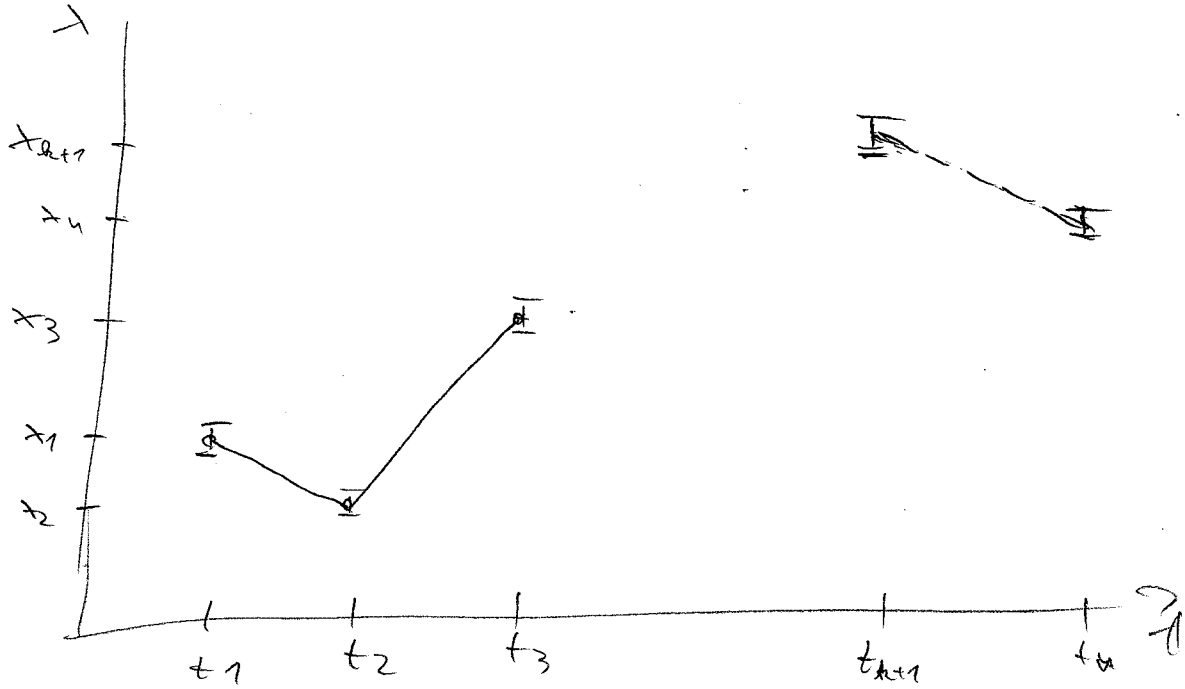
$$t_1 < t_2 \ldots < t_n$$

⇒ This is pretty hopeless!

Introduce conditional probabilities.

$$p_n(x_n, t_n; x_{n-1}, t_{n-1}; \ldots; x_1, t_1) = p_{n-k|k}(x_n, t_n; x_{n-1}, t_{n-1}; \ldots; x_{k+1}, t_{k+1} | x_k, t_k; \ldots; x_1, t_1)\, p_k(x_k, t_k; \ldots; x_1, t_1)$$

Example



Special case:

$$p_n(x_n,t_n;\dots;x_1,t_1) = p_{1|n-1}(x_n,t_n|x_{n-1},t_{n-1};\dots;x_1,t_1)\,p_{n-1}(x_{n-1},t_{n-1};\dots;x_1,t_1)$$

<u>Def.</u>: For a Markov process one has

$$p_{1|n-1}(x_n,t_n|x_{n-1},t_{n-1};\dots;x_1,t_1) = p_{1|1}(x_n,t_n|x_{n-1},t_{n-1})$$

"The process has no memory". The current state determines the probability for the next one.

For a Markov process one therefore has

$$p_n(x_n,t_n;\dots;x_1,t_1) = \prod_{\ell=2}^{n} p_{1|1}(x_\ell,t_\ell|x_{\ell-1},t_{\ell-1})\,p_1(x_1,t_1)$$

i.e. one needs to know only two functions

$p_1(x,t)$: one point probability density

$p_{1|1}(x',t'|x,t)$: conditional (transition) probability, propagator

Let us now look at a 3-point function

$$p_3(x_3,t_3;x_2,t_2;x_1,t_1) = p_{1|1}(x_3,t_3|x_2,t_2)\,p_{1|1}(x_2,t_2|x_1,t_1)\,p_1(x_1,t_1)$$

Integration over x_2:

$$p_2(x_3,t_3;x_1,t_1) = \int dx_2\, p_{1|1}(x_3,t_3|x_2,t_2)\,p_{1|1}(x_2,t_2|x_1,t_1)\,p_1(x_1,t_1)$$

$$= p_{1|1}(x_3,t_3|x_1,t_1)\,p_1(x_1,t_1)$$

$\Rightarrow$

$$p_{1|1}(x_3,t_3|x_1,t_1) = \int dx_2\, p_{1|1}(x_3,t_3|x_2,t_2)\, p_{1|1}(x_2,t_2|x_1,t_1)$$

$t_3 > t_2 > t_1$

Chapman - Kolmogorov equation ; consistency equation for Markov processes.

Let us now consider a Markov process with stationary (time translation invariant) transition probabilities

$\Rightarrow$ $p_{1|1}(x_2,t_2|x_1,t_1) = p_{1|1}(x_2,t_2+\tau|x_1,t_1+\tau)$

$$p_{1|1}(x_2,t_2|x_1,t_1) = p_t(x_2|x_1) \qquad t \equiv t_2 - t_1$$

p_t : transition probability within a time interval t

$\hookrightarrow$ $$p_{t+t'}(x_3|x_1) = \int dx_2\, p_{t'}(x_3|x_2)\, p_t(x_2|x_1)$$

Derivation of the differential form of the Chapman-Kolmogorov equation : t' small

$$p_{t'}(x_3|x_2) = \left[1 - w_{tot}(x_2)\,t'\right)\delta(x_3 - x_2) + t'\,w(x_3|x_2) + o(t')$$

$\left[\int dx_3\, p_{t'}(x_3|x_2) = 1\right]$

$w(x_3|x_2)$: transition rate from x_2 to x_3

$$w_{tot}(x_2) = \int dx_3\, w(x_3|x_2)$$

CKE :

$$p_{t+t'}(x_3|x_1) = (1 - w_{tot}(x_3)\,t')\,p_t(x_3|x_1) + t' \int dx_2\, w(x_3|x_2)\, p_t(x_2|x_1)$$

$\Rightarrow$

$$\frac{p_{t+t'}(t_3|t_1) - p_t(t_3|t_1)}{t'} = \int dt_2\, w(t_3|t_2)\, p_t(t_2,t_1) - \int dt_2\, w(t_2|t_3)\, p_t(t_3|t_1)$$

limit $t' \to 0$ $\Rightarrow$ Master equation

$$\frac{\partial}{\partial t} p_t(t_3|t_1) = \int dt_2\, w(t_3|t_2)\, p_t(t_2|t_1) - \int dt_2\, w(t_2|t_3)\, p_t(t_3|t_1)$$

now multiply by $p_1(t_1|t)$ and integrate over t_1:

$$\underbrace{\frac{\partial}{\partial t} p_1(t_3,t)}_{\text{rate of change}} = \underbrace{\int dt_2\, w(t_3|t_2)\, p_1(t_2,t)}_{\text{influx}} - \underbrace{\int dt_2\, w(t_2|t_3)\, p_1(t_3,t)}_{\text{outflux}}$$

Master equation for probability density is a continuity equation

Discrete (time continuous) version:

$$\frac{\partial}{\partial t} p_1(t_3,t) = \sum_{t_2} w(t_3|t_2)\, p_1(t_2,t) - \sum_{t_2} w(t_2|t_3)\, p_1(t_3,t)$$

time discrete version ($t_3 \to t$; $t_2 \to t'$; $p_1 \to p$)

$t_n = n\,\Delta t$

$$p(t, n+1) = p(t,n) + \sum_{t'} \Delta t\, w(t|t')\, p(t',n) - \sum_{t'} \Delta t\, w(t'|t)\, p(t,n)$$

At $\omega = W$ is a transition probability per step.

$$\Rightarrow \quad p(x, n+1) = p(x, n) + \sum_{x'} W(x|x')\, p(x', n) - \sum_{x'} W(x'|x)\, p(x, n)$$

This is the time evolution equation that is "solved" (iterated) in a Markov-chain Monte Carlo method.

How does this help us?

Define $p_{eq}(x)$ to be the stationary solution of the master equation.

$$\rightarrow \quad \sum_{x'} W(x|x')\, p_{eq}(x') = \sum_{x'} W(x'|x)\, p_{eq}(x)$$

We request $p_{eq}(x) = \frac{1}{Z} e^{-\beta \mathcal{H}(x)}$ &

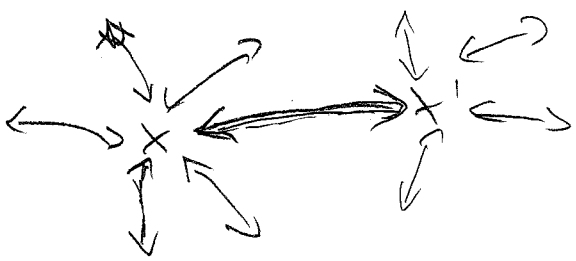
$$\Rightarrow \quad \sum_{x'} W(x|x')\, e^{-\beta \mathcal{H}(x')} = \sum_{x'} W(x'|x)\, e^{-\beta \mathcal{H}(x)}$$

$\Rightarrow$ consistency equation for the transition probabilities, but impractical

sufficient (but not necessary)

Detailed balance:

$$W(x|x')\, e^{-\beta \mathcal{H}(x')} = W(x'|x)\, e^{-\beta \mathcal{H}(x)}$$



Note: Except for the detailed balance condition, the transition probabilities are completely unspecified.

$$\frac{W(x/x')}{W(x'/x)} = e^{-\beta[\mathcal{H}(x)-\mathcal{H}(x')]}$$

Now define:

$$W(x/x') = W_0(x|x')\, W_T(x/x')$$

$W_0(x|x')$: probability to suggest a new state x starting from x'

example: single-spin-flip Ising model

one considers N spins and randomly selects one spin which one tries to flip:

$$W_0(x|x') = \frac{1}{N}$$

microscopic reversibility:

require $W_0(x|x') = W_0(x'|x)$

makes life easier, often assumed, but not necessary [keep in mind !]

$W_T(x|x') \equiv$ thermal part $\equiv$ acceptance probability

Again one has infinitely many choices.

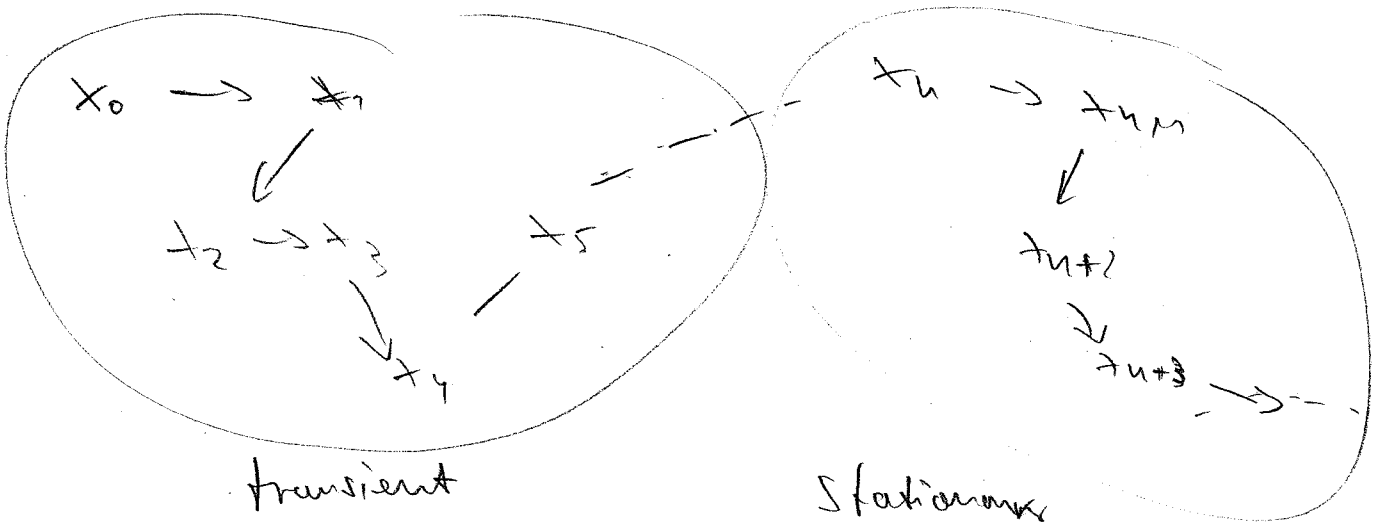
Most often used: Metropolis acceptance probability.

$$W_\mu(x|x') = \min(1, e^{-\beta \Delta x})$$
$$= \min(1, e^{-\beta [x(x') - x(x)]})$$

Great strength of the Monte Carlo method:
freedom to design efficient moves: W_0

<u>efficient</u>: i) one needs to start from some arbitrary configuration and it may take some time to „reach equilibrium", i.e., to reach the stationary part of the Markov chain

ii) Even in equilibrium measurements of observables (functions in configuration space) have a finite autocorrelation time which has to be made as small as possible.



Question: Is it guaranteed that a unique stationary state exists and is reached?

This is a question about the ergodic properties of the Markov chain.

i) the Markov chain has to be irreducible, i.e., every state can be reached in a finite number of steps from every other one.

ii) the chain is aperiodic: no integer $k > 1$ exists such that:

$$p_{jj}^{n} = 0 \quad \text{for} \quad n \neq \nu k$$

↑ return probability in n steps

$\Rightarrow$ all states are ergodic and the mean recurrence time to a state is given by

$$\tau_{rec}(x_i) = \frac{1}{p_{eq}(x_i)} \qquad \text{[discrete state space]}$$

Literature: W. Feller: An Introduction to Probability Theory and Its Applications, Vol I + II Wiley ...

[check in library because it is expensive]