

# Monte Carlo methods

Dénes Sexty

University of Graz

2020-2021, winter term

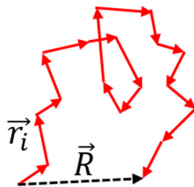


- 1 Examples: Ising model etc.
- 2 Integration vs Sampling
- 3 Pseudo random number generators
- 4 Generation of random numbers with a given given distribution
- 5 Percolation, **Random Walks**
- 6 **Importance Sampling, Markov chains, Metropolis Alg.**
- 7 **Statistics, error estimates: Jackknife, bootstrap**

Conformation of a polymer chain in a solution

How far are the ends?

$N$  monomers  $\rightarrow R \sim N^\nu$



Simplest model: step in a random direction for each  $i$   
Also models a drunkard's movement  $\rightarrow$  "Random walk"

$R \sim t^{1/2}$  disagrees with experiment (of polymer chains)



We walk on a line, every step  $p = 0.5$  for left,  $p = 0.5$  for right.

Step at time  $i$ :  $Z_i$ , i.e.  $Z_i \in \{-1, 1\}$ , after  $i$  steps:  $R_i = \sum_{j=1}^i Z_j$   
 each  $Z_i$  is independent  $\implies \langle R_i \rangle = \sum \langle Z_i \rangle = 0$

What's the typical distance?

$$\langle R_n^2 \rangle = \sum_{i=1}^n \langle Z_i^2 \rangle + 2 \sum_{i < j} \langle Z_i Z_j \rangle = n \quad (1)$$

If  $n$  is large  $R_n$  is the sum of many independent variables  $\rightarrow$  Gaussian distribution.

### Generalizations, relations:

- random walk in many dimensions: each direction has  $1/(2d)$  chance
- Gaussian random walk: stepsize is a Gaussian random variable. Used in finance
- Random walk without lattice: step in a random direction (sometimes also at random times)
- Wiener Process: limit of random walk with very small stepsizes (both space and time)

monomers repulse each other  $\rightarrow$  Polymer chain cannot cross itself  $\rightarrow R \sim N^{1/2}$   
is wrong

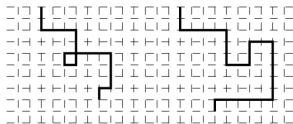
**No backtrack random walk** The next step cannot immediately turn back.  
In the first step there are  $2d$  possibilities, later only  $2d - 1$   
Crossings are still allowed.

**Self avoiding random walk (SAW)** Visiting the same site twice is not allowed  
(monomers repulse)

Defining the ensemble: What should be the weight of a random walk?

**RW:** all walks of length  $N$  are equally probable.

**SAW:** all walks of length  $N$  are equally probable, as long as they are SAWs.



Not a SAW

SAW

⇒ **naive algorithm**: Generate random walks. Throw result away if it intersects.

Works if  $N$  is small, otherwise most walks get thrown away: “attrition problem”

## Biased sampling

Check at each step how many steps are available:  $l_i$ , choose one among them with equal probabilities. If  $l_i = 0$ : throw away and start from the beginning.

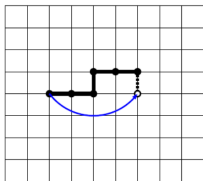
Now each walk has a weight:

$$w = \prod_{i=1}^N \frac{l_i}{2d-1} \quad (2)$$

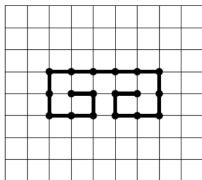
use these weights when calculating averages.

Generating a “chain” of walks:  $\dots \rightarrow W_i \rightarrow W_{i+1} \rightarrow \dots$

- 1 Assume we have a walk of length  $N$ :  $W_i$
- 2 delete a point from a random end
- 3 add one point to a random end with random direction.
  - if it is intersecting:  $W_{i+1} = W_i$  (change rejected)
  - else  $W_{i+1} = \text{new walk}$  (change accepted)



Can we get to every walk? No, there are trapped configurations:



Reptation can not change this or arrive here  $\implies$  process is non ergodic, sampling is biased (i.e. not uniform).

Such configurations are so rare that their absence in the sampling is not noticeable on the results.

### Pivot algorithm

- 1 Pick a random point on the random walk, this divides walk to two sides
- 2 choose a random side
- 3 transform the chosen side using lattice symmetries (rotations, mirroring)
- 4 accept the new walk if it is SAW, otherwise keep the old walk



Results from B. Li et. al. J. Stat. Phys. **80**, 661 (1995)

$$\begin{aligned} \nu &= 3/4 && \text{in } d = 2, \\ \nu &= 0.59\dots && \text{in } d = 3, \\ \nu &= 1/2 && \text{in } d = 4 \end{aligned} \tag{3}$$

Further generalizations of the polymer model:  
monomer interactions (introducing a potential)  
interaction with solvent  
Thermal effects

Usual question (thermal average):

$$\langle F \rangle = \frac{1}{Z} \int F(p, q) e^{-\beta H(p, q)} d^N p d^N q =? \quad (4)$$

Can be also written as (assume  $F$  depends on energy  $E$  or do an average at constant energy otherwise)

$$\langle F \rangle = \int F(E) \rho(E) e^{-\beta E} dE \quad (5)$$

For  $N \gg 1$  the energy is sharply peaked. A typical uniform random configuration will miss the peak  $\implies$  random sampling very unefficient

If we could sample configurations with probability  $p(C_i) \sim e^{-\beta H(p_i, q_i)}$   
Averages calculated simply as

$$\langle F \rangle = \frac{1}{N} \sum_{i=1}^N F(p_i, q_i) \quad (6)$$

This is called **Importance sampling**

Consider a sequence of “configurations” (e.g. phase space points of some system) governed by some random process

$$\dots \rightarrow \Phi_i \rightarrow \Phi_{i+1} \rightarrow \dots$$

It's called a **Markov chain** if  $T(\Phi'_{i+1}|\Phi_i)$ , the probability to go from  $\Phi$  does not depend on  $i$  and only depends on  $\Phi_i$ , not on older  $\Phi_j$  with  $j < i$  (no memory)

Notation: transition probabilities:  $T(\Phi_k|\Phi_j) = T_{kj}$

At step  $i$  the system is in state  $k$  with probability  $P_k^{(i)}$

conservation of probability  $\implies$  :

$$\sum_j T_{jk} = 1 \tag{7}$$

at step  $i + 1$  we have

$$P_k^{(i+1)} = T_{kj} P_j^{(i)} \tag{8}$$

How does  $P^{(i)}$  change? We had  $P_k^{(i+1)} = T_{kj}P_j^{(i)}$

$$P_j^{(i+1)} - P_j^{(i)} = \sum_k T_{jk}P_k^{(n)} - P_j^{(n)} = \underbrace{\sum_k T_{jk}P_k^{(n)}}_{\text{inflow}} - \underbrace{\sum_k T_{kj}P_j^{(n)}}_{\text{outflow}} \quad (9)$$

Similar to the master equation

$$\frac{\partial P(\Phi, t)}{\partial t} = \sum_{\Phi'} W(\Phi \leftarrow \Phi')P(\Phi') - \sum_{\Phi'} W(\Phi' \leftarrow \Phi)P(\Phi)$$

We are looking for a stationary solution  $\Pi_k$

$\implies T_{ik}\Pi_k = \Pi_i$  eigenvector with eigenvalue  $\lambda = 1$

**Balance equation:**

$$\sum_k T_{jk}\Pi_k^{(n)} = \sum_k T_{kj}\Pi_j^{(n)} \quad (10)$$

**Detailed balance:**  $T_{jk}\Pi_k^{(n)} = T_{kj}\Pi_j^{(n)}$

Suppose we choose  $T_{jk}$  such that the detailed balance is satisfied for some  $\Pi_k$

Consider the evolution of  $P$ :  $P^{(1)} \rightarrow P^{(2)} \rightarrow \dots$

Does it converge to  $\Pi_k$ ? Let's define  $\sigma_n = \sum_i |P_i^{(n)} - \Pi_i|$

$$\sigma_{n+1} = \sum_i \left| \sum_j T_{ij} P_j^{(n)} - \Pi_i \right| = \sum_i \left| \sum_j (T_{ij} P_j^{(n)} - T_{ij} \Pi_j) \right| = \quad (11)$$

using detailed balance

$$= \sum_i \left| \sum_j (T_{ij} P_j^{(n)} - T_{ij} \Pi_j) \right| = \sum_i \left| \sum_j T_{ij} (P_j^{(n)} - \Pi_j) \right| \leq \sum_{i,j} T_{ij} |P_j^{(n)} - \Pi_j| = \sigma_n$$

Since  $\sum_j (P_j^{(n)} - \Pi_j) = 0$ , there is equality only if  $\sigma_n = 0$ , if we have  $T_{ij} > 0$   
(strong ergodicity)

$\implies$  convergence to unique equilibrium  $\Pi_k$

Convergence rate is given by second largest eigenvalue of  $T_{jk}$

If we can set up a process that satisfies detailed balance for some  $\Pi_k$ , the process will converge to  $\Pi_k$ .

**Metropolis algorithm** achieves this.

Starting from configuration  $\Phi_n$

- 1 proposal with probability  $T_0(\Phi'|\Phi_{n-1})$
- 2 Accepted with probability  $T_A(\Phi'|\Phi_{n-1})$   
if accepted:  $\Phi_{n+1} = \Phi'$   
else:  $\Phi_{n+1} = \Phi_n$

To satisfy detailed balance, we must have

$$T_0(\Phi'|\Phi)T_A(\Phi|\Phi')\Pi(\Phi) = T_0(\Phi|\Phi')T_A(\Phi|\Phi')\Pi(\Phi') \quad (12)$$

detailed balance:

$$\frac{T_A(\Phi|\Phi')}{T_A(\Phi'|\Phi)} = \frac{\Pi(\Phi)}{\Pi(\Phi')} \frac{T_0(\Phi'|\Phi)}{T_0(\Phi|\Phi')} = F(\Phi|\Phi') \quad (13)$$

Note that  $F(\Phi|\Phi') = 1/F(\Phi'|\Phi)$

We choose e.g.

$$T_A(\Phi'|\Phi) = \begin{cases} F(\Phi'|\Phi) & \text{if } F(\Phi'|\Phi) < 1, \\ 1 & \text{otherwise} \end{cases} \quad (14)$$

or

$$T_A(\Phi'|\Phi) = \frac{F(\Phi'|\Phi)}{1 + F(\Phi'|\Phi)} \quad (15)$$

Usually we choose the symmetric (aka. reversible) proposal:

$$T_0(\Phi'|\Phi) = T_0(\Phi|\Phi')$$

Equilibrium is given by some action:  $\Pi(\Phi) = e^{-S[\Phi]}$  In this case:

$$F(\Phi'|\Phi) = \frac{\Pi(\Phi')}{\Pi(\Phi)} = e^{-S(\Phi') + S(\Phi)} = e^{-\Delta S} \quad (16)$$

In this case the acceptance probability is:

$$T_A(\Phi'|\Phi) = \begin{cases} e^{-\Delta S} & \text{if } \Delta S > 0, \\ 1 & \text{if } \Delta S < 0 \end{cases} \quad (17)$$

How to choose proposal probability?

change too big  $\implies$  acceptance rate will be small

change too small  $\implies$  exploration of configuration space too slow

Should be optimized



- 1 Initialize configuration
- 2 Choose parameters for proposal step (if applicable)
- 3 Generate  $\Phi'$  for  $\Phi_n$
- 4 calculate  $T_A(\Phi'|\Phi_n)$
- 5 generate random uniform  $r \in [0, 1]$
- 6 accept if  $r < T_A \rightarrow \Phi_{n+1} = \Phi'$   
else  $\Phi_{n+1} = \Phi_n$
- 7 goto 3 if  $n_{max}$  not reached

The sequence of configurations can be seen as dependent on “MC time”. This is not the same as the real dynamical evolution of the system. One measures time in MC hits/lattice site to make it volume independent

$$H = -J \sum_{\text{neighbors}} s_i s_j - h \sum_i s_i \quad (18)$$

Update step: take a spin, and flip it.

Either go through spins over the lattice sequentially, or always choose a random spin. Ergodic (any configuration can be reached) and reversible.

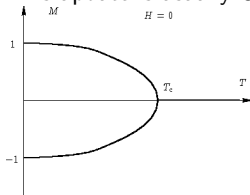
Since energy is local, only the neighboring spins are needed to calculate energy change  $\Delta E$ , which depends on the number of antiparallel spins.

Can calculate probabilities beforehand.

e.g. in  $d = 2$  and  $h = 0$  (there are more possibilities for  $h \neq 0$ )

# of antipar. spins	0	1	2	3	4
$\Delta E/J$	8	4	0	-4	-8
$\ln(T_A)$	$-8\beta J$	$-4\beta J$	1	1	1

This update is usually OK, near critical points it is too slow



Correlation length  $\xi$  diverges

Time to flip a correlated region  $\tau \sim \xi^z$

$z$  is usually close to 2

Near criticality, time to thermalization  $t_{CPU} \sim L^{d+z}$

### Efficient update strategies

$z$  is dependent on the update algorithm. If we flip a correlated cluster of spins at once, we eliminate  $z$  **Cluster algorithm**

Another fast algorithm: **Worm algorithm**

Suppose we have a MC simulation, results:  $x_i$ ,  $i = 1 \dots N$   
 In equilibrium, they all have the same average and variance:

$$\langle x_i \rangle = \langle x \rangle, \quad \sigma_{x_i}^2 = \langle (x_i - \langle x_i \rangle)^2 \rangle = \sigma_x^2 \quad (19)$$

Unbiased estimators:

$$\hat{x} = \frac{1}{N} \sum x_i, \quad \hat{\sigma}_x^2 = \frac{1}{N-1} \sum (x_i - \hat{x})^2 \quad (20)$$

$\langle \hat{x} \rangle = \langle x \rangle$ , but how close is it? Look at its variance:

$$\begin{aligned} \sigma_{\hat{x}}^2 &= \langle (\hat{x} - \langle x \rangle)^2 \rangle = \left\langle \left( \frac{1}{N} \sum (x_i - \langle x \rangle) \right)^2 \right\rangle = \\ &= \frac{1}{N^2} \left\langle \sum_{i,j} (x_i - \langle x \rangle)(x_j - \langle x \rangle) \right\rangle = \frac{1}{N} \langle x^2 \rangle + \frac{1}{N^2} \sum_{i \neq j} \langle x_i x_j \rangle + \langle x \rangle^2 \end{aligned} \quad (21)$$

For uncorrelated configurations, we have  $\langle x_i x_j \rangle = \langle x_i \rangle \langle x_j \rangle$

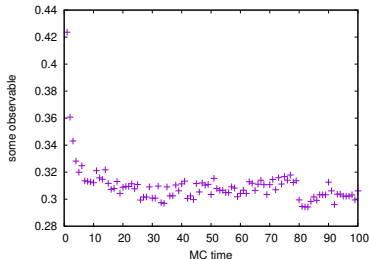
$$\sigma_{\hat{x}}^2 = \frac{1}{N} \hat{\sigma}_x^2 \quad (22)$$

In this case we quote the results of the simulation as

$$\hat{x} \pm \sigma \quad (23)$$

with  $\sigma = \sigma_{\hat{x}} = \hat{\sigma}_x / \sqrt{N}$

In a Monte Carlo simulation, measuring observables as a function of  $n$  one gets  
e.g.:



exponential thermalization to thermal averages

Initial thermalization should be discarded from averages

Thermalization rate can be observable dependent!

Averages are easy:

$$\langle O \rangle = \frac{1}{n_{max} - n_{therm}} \sum_{i > n_{therm}} O_i \quad (24)$$

How do we calculate errors?

Since we generate  $\Phi_{n+1}$  from  $\Phi_n$ , they are correlated ( $\langle A \rangle = \mu_A$ ).

$$C(A, B) = \frac{\langle (A - \mu_A)(B - \mu_B) \rangle}{\sigma_A \sigma_B} = \frac{\langle AB \rangle - \langle A \rangle \langle B \rangle}{\sigma_A \sigma_B} \quad (25)$$

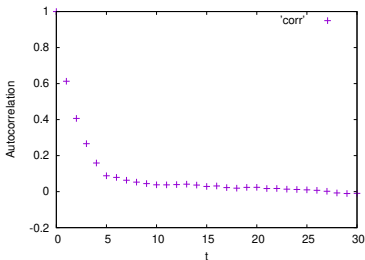
Suppose we have some observable  $A_n$  at MC time  $n$ . We define **Autocorrelation**:

$$C_A(t) = C(A_i, A_{i+t}) = \frac{\langle A_i A_{i+t} \rangle - \langle A \rangle^2}{\langle A^2 \rangle - \langle A \rangle^2} \quad (26)$$

(In equilibrium correlation depends only on  $t$ )

$C_A(t) \sim \exp(-t/\tau_A)$   
 $\tau_A$ : autorrelation time

We need  $\tau = \sup \tau_A$



In the correlated case, we have to be more careful estimating the variance.

Using  $D_A(t) = \langle A_i A_{i+t} \rangle - \langle A \rangle^2$

$$\begin{aligned}
 \sigma_{\bar{x}}^2 &= \frac{1}{N^2} \left\langle \sum_{i,j} (x_i - \langle x \rangle)(x_j - \langle x \rangle) \right\rangle = \frac{1}{N^2} \sum_{i,j} D_A(|i-j|) = \\
 &= \frac{1}{N^2} \sum_{t=-(N-1)}^{N-1} \sum_{k=1}^{N-|t|} D_A(|t|) = \sigma_A^2 \sum_{t=-(N-1)}^{N-1} \frac{N-|t|}{N^2} C_A(t) \\
 &= \frac{\sigma_A^2}{N} \sum_{t=-(N-1)}^{N-1} C_A(t) \left(1 - \frac{|t|}{N}\right) \approx \frac{\sigma_A^2}{N} 2 \left(\frac{1}{2} + \sum_{t=1}^N C_A(t)\right) = \frac{\sigma_A^2}{N} 2\tau_{A,int}
 \end{aligned}$$

we introduced **integrated autocorrelation time**:

$$\tau_{A,int} = \frac{1}{2} + \sum_{t=1}^N C_A(t) \approx \int_0^\infty e^{-t/\tau} dt = \tau \quad (27)$$



This can be understood as having

$$N_{indep} = \frac{N}{2\tau_{A,int}} \quad (28)$$

independent configurations.

Quote results as :

$$\hat{x} \pm \sqrt{\frac{1}{N} 2\tau_{A,int} \sigma_A^2} \quad (29)$$

Getting a good estimate for the autocorrelation requires a large dataset

Divide dataset into consequent dataset of size  $N_B$ .

If uncorrelated, their variance should decrease as  $\sim 1/N_B$ .

$\implies$  repeat blocking for different sizes, and consider the results statistically independent as  $\sim 1/N_B$  dependence sets in.

Problem: Some observables require "global quantities".

example 1: do a fit on the data as a function of some parameter. What is the statistical error of the fit parameters? one data point might give very noisy data such that no senseful fit can be performed

example 2: What is the error of the variance estimator?

How to calculate errors for such quantities? Using standard error propagation might be cumbersome or too difficult

(standard error propagation for 1 variable:  $\sigma_{f(A)} = \frac{df}{dA} \sigma_A$ )

Consider an initial sample set of  $N$  uncorrelated datapoints  $x_i$ .  
The original observable from the dataset:

$$\hat{\Theta} = f(\{x_i\}) \quad (30)$$

Now construct  $N$  smaller datasets by removing the  $n$ -th datapoint from the set.  
For each of this smaller datasets, we determine our observable  $\Theta_n$ .  
Estimator of the variance:

$$\sigma_{\hat{\Theta}}^2 = \frac{N-1}{N} \sum_{n=1}^N (\Theta_n - \hat{\Theta})^2 \quad (31)$$

We can also remove a bias from the estimator (or at least the  $1/N$  part), by calculating

$$\tilde{\Theta} = \frac{1}{N} \sum_{k=1}^N \Theta_k \quad (32)$$

and finally

$$\langle \Theta \rangle = \hat{\Theta} - (N-1)(\tilde{\Theta} - \hat{\Theta}) \pm \sigma_{\hat{\Theta}} \quad (33)$$

Usually one needs blocking to get rid of autocorrelation.  
divide the set  $x_i$  into  $N_{jk}$  separate “streams” of data  
Consider the blocks as the jackknife “points”  
e.g.  $x_i$ ,  $i = 1 \dots 1000$ , and take  $N_{jk} = 10$ , then we have

$$\begin{aligned}\hat{\Theta} &= f(\{x_{i=1\dots 1000}\}), \\ \Theta_1 &= f(\{x_{i=101\dots 1000}\}), \\ \Theta_2 &= f(\{x_{i=1\dots 100, 201\dots 1000}\}), \text{ and so on}\end{aligned}\tag{34}$$

and finally use the formulas on prev. slide for  $N = 10$

This is a resampling method. Consider an initial sample set of  $N$  uncorrelated datapoints  $x_i$

The original observable from the dataset:

$$\hat{\Theta} = f(\{x_i\}) \quad (35)$$

now create  $K$  new datasets  $S_k$  by drawing  $N$  random elements from the set  $\{x_i\}$ , with repetitions allowed.

for each set we calculate  $\Theta_k = f(S_k)$ .

We then calculate

$$\tilde{\Theta} = \frac{1}{K} \sum_{k=1}^K \Theta_k, \quad \sigma_{\tilde{\Theta}}^2 = \frac{1}{K} \sum_{k=1}^K (\Theta_k - \tilde{\Theta})^2 \quad (36)$$

We can use this as estimators of the average and the variance (these are biased, i.e. one needs a large  $K$  (check  $\hat{\Theta} - \tilde{\Theta}$ ):

$$\langle \Theta \rangle = \tilde{\Theta} \pm \sigma_{\tilde{\Theta}} \quad (37)$$