

Metropolis algorithm in the canonical ensemble

-

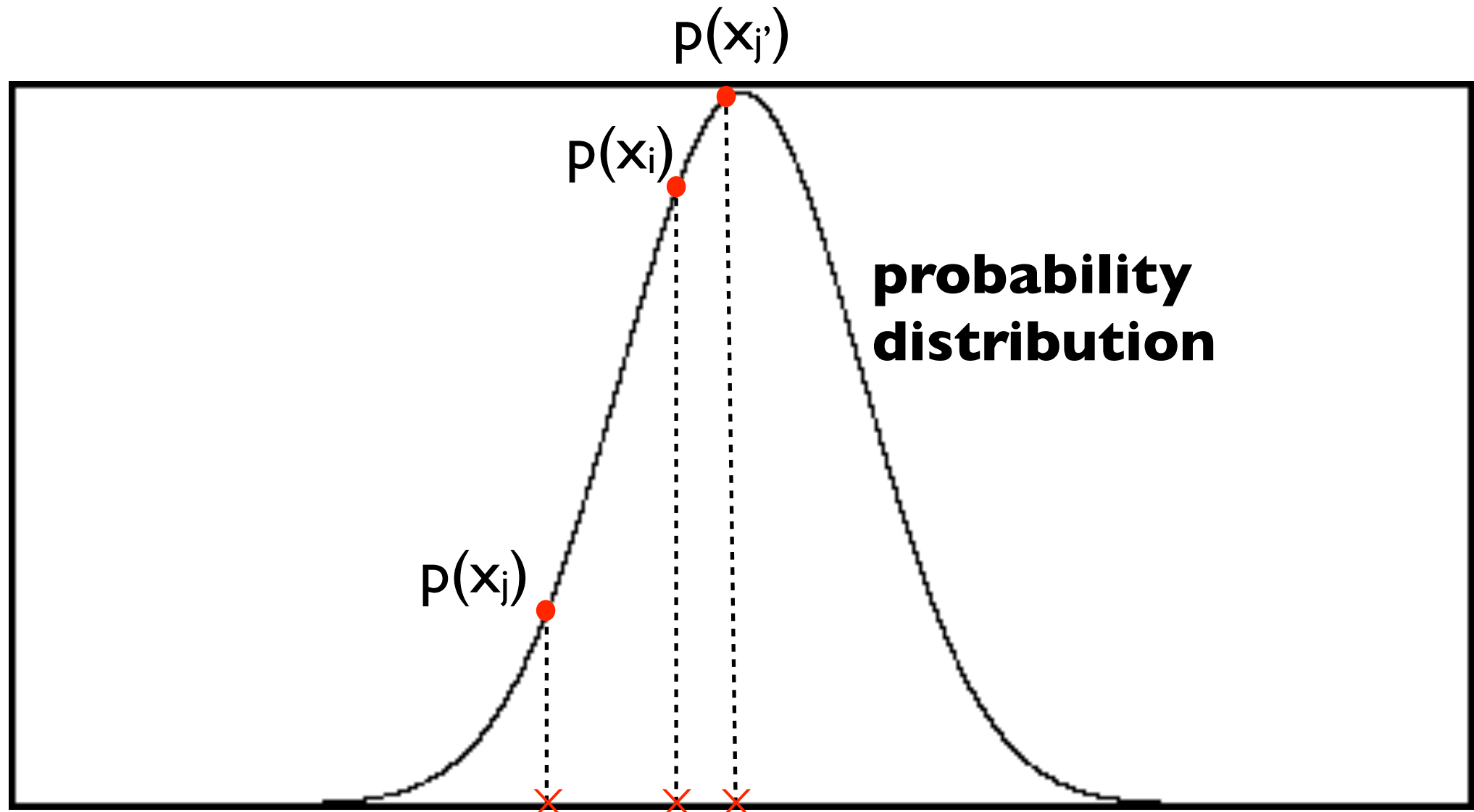
Microstates & macrostates

M. Peressi - UniTS - Laurea Magistrale in Physics
Laboratory of Computational Physics - Unit VIII

I part

from Lect.VII

- Metropolis Monte Carlo method (ex. 1,2)
- correlations (ex. 3)
- Metropolis and canonical ensemble (ex. 4)



move with probability
 $\frac{p(x_j)}{p(x_i)} < 1$,
 otherwise keep x_i

x_j x_i x_j'

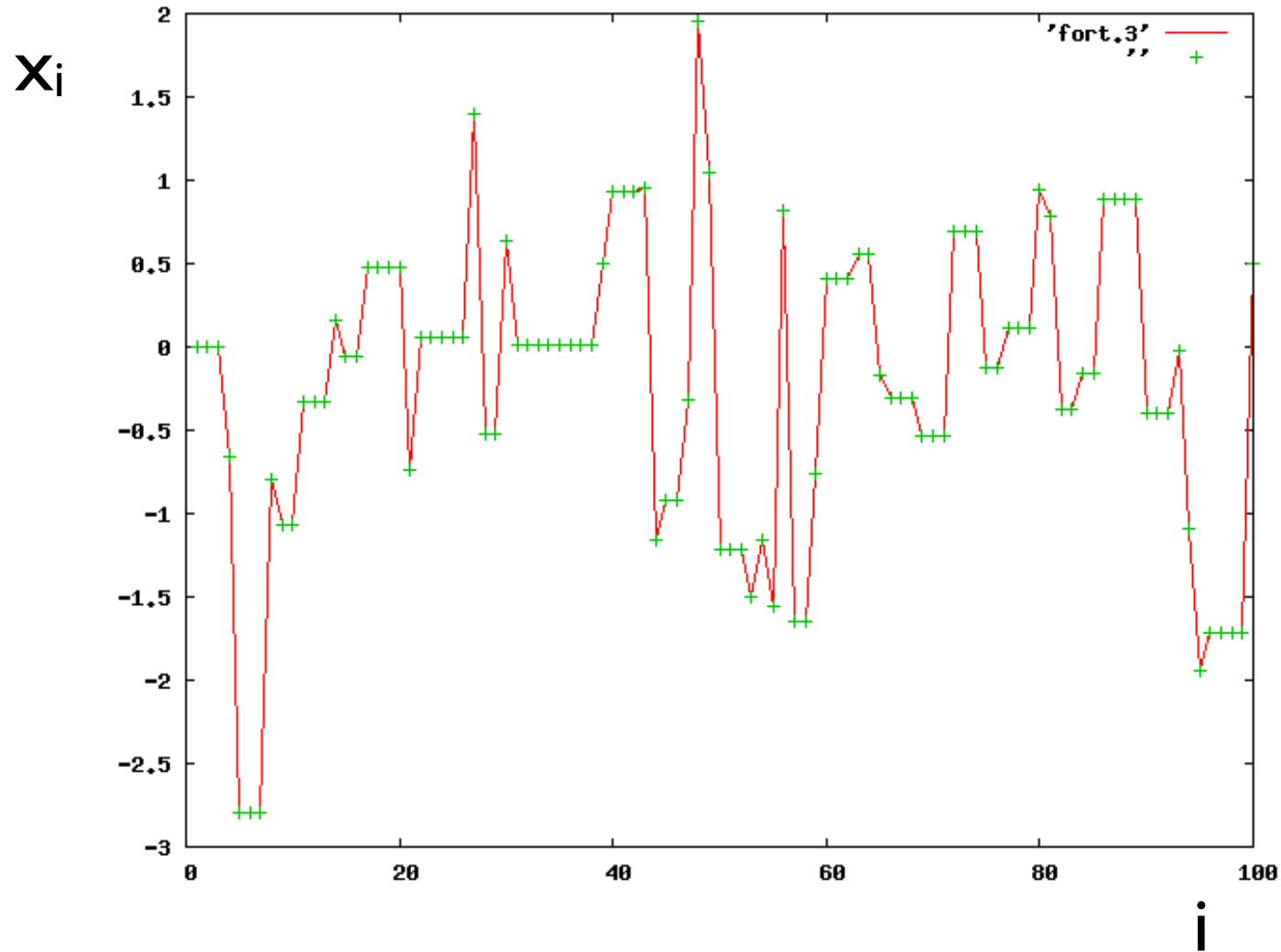


GO! since $\frac{p(x_j')}{p(x_i)} > 1$

initial position

$$T(x_i \rightarrow x_j) = \min \left[1, \frac{p(x_j)}{p(x_i)} \right]$$

example of a sequence of points randomly (Gaussian) distributed
generated according Metropolis



by its nature, Metropolis method introduces (at least short-range) correlations

ex.VII.3 - Correlations

[a very simple calculation if the array points(n) is stored]

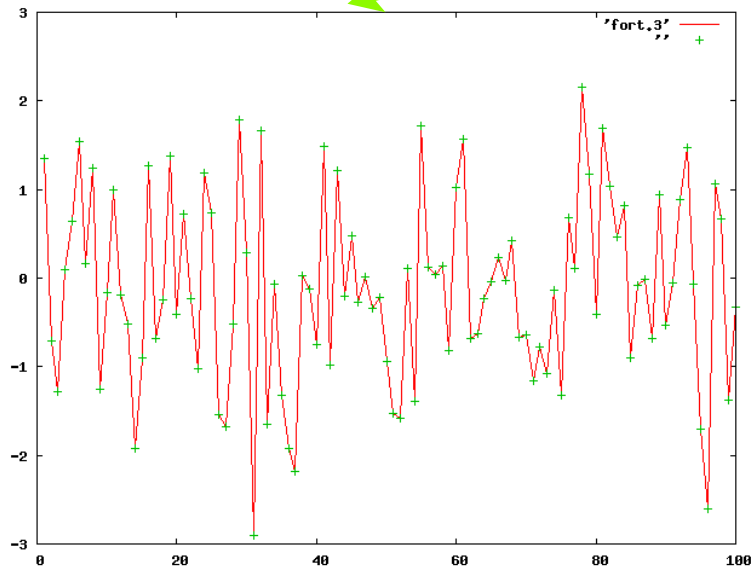
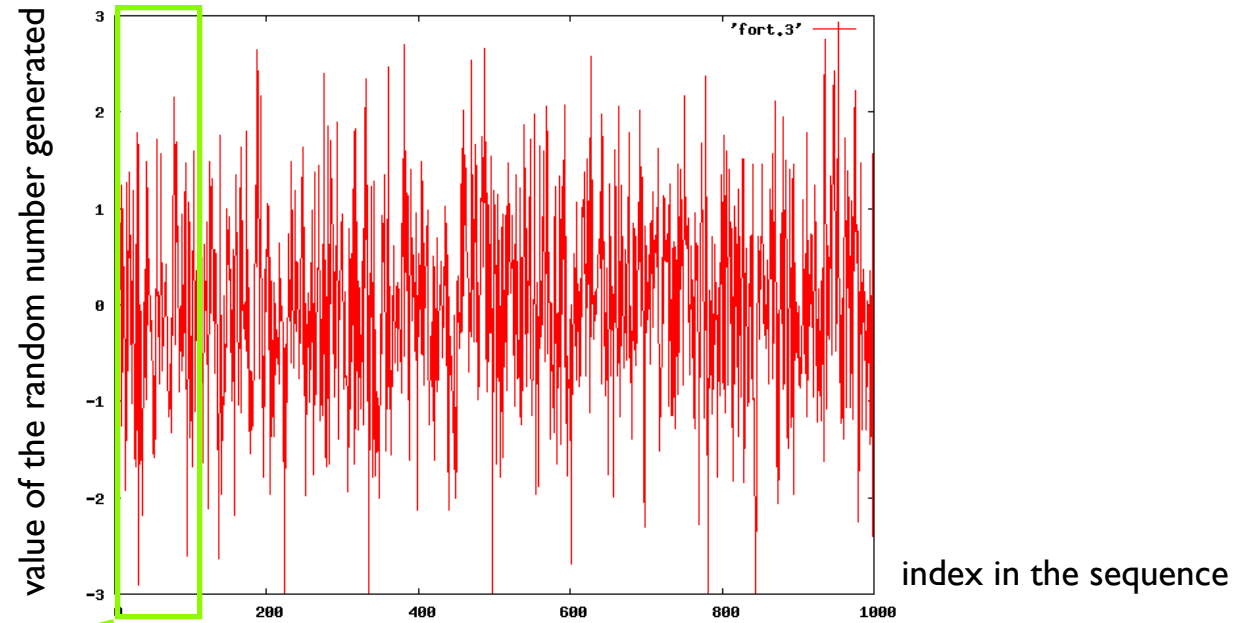
```
do j = 0, jmax
si=0._dp ; si2=0._dp ; sij=0._dp
  do i = 1,n-j
    si = si + points(i)
    si2 = si2 + points(i)**2
    sij = sij + points(i)*points(i+j)
  end do
  si = si/(n-j)
  si2 = si2/(n-j)
  sij = sij/(n-j)

write(2,*), j, (sij-si**2)/(si2-si**2)

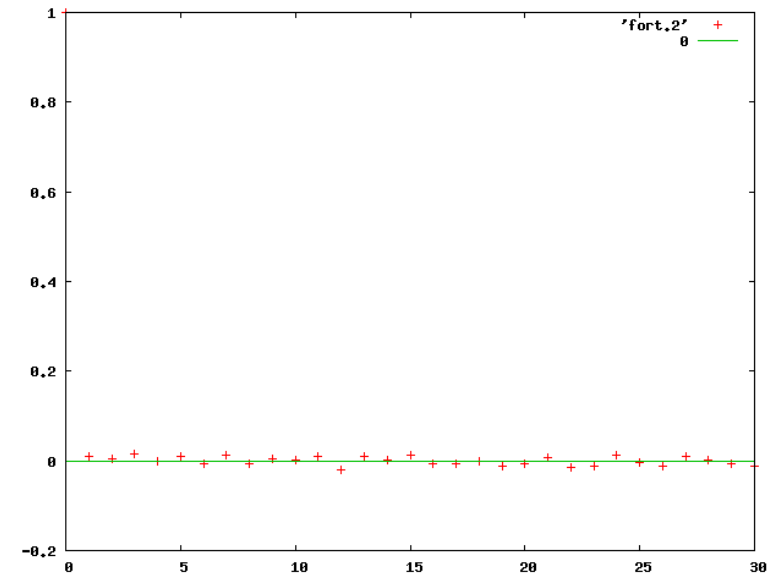
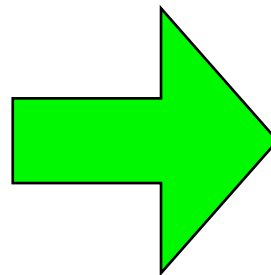
end do
```

$$C(j) = \frac{\langle x_{i+j}x_i \rangle - \langle x_i \rangle^2}{\langle x_i^2 \rangle - \langle x_i \rangle^2}$$

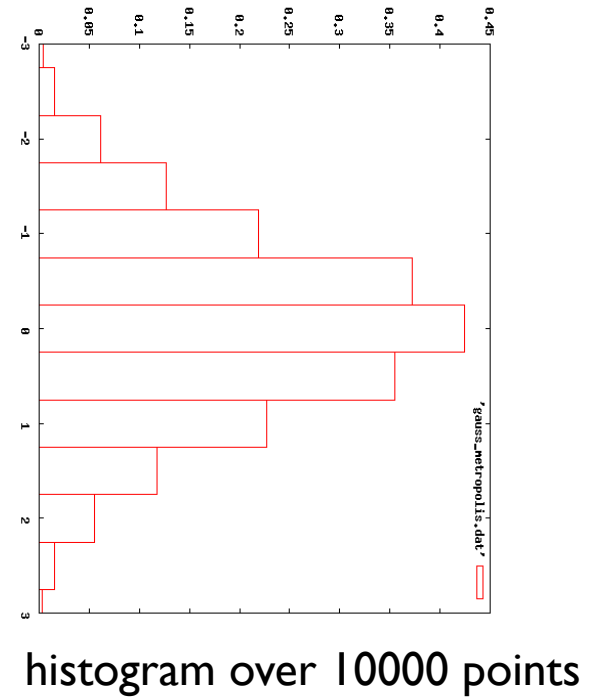
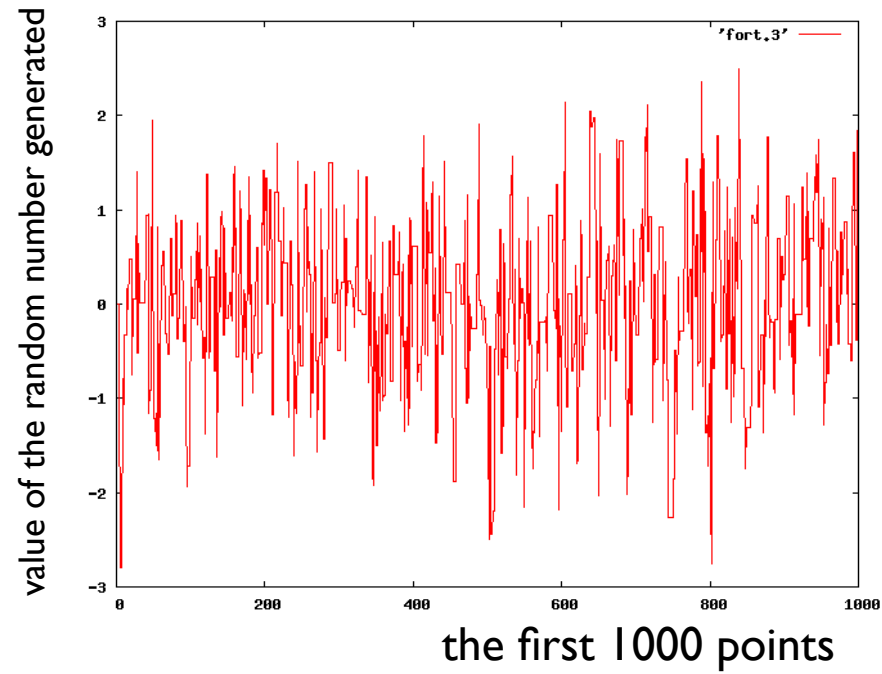
Correlations - Box-Muller algorithm



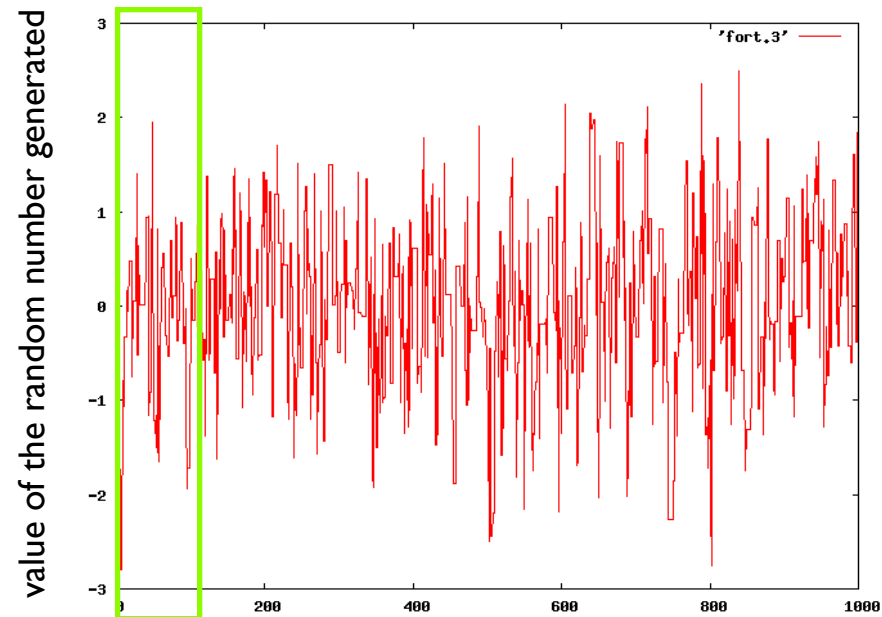
no correlations



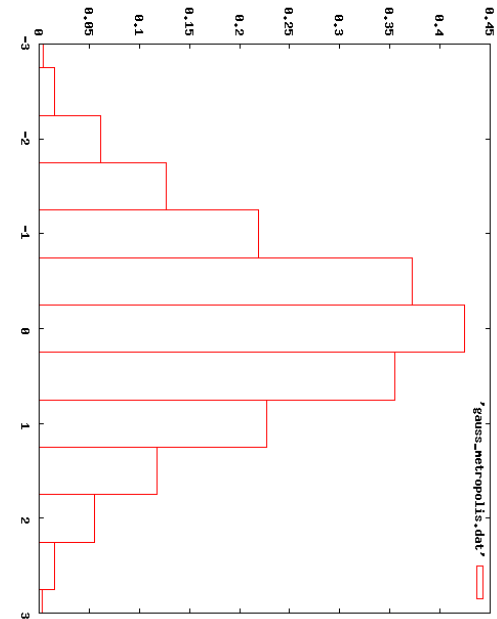
Correlations - Metropolis algorithm



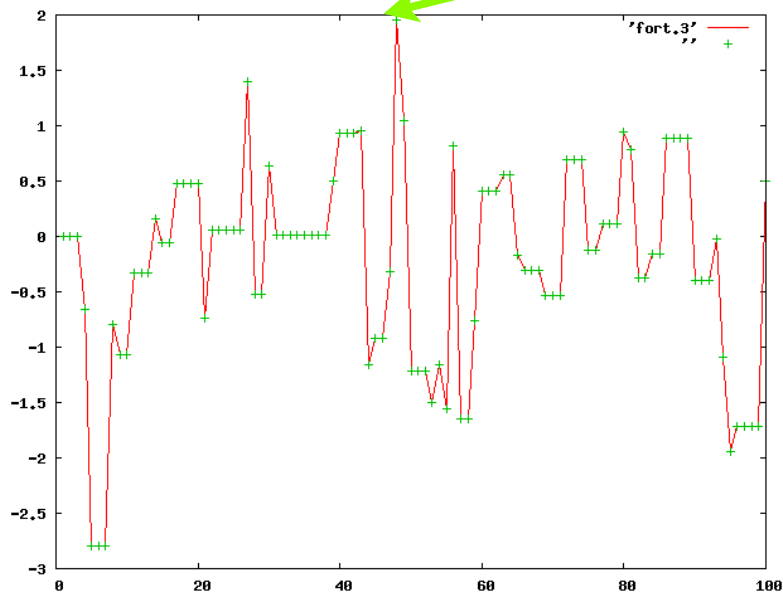
Correlations - Metropolis algorithm



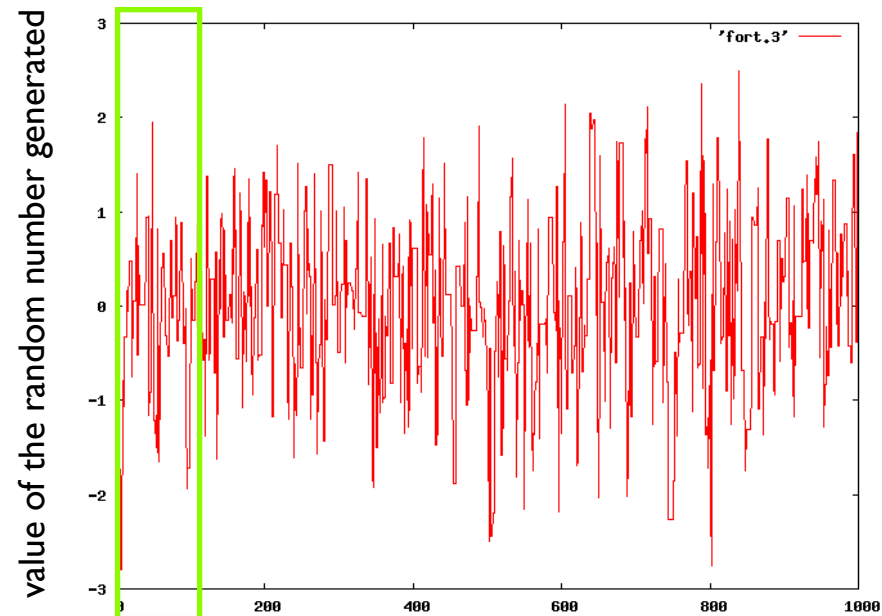
the first 1000 points



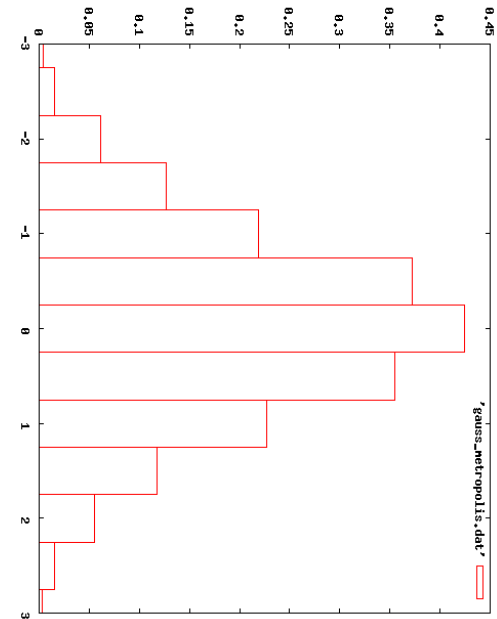
histogram over 10000 points



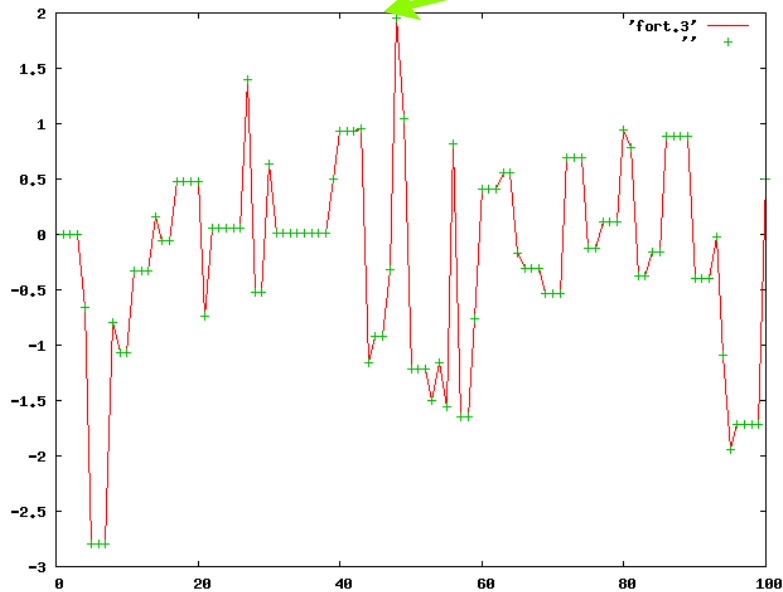
Correlations - Metropolis algorithm



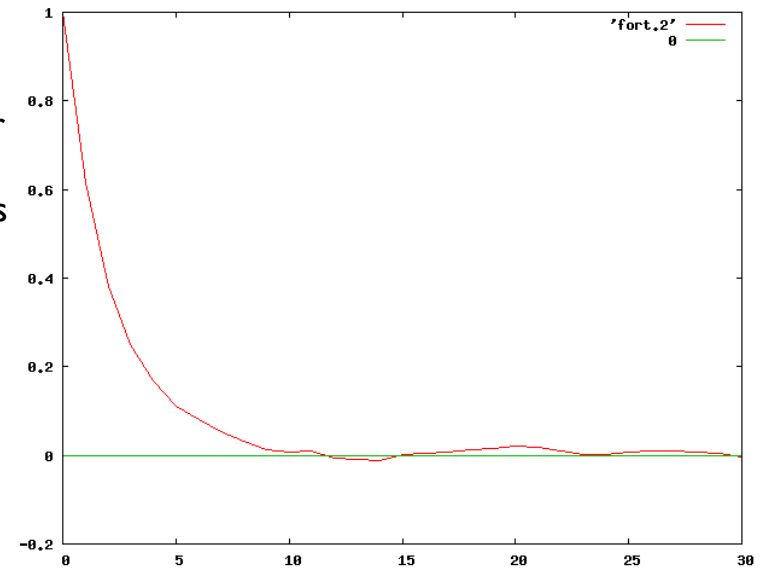
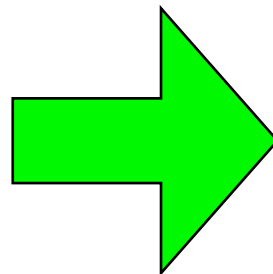
the first 1000 points



histogram over 10000 points



the short-term behavior
of the Markov chain
explains the correlations



ex VII. 4

application of the Metropolis algorithm
to the canonical ensemble

canonical ensemble :

fix N, V, T (*system in equilibrium with a thermal bath*)

The canonical ensemble

(N,V,T) fixed. The **probability** P_s for a system to be in the **microstate** s depends only on the **energy** E_s of the microstate and the **temperature** T (common to the system and the thermal bath):

$$P_s = \frac{1}{Z} e^{-\beta E_s}, \text{ (canonical distribution)}$$

where $\beta = 1/kT$, and Z is a normalization constant.

(in the canonical ensemble the energy is a characteristic of the **microstate**, **not** of the **macrostate**)
The higher is the energy of the state, the lower is the probability of finding the system in that state.

Because $\sum P_s = 1$,

$$Z = \sum_{s=1}^M e^{-E_s/kT}$$

(canonical partition function)

(M: all accessible microstates of the system, having the same N, V, T)

Averages in the canonical ensemble

The ensemble average of a physical quantity A can be calculated using the canonical distribution function:

$$\langle A \rangle = \sum_{s=1}^M A_s P_s = \frac{1}{Z} \sum_{s=1}^M A_s e^{-\beta E_s}$$

(M :
total number of
possible microstates)

In practice, approximating the sums over a finite number m of the total number M of accessible microstates:

$$\langle A \rangle \approx A_m = \frac{\sum_{s=1}^m A_s e^{-\beta E_s}}{\sum_{s=1}^m e^{-\beta E_s}}$$

(Note: m , not M !)

Averages in the canonical ensemble

Due to the functional form of $e^{-\beta E_s}$, this formulation is highly inefficient (many microstates have very small probability):

$$\langle A \rangle \approx A_m = \frac{\sum_{s=1}^m A_s e^{-\beta E_s}}{\sum_{s=1}^m e^{-\beta E_s}}$$

=> much better to use the **importance sampling**, i.e. simple averages of the desired physical quantity over microstates s generated according to the proper distribution:

$$A_m = \frac{1}{m} \sum_{s=1}^m A_s$$

Averages in the canonical ensemble

How to generate microstates properly distributed ?

=> generate a Markov chain with the transition matrix:

$$T_{s,trial} = \min \left[1, \frac{P_{trial}}{P_s} \right] = \min \left[1, \frac{e^{-\beta E_{trial}}}{e^{-\beta E_s}} \right] = \min [1, e^{-\beta \Delta E}]$$

where $\Delta E = E_{trial} - E_s$

Peculiarities w.r.t. the application of the Metropolis algorithm already discussed:

$e^{-\beta E_s}$ is a monotonic decreasing function of E_s ,
therefore we **accept all the moves decreasing the energy.**

Metropolis algorithm in the canonical ensemble

1. Establish an initial microstate.
2. Make a random trial change in the microstate. For example, choose a spin at random and flip it. Or choose a particle at random and displace it a random distance.
3. Compute $\Delta E \equiv E_{\text{trial}} - E_{\text{old}}$, the change in the energy of the system due to the trial change.
4. If ΔE is less than or equal to zero, accept the new microstate and go to step 8.
5. If ΔE is positive, compute the quantity $w = e^{-\beta\Delta E}$.
6. Generate a random number r in the unit interval.
7. If $r \leq w$, accept the new microstate; otherwise retain the previous microstate.
8. Determine the value of the desired physical quantities.
9. Repeat steps (2) through (8) to obtain a sufficient number of microstates.
10. Periodically compute averages over microstates.

Metropolis algorithm in the canonical ensemble

A few remarks:

- 1) Because it is necessary to evaluate only the ratio $P_j/P_i = e^{-\beta\Delta E}$, it is not necessary to normalize the probability.

$$(P_j/P_i = \pi_j/\pi_i)$$

2) ERGODICITY implicitly assumed!

3) TEMPERATURE:

If $E_B > E_A$, accept the new (higher energy) configuration with probability $p = e^{-(E_B - E_A)/T}$. This means that when the temperature is high, we don't mind taking steps in the "wrong" direction, but as the temperature is lowered, we are forced to settle into the lowest configuration we can find in our neighborhood.

Metropolis algorithm in the canonical ensemble

4)

Don't confuse P_s with $P(E_s)$, since $\Omega(E_s)$ microstates with the same energy E_s can exist, in general (*unless in particular cases where E_s uniquely identifies the microstate*). Therefore:

$$P(E_s) = \frac{\Omega(E_s)}{Z} e^{-\beta E_s}$$

$\Omega(E_s)$ is a density of states

where the **degeneracy of the energy level is accounted for**, whereas:

$$P_s = \frac{1}{Z} e^{-\beta E_s}$$

Boltzmann distribution in the canonical ensemble

The Metropolis algorithm really produces microstates with the Boltzmann distribution:

application to ideal classical 1D gas (ex. n. 4 Lect.VII)

1 free particle: Energy: $E = \frac{1}{2}mv^2$

in this case, velocity or energy labels a microstate

(the energy with a factor of 2, due to +/- sign of v);

we generate different microstates by random variations of the velocity and we accept/reject with Metropolis

Important quantities are the probabilities:

$P(v)dv$ that the system has a velocity between v and $v+dv$

or $P(E)dE$ that the system has an energy between E and $E+dE$

ideal classical 1D gas

A particle moving randomly has in each direction a distribution of the component of the velocity:

$$f(v_x) = \left(\frac{m}{2\pi k_B T} \right)^{1/2} e^{-mv_x^2/2k_B T} \quad (1)$$

$$\langle v_x^2 \rangle = \int_{-\infty}^{+\infty} v_x^2 f(v_x) dv_x = \frac{k_B T}{m} \quad (2)$$

In 1D:

$$f(v)2dv = P(E)dE$$

that gives:

$$P(E) = \frac{1}{(\pi k_B T)^{1/2}} \frac{1}{\sqrt{E}} e^{-E/k_B T}$$

In 3D:

$$P(E) = \frac{2}{\sqrt{\pi}} \frac{1}{(k_B T)^{3/2}} \sqrt{E} \exp\left(-\frac{E}{k_B T}\right) \quad (3D)$$

Note:

- recognize the density of states (analogy with the electronic DOS in the Sommerfeld model...)
- in the proposed 1D problem, $\Omega(E_s)$ is not just =2 because +/- v labels the 2 microstates corresponding to the same energy E .
The problem has a continuum energy spectrum.

Boltzmann distribution in the canonical ensemble

```
# T      : 1.00000
# <E0>   : .000000
# <v0>   : .000000
# dvmax  : 2.00000
# deltaE : 5.000000E-02
# nbin   : 79
```

$$T = 1 \rightarrow \langle E \rangle (\text{expected}) = 0.5 \quad (m = 1)$$

```
==> boltzmann.1K <==
# nMCsteps: 1000
# <E>      : .501263
# <v>      : 7.456664E-02
# accept.  : .692000
# sigma    : .713780
```

$$\sigma / \sqrt{n} = 0.022$$

(σ is the variance of the energy)

```
==> boltzmann.10K <==
# nMCsteps: 10000
# <E>      : .507580
# <v>      : 3.366172E-02
# accept.  : .707700
# sigma    : .726145
```

$$\sigma / \sqrt{n} = 0.007$$

```
==> boltzmann.1M <==
# nMCsteps: 1000000
# <E>      : .500138
# <v>      : 1.833840E-04
# accept.  : .693837
# sigma    : .707472
```

$$\sigma / \sqrt{n} = 0.0007$$

NOTE:

- Accuracy of $\sim 1\%$ on $\langle E \rangle$ and 10% on $\langle v \rangle$: NMCS=1000 is enough
- NOT ENOUGH to well reproduce the BOLTZMANN DISTRIBUTION! (1M needed!)
- ACCEPTANCE RATIO: constant, depends only on dvmax
- SIGMA also

Boltzmann distribution in the canonical ensemble

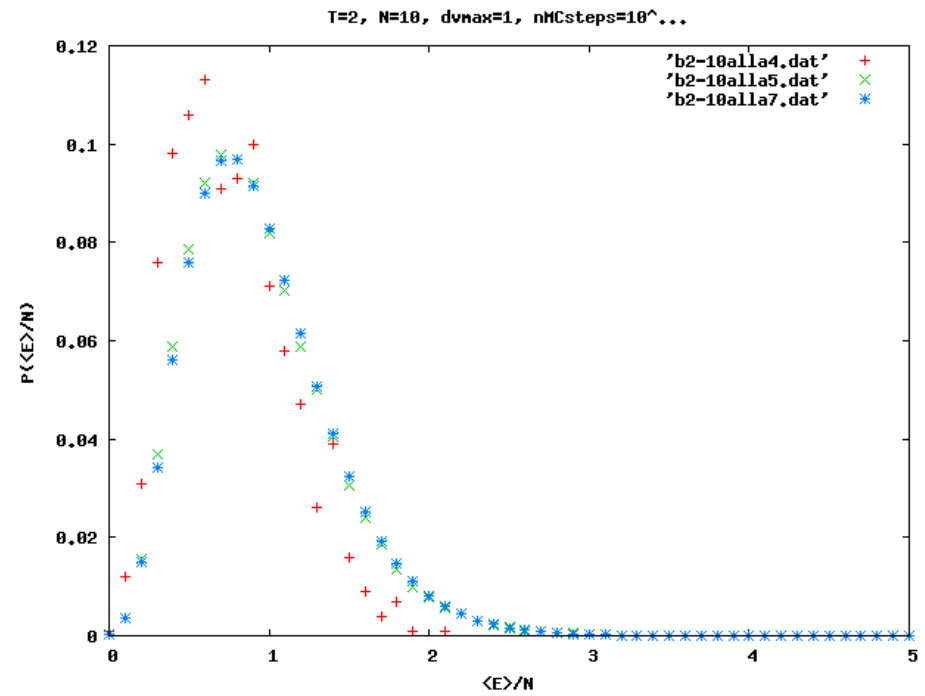
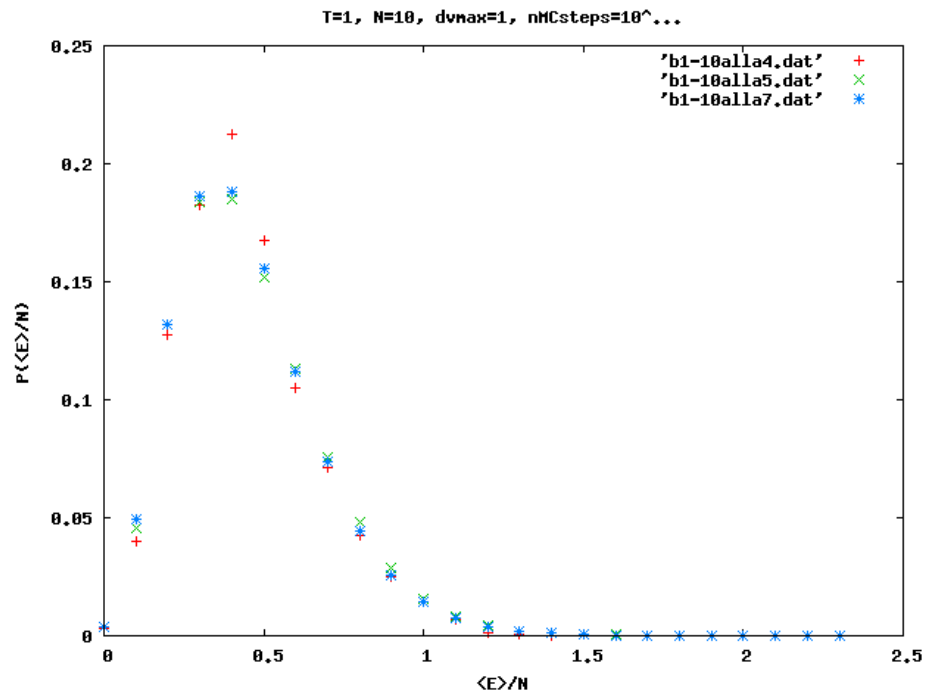
many particles: Energy: $E = \sum_{i=1}^N \frac{1}{2} m_i v_i^2$
in this case, the energy is **NOT** a label of a microstate
(there are several **microstates with the same total energy**)

Note: the energy histogram is NOT the distribution of microstates!

$$P(E) = \sum_{\substack{\text{states } s \\ \text{with } E_s = E}} P_s \quad \text{with } P_s = \frac{1}{Z} e^{-\beta E_s}$$

$$P(E) \propto e^{-\frac{(E - \langle E \rangle)^2}{2\sigma^2}} \quad \text{with } \langle E \rangle \text{ average over all the microstates}$$

What is P(E)? (exercise n. 4 Lec.VII - many particles)



ex.VII.4 introduces to another important issue
in stochastic simulations...

II part

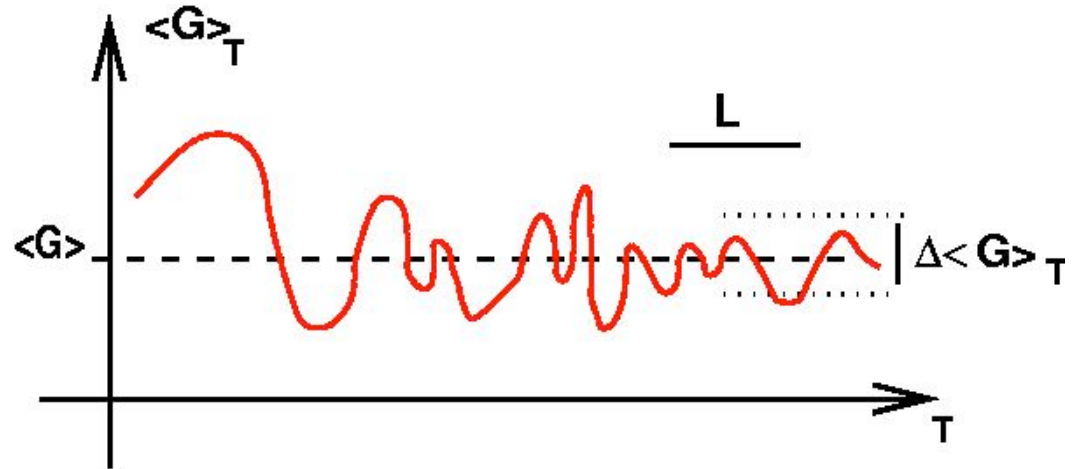
Statistical averages and
stochastic fluctuations

Averages

$$\langle G \rangle_T = \frac{1}{T} \sum_{t=1}^T G_t$$

Thermally equilibrated averages:

$$\langle G \rangle = \lim_{T \rightarrow \infty} \langle G \rangle_T$$



But in practice T is finite, and $\langle G \rangle_T$ oscillates (varying T):

divide T into intervals $A, B, C \dots$ of length L and sum **(block averages)**:

$$\langle G \rangle_T = \frac{L}{T} \sum_{I=A, B, C, \dots} \langle G \rangle^{(I)}$$

(σ_s / \sqrt{s})
 (σ_n / \sqrt{n})

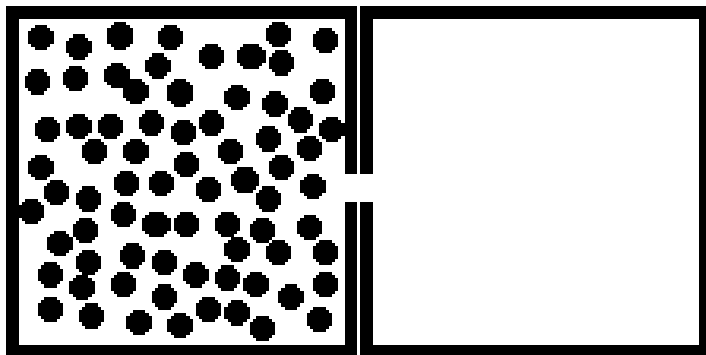
$$\Delta \langle G \rangle_T = \left[\frac{L}{T} \sum_I \left(\langle (G^{(I)})^2 \rangle - \langle G^{(I)} \rangle^2 \right) \right]^{1/2} \xrightarrow{T \rightarrow \infty} \sim \frac{1}{\sqrt{T}}$$

Note: not always $\Delta \langle G \rangle_T$ is a good indicator of the actual error!
(remind "ergodicity")

Stochastic fluctuations

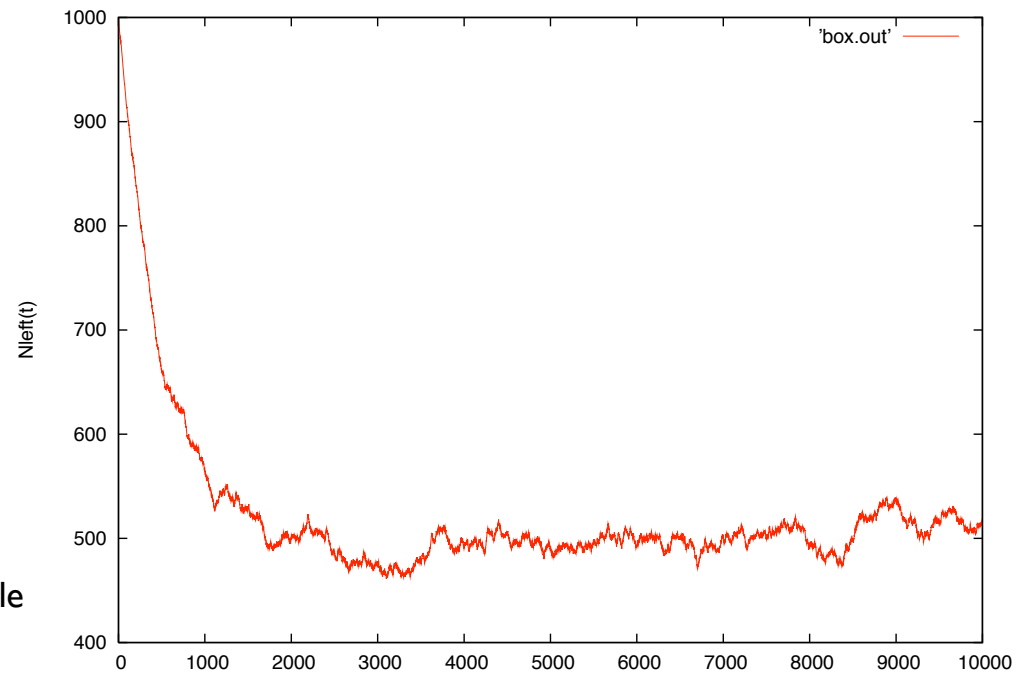
Fluctuations are always present, due to the nature of the system, also when evolving towards equilibrium.

A simple example: non-interacting classical particles in a box (gas diffusion)



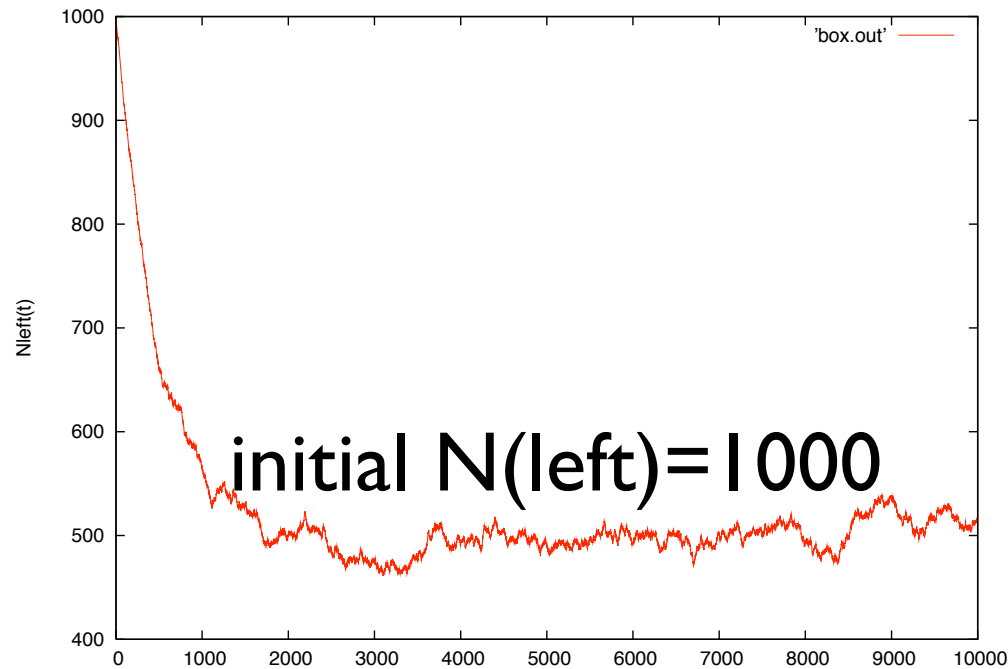
A box is divided into two parts communicating through a small hole. One particle randomly can pass through the hole per unit time, from the left to the right or viceversa.

$N_{\text{left}}(t)$: number of particles present at time t in the left side
Given $N_{\text{left}}(0)$, what is $N_{\text{left}}(t)$?



Approach to equilibrium
with fluctuations

Stochastic fluctuations



How to reduce **fluctuations**?

- more particles
- average over many simulation runs
- ...

What can we do with fluctuations?

Intrinsic energy fluctuations in the canonical ensemble - I

Remind: $\langle E \rangle = \frac{1}{Z} \sum_s E_s e^{-\beta E_s}$ and $Z = \sum_s e^{-\beta E_s}$, therefore: $\langle E \rangle = -\frac{\partial}{\partial \beta} \ln Z$

Consider the thermal capacity:

$$C_v = \frac{\partial \langle E \rangle}{\partial T} = \dots = -\frac{1}{kT^2} \frac{\partial \langle E \rangle}{\partial \beta}$$

we have

$$\frac{\partial \langle E \rangle}{\partial \beta} = -\frac{1}{Z^2} \frac{\partial Z}{\partial \beta} \sum_s E_s e^{-\beta E_s} - \frac{1}{Z} \sum_s E_s^2 e^{-\beta E_s} = \langle E \rangle^2 - \langle E^2 \rangle = -\langle (\delta E)^2 \rangle$$

where $\delta E \equiv E - \langle E \rangle$

Result:

$$C_v = \frac{\langle (\delta E)^2 \rangle}{k_B T^2}$$

The thermal capacity (or specific heat if considered for each particle) is related to the intrinsic stochastic energy fluctuations

Intrinsic energy fluctuations in the canonical ensemble - II

Since:

$$C_v = \frac{\langle (\delta E)^2 \rangle}{k_B T^2}$$

if N is the number of particles, we have:

$$\frac{\sqrt{\langle (\delta E)^2 \rangle}}{\langle E \rangle} = \frac{\sqrt{k_B T^2 C_v}}{\langle E \rangle} \propto \frac{\sqrt{N}}{N} \sim \frac{1}{\sqrt{N}}$$

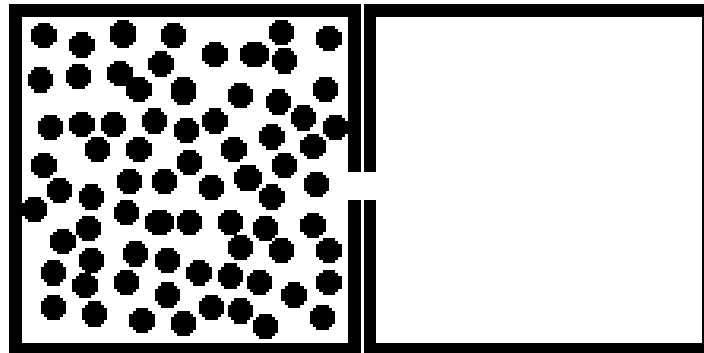
i.e., the relative energy fluctuations reduce when N is large

(correct; in the thermodynamic limit: $E \rightarrow const.$, macro \sim micro)

Macrostates, microstates and entropy at equilibrium

Macroscopic systems towards equilibrium

Again the simple example of non-interacting classical particles in a box (gas diffusion)



A box is divided into two parts communicating through a small hole. One particle randomly can pass through the hole per unit time, from the left to the right or viceversa.

$N_{\text{left}}(t)$: number of particles present at time t in the left side
Given $N_{\text{left}}(0)$, what is $N_{\text{left}}(t)$?

microcanonical ensemble :
fix E, N, V (*isolated closed system*)

Microcanonical ensemble

-a fundamental postulate-

Given an isolated system in equilibrium, it will assume with equal probability each of its accessible microstates

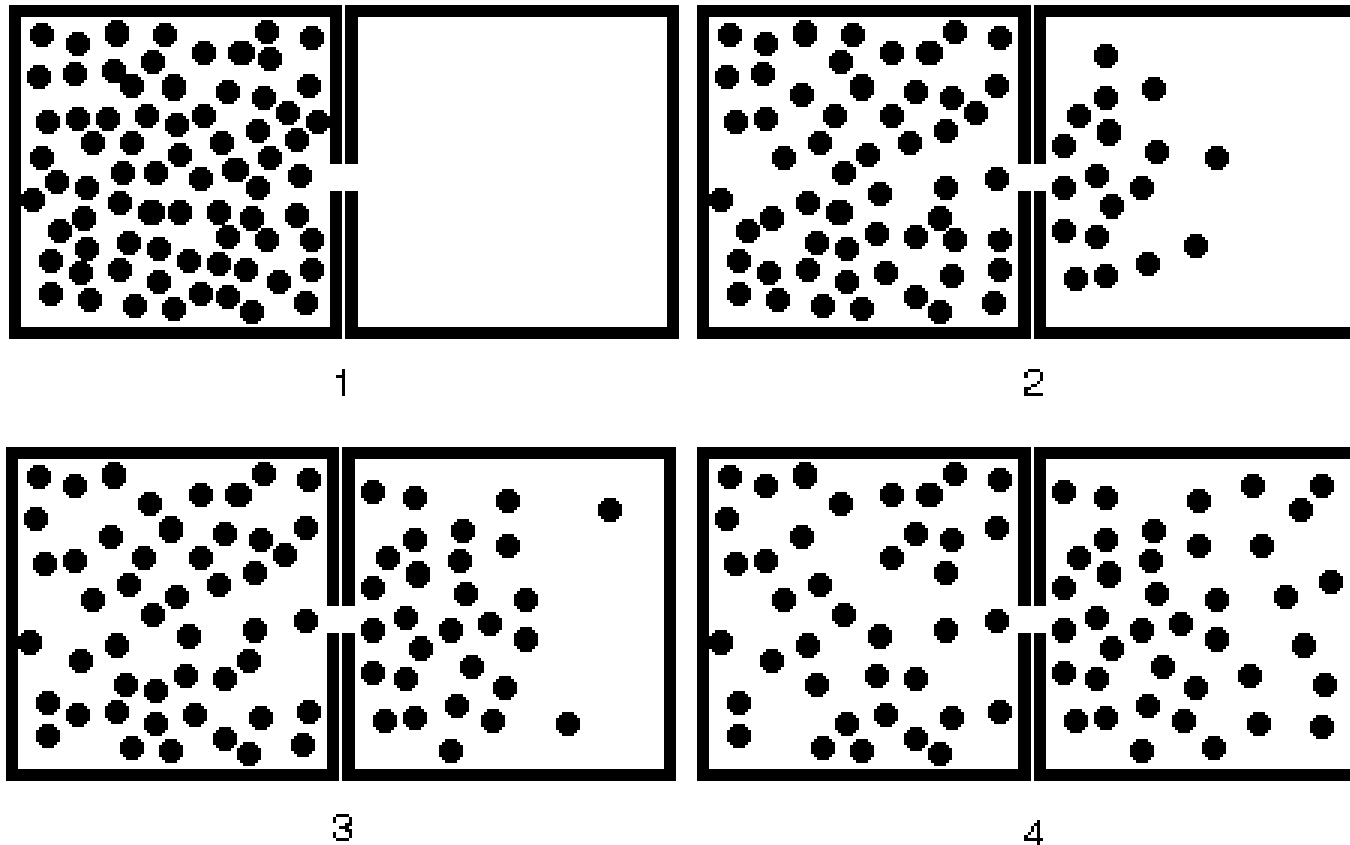
(i.e., a system in equilibrium does not have any preference for any of its available microstates)

Example: suppose a **macrostate** defined by (N, V, E) ; if $\Omega(N, V, E)$ is the # of microstates with energy $E \div dE$, for the **microstate** s the probability of occurrence is:

$$P_s = \begin{cases} 1/\Omega, & \text{if } s \text{ is accessible} \\ 0, & \text{otherwise.} \end{cases}$$

Consequence: for a system at equilibrium, the thermodynamic state (macrostate) which could result from the **largest number of microstates** is also the **most probable macrostate** of the system.

Approach to equilibrium



A box divided into left/right; total number N of non-interacting particles.

macrostate: specified by the **number** of particles n on the left side;

microstate: specified by the **specific list** of the n particles on the left side

Equilibrium and entropy

The macrostate is specified by the number of particles on the left side, n , and the corresponding microstates are:

$$\text{number of microstates} = \frac{N!}{n!(N-n)!} = \binom{N}{n} = \binom{N}{N-n}$$

The total number of microstates for $N=10$ is $2^{10}=1024$

n	# of microstates	$\log(\# \text{ of micr.})$
0	1	0.00
1	10	2.30
2	45	3.81
3	120	4.79
4	210	5.35
5	252	5.53
6	210	5.35
7	120	4.79
8	45	3.81
9	10	2.30
10	1	0.00

$n=5$ is the macrostate with the largest number of microstates!

($n=5$ is the most probable macrostate)



Equilibrium =
Maximum number of
possible microstates =
Maximum entropy

Equilibrium and entropy

The equilibrium corresponds to the most probable macrostate, which is that with the larger number of corresponding microstates; how to identify it?

We could simply generate microstates and count them...

but also use some smart algorithm (more suitable for computer simulations)...

Entropy: Coincidence method

(S.K. Ma, J. Stat. Phys. 26, 221 (1981))

Equilibrium = Maximum entropy = Maximum number of possible microstates

Too much effort to enumerate all of them!

Alternative procedure (good for computing):

A system evolving in time will duplicate a microstate, before or later..

The longer it takes for duplication, the fewer are the microstates in the corresponding macrostate. Hence, the lower is the entropy.

Idea: measure the ratio of the number of pairs of duplicated microstates to the total number of possible pairs; entropy is the log of the inverse ratio.

E.g.: suppose as in the previous slide $N=10$, and the **macrostate n=1**; consider 20 different **microstates** labelled with the “name” of the particle:

8 7 5 10 7 2 4 6 2 10 3 4 3 9 6 5 2 9 2 4

number of pairs of microstates: $20 \cdot (20-1)/2 = 190$. Here: 6 pairs for particle “2”; 1 pair with particle “10” etc etc... Sum all of them: get 15.

Ratio = $15/190$, Entropy: $S \propto \log(190/15) \sim 2.5$

Remind the definition of entropy:

$$S = -k_B \sum_s P_s \ln P_s \quad \text{in the canonical ensemble}$$

$$S = k_B \log \Omega \quad \text{in the microcanonical ensemble,}$$

where all the microstates
corresponding to a macrostate have
the same energy
(Ω is the number of microstates)

Metropolis method in the canonical ensemble and the simulated annealing

Metropolis and simulated annealing - I

- Stochastic search for **global minimum (a big problem!)**. Monte Carlo optimization.
- The concept is based on the manner in which liquids freeze or metals recrystallize. Sufficiently high starting temperature and slow cooling are important to avoid freezing out in metastable states.

Metropolis and simulated annealing - II

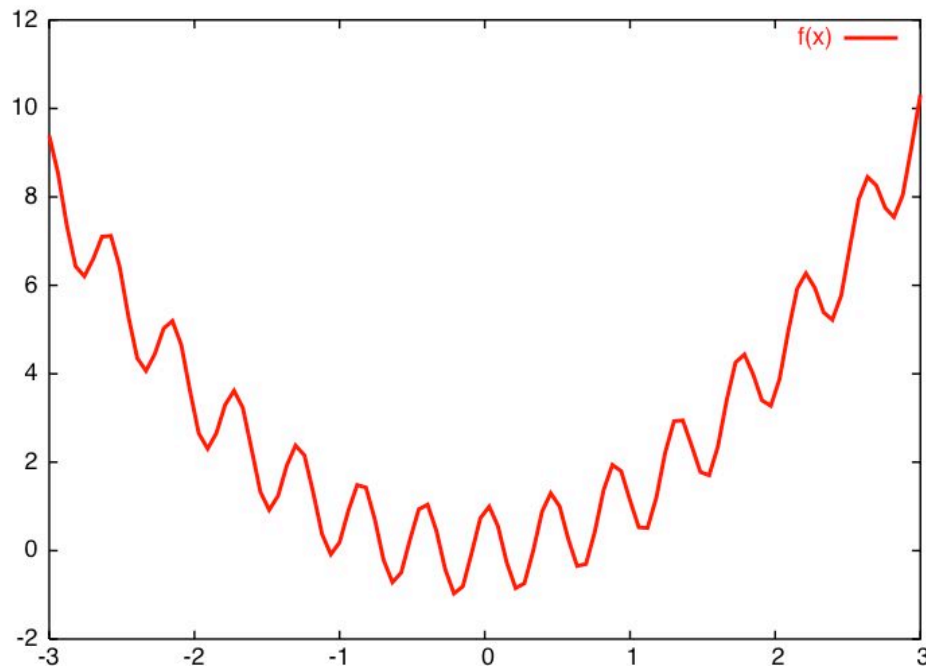
usual
Metropolis
procedure
in the
canonical
ensemble

- Thermodynamic system at temperature T , energy E .
- *Perturb configuration (generate a new one).*
- *Compute change in energy dE . If dE is negative the new configuration is accepted. If dE is positive it is accepted with a probability given by the Boltzmann factor : $\exp(-dE/kT)$.*
- *The process is repeated many times for good sampling of configuration space.*
- **then the temperature is slightly lowered and the entire procedure repeated, and so on, until a frozen state is achieved at $T = 0$.**

Example

in **simulated annealing.f90**:
minimization of

$f(x) = (x+0.2) * x + \cos(14.5 * x - 0.3)$
considered as an energy function and
using a fictitious temperature



Function to be minimized: $f(\mathbf{x})$; Starting point: \mathbf{x} , $\mathbf{fx}=f(\mathbf{x})$

	initial (high) temperature:	temp
Annealing schedule:	annealing temperature reduction factor:	tfactor (<1)
	number of steps per block:	nsteps
	'ad hoc' parameter for trial move:	scale

```
DO WHILE (temp > 1E-5) ! anneal cycle
```

```
  DO istep = 1, nsteps
```

```
    CALL RANDOM_NUMBER(rand) ! generate 2 random numbers; dimension(2) :: rand
```

```
    x_new = x + scale*SQRT(temp)*(rand(1) - 0.5) ! stochastic move
```

```
    fx_new = func(x_new) ! new object function value
```

```
    IF (EXP(-(fx_new - fx)/temp) > rand(2)) THEN ! success, save
```

```
      fx = fx_new
```

```
      x = x_new
```

```
    END IF
```

```
    IF (fx < fx_min) THEN
```

```
      fx_min = fx
```

```
      x_min = x
```

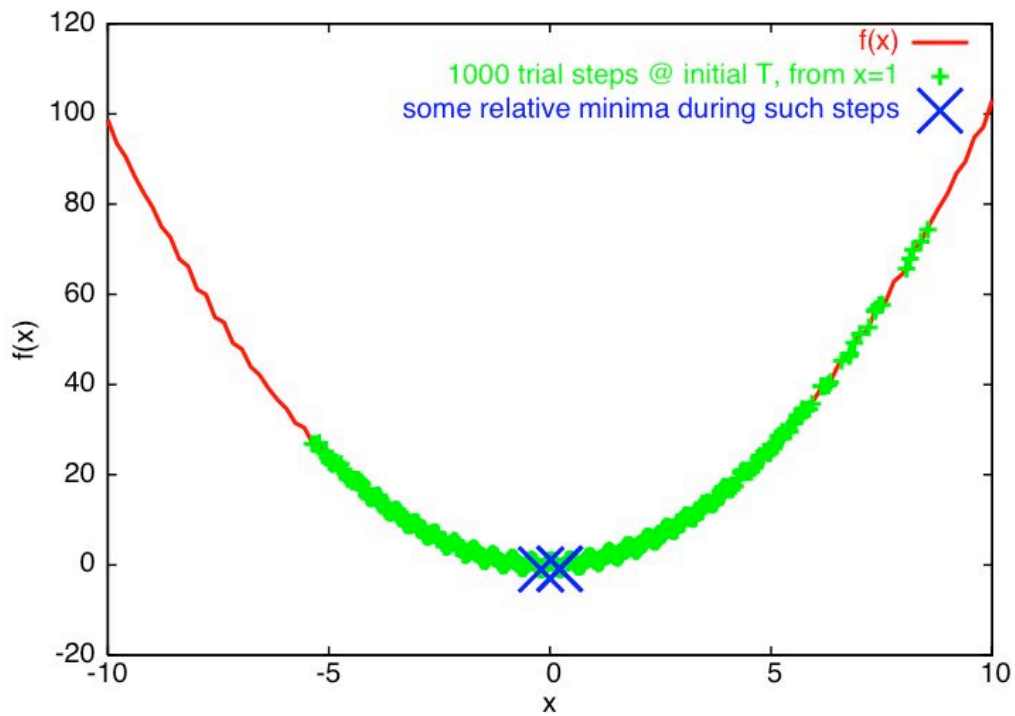
```
      PRINT '(3ES13.5)', temp, x_min, fx_min
```

```
    END IF
```

```
  END DO
```

```
  temp = temp * tfactor ! decrease temperature
```

```
END DO
```

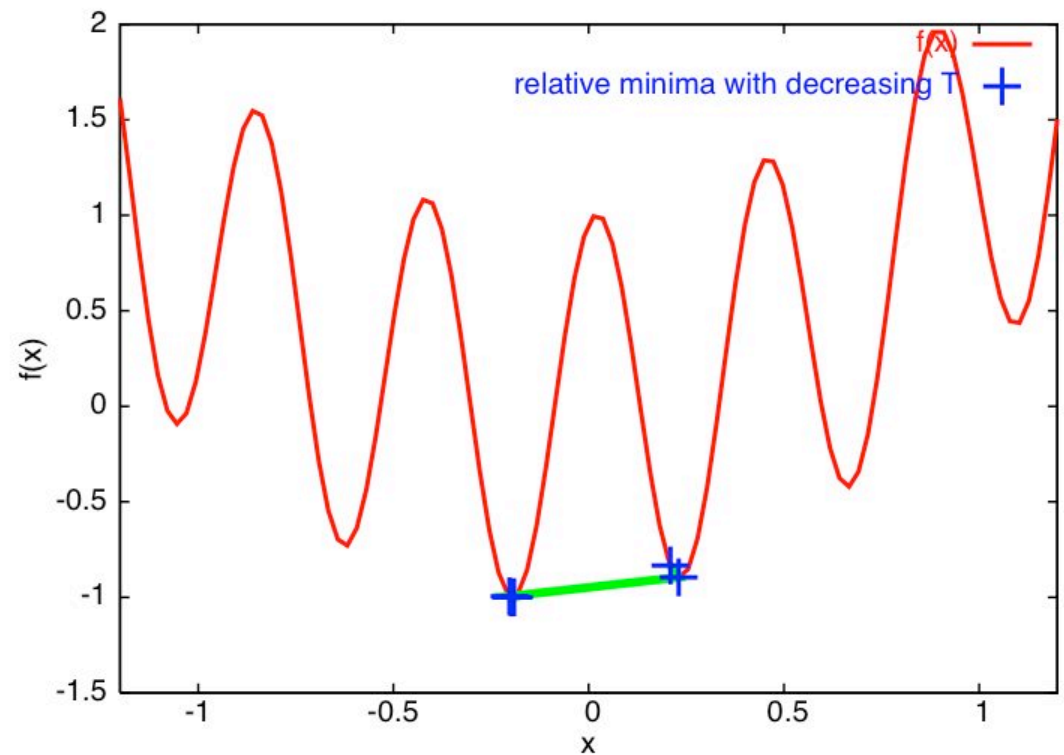


```

initial T: 10 (KB units)
initial x: 1.000000
initial f(x): 1.137208
  
```

```

final T: 2.50315E-01
final x: -1.95067E-01
final f(x): -1.00088E+00
  
```



Some programs:

on

`$/home/peressi/comp-phys/VIII-canonical/`

`[do: $cp /home/peressi/.../VIII-canonical/* .]`

boltzmann_metropolis.f90

(already given for Lect.VII)

box.f90

entropy.f90

simulated_annealing.f90