Università degli studi di Trieste

Corso di laurea magistrale in Fisica della materia

Laboratorio di Fisica Computazionale

Unit VII
Metropolis - Monte Carlo algorithm.
Gaussian and Boltzmann distributions

Davide Bernocco
Matricola s285296

Programming language used:
Python

Contents

1 Random numbers with gaussian distribution: Metropolis algorithm

The focus in this unit will be the algorithm proposed by Metropolis &*Co* and its main applications.
It is a *Makov Chain Monte Carlo* method used to obtain a sequence of random samples from a probability distribution $p(x)$ for which direct sampling could be difficult. The key idea is to implement a 1D *Random Walk* with points x_i whose distribution $\pi_N(x)$ asymptotically approaches the desired $p(x)$ after a large number on steps. For clarity sake, we report the general Metropolis algorithm scheme that will be adopted[1]:

- Choose a function $f(x)$ proportional to $p(x)$
- Pick an arbitrary point $x_{t=0}$ to be the first element of the sample and choose a "jumping" probability density function $g(x_t|x*)$ from which we will pick the next sample point $x*$ (given the actual one). A Usual choice is to let $g(x_t|x*)$ be the gaussian $\mathcal{N}(x_t, \delta^2)$ or the *uniform* distribution$unif(x_t+\delta, x_t-\delta)$ where the parameter δ plays the role of "maximum step size". Here we use the uniform distribution.[2]
- *Accept/reject test*
  - Generate a random number from a *uniform distribution* $w \in [0, 1[$
  - If $w \leqslant \alpha$, then *accept* the candidate setting $x_{t+1} = x*$
  - If $w > \alpha$, then *reject* the candidate setting $x_{t+1} = x_t$
- *Iterate* for a large N of steps

We are now going to study in more detail the primary features of algorithm through an easy example. The target distribution is $e^{-x^2/2\sigma^2}$ (slightly different from a proper gaussian because of the normalization), with $\sigma = 1$ set for simplicity.

1.1 Generated distribution for different n

Let us vary the number of points n in order to qualitatively check when the agreement of the generated sequence distribution start to be satisfactory when compared to the expected $p(x)$.
The initial parameters are set to $x_0 = 0$ and $\delta = 5\sigma = 5$.

It is just from $n = 10000$ that the histogram shape tends to the typical symmetrical bell-shaped form of the $p(x)$ distribution. However, if we wanted to establish a more rigorous criterion, we could look at the fit parameters for μ and σ and accept just the cases which gives accordance within one standard deviation for both of them. This confirm anyway our previous guest, since:

$$\mu_{10^3} = -0.2 \pm 0.1 \text{ and } \sigma_{10^3} = 0.97 \pm 0.08$$
$$\mu_{10^4} = -0.02 \pm 0.02 \text{ and } \sigma_{10^4} = 1.01 \pm 0.02$$

1.2 Dependence of the acceptance ratio on δ/σ

Keeping fixed the number of sampled points of 10^4 just identified, we are now going to shine a light on the parameter δ and its relationship with the *acceptance ratio*.The desired acceptance rate depends on the target distribution, however it has been theoretically shown that the ideal acceptance rate for a generic 1D distribution lies between $\approx 1/3$ and $\approx 1/2$.[3]
Finding the appropriate value is a peculiar aspect of the algorithm and can be crucial for its success: if δ is too large, just a tiny part of the trial points will be accepted (the proposals are likely to land in regions of much lower probability density, so α will be very small), while if it is too small, the Markov chain will move around very slowly. Anyway, both cases would result in a slower convergence towards the desired $p(x)$.

[1] See "Metropolis - Hastings algorithm" on Wikipedia

[2] In order for the stationary Markov Chain distribution $\pi(x) \to p(x)$ to exist, a sufficient but not necessary condition is *detailed balance*: for every pair of states x, x', the probability of being in state x and transitioning to state x' must be equal to the probability of being in state x' and transitioning to state x. The uniform and gaussian distribution both satisfy this requirement.
Furthermore, the uniqueness of the stationary distribution is guaranteed by the *ergodicity* of the Markov process

[3] See again Wikipedia

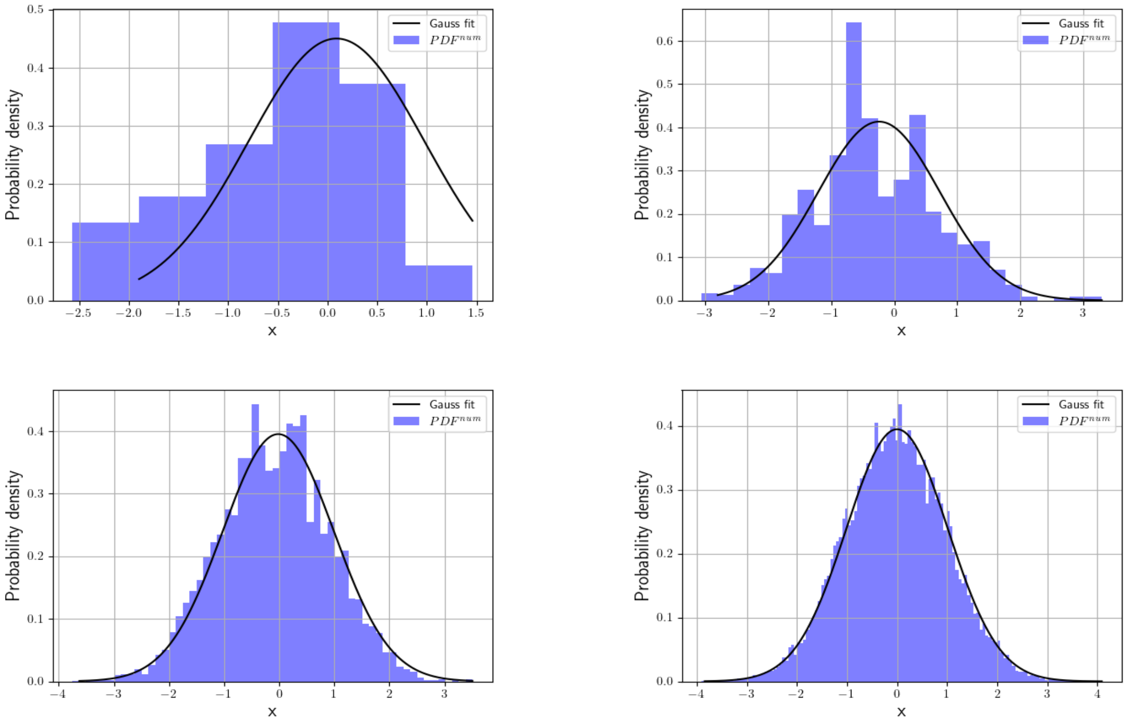

Fig. 1.1 Simulated probability distribution and expected trend for a) $n = 10^2$, *b)* $n = 10^3$, *c)* $n = 10^4$, *d)* $n = 10^5$.

The plot in fig. 1.2 gives us an idea of the values of δ that correspond to the ideal range of acceptance:

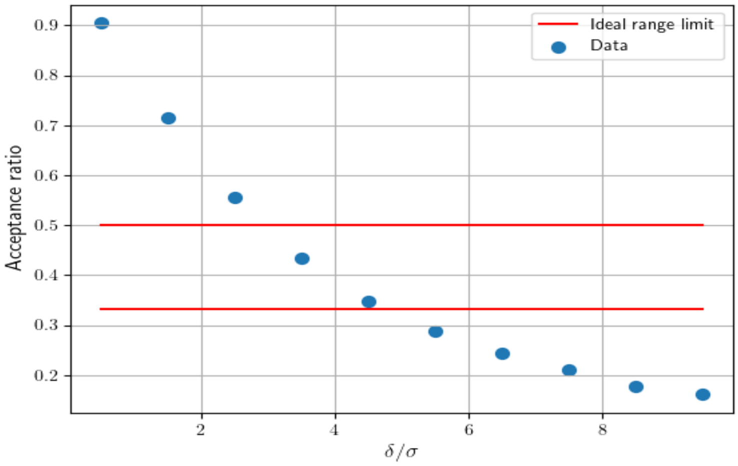

Fig 1.2 Acceptance rate as function of δ/σ

In this specific case, it turns out to be approximately:

$$\delta_{ideal}/\sigma \in [3, 5]$$

1.3 Thermalization time and starting point

As we already mentioned, in order for the algorithm to give a reasonable sampling, a large number n of points has to be produced. Hence, a question arises quite spontaneously: how large n should actually be? We would like, indeed, the produced sequence "looses memory" about the initial value x_0 and its distribution starts converging towards the desired one. A simple way to fix this problem is to let the algorithm run for a while before starting to collect actual samples. The length of time needed to perform these "stabilizing" steps is known as *thermalization time* or *burn in period*.

Depending on the specific case, there could be more than one possible criterion to estimate the burn in sequence length. Here we look at the numerical variance of the sample and compare it with the expected value of σ^2, checking whether they are close enough: the threshold we set for the relative error is 5%.

In Fig 1.3 we can see the relative error evaluated on subsets of increasing size of a single Metropolis run with $n_{max} = 10^4$ and averaged out on a total of 10000 runs with different seeds. The value of $n = 2700$ seems to fit our condition: from that point on, all the calculated numerical variances are smaller than σ^2_{exp}.
Clearly the inclusion of the equilibration phase in a Metropolis algorithm gets more and more important as the number of sampled point is lowered (e.g for efficiency reason in more complex circumstances). On the opposite, as will follow in the rest of this unit, if $n >> n_{Burn-in}$ the bias introduced by the "not-burnt" initial sequence will be negligible.

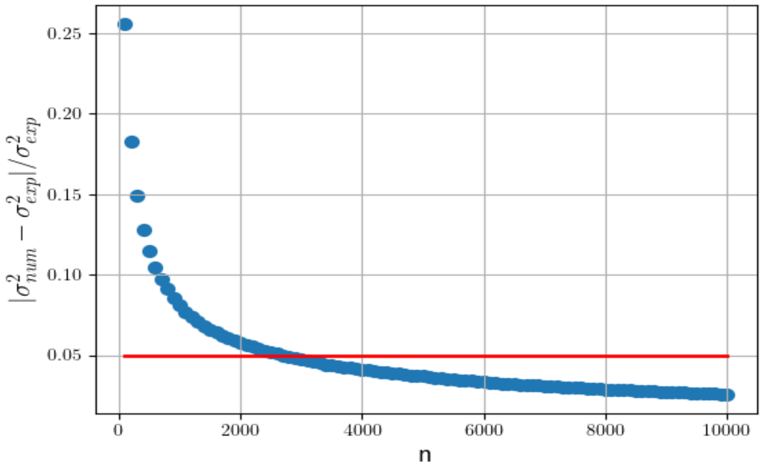

Fig 1.3 According to the "variance criterion", $n = 2700$ emerges as a good approximation of the thermalization time for a gaussian target distribution

2 Correlations

For how the algorithm is built (RW + acceptance test), a peculiar characteristic of Markov Chain Monte Carlo sampling is *autocorrelation*. Even though over the long term the distribution of points do correctly follow $p(x)$, a set of nearby samples will be correlated with each other and thus not correctly reflect the distribution. This must be taken into account for a correct error determination since correlations can affect the variance estimation, making it appear much smaller than the actual error. Among the error evaluation techniques we studied in unit v, the "block average" is therefore the only one at our disposal to control the reliability of the statistical Metropolis sampling.

In this section, we are going to answer the interesting question: how many MC steps should elapse between two samples so that they can be considered uncorrelated? To do so, we study the *correlation function* $C(j)$[4]:

$$C(j) = \frac{\langle x_i x_{i+j} \rangle - \langle x_i \rangle^2}{\langle x_i^2 \rangle - \langle x_i \rangle^2} \tag{1}$$

where we know:

- $C(j = 0) = 1$ typical, by definition, of every process based on random walks
- $C(j > 0) \to 0$ for $j \to \infty$ if the simulation is *ergodic*, as in this case

In particular, we expect an exponential decay of $C(j)$ with characteristic parameter τ, where only points separated by $2\tau - 3\tau$ can be actually considered statistically independent.[5]

In *Fig 2.1* are shown correlations for different values of δ/σ. The first thing we notice is all of them seems to satisfy the expected exponential behaviour. Secondly, as $j \to \infty$ they approach zero with different "speeds": the curves associated to values of C close enough to the ideal range of $[3, 5]$ (see §1.2) seems to converge to zero more quickly, with a resulting smaller τ. In conclusion, not only "too small" or "too large" values of δ/σ lead to a slower algorithm convergence as explained in §1.2, but we have now showed they bring to highly correlated chains.

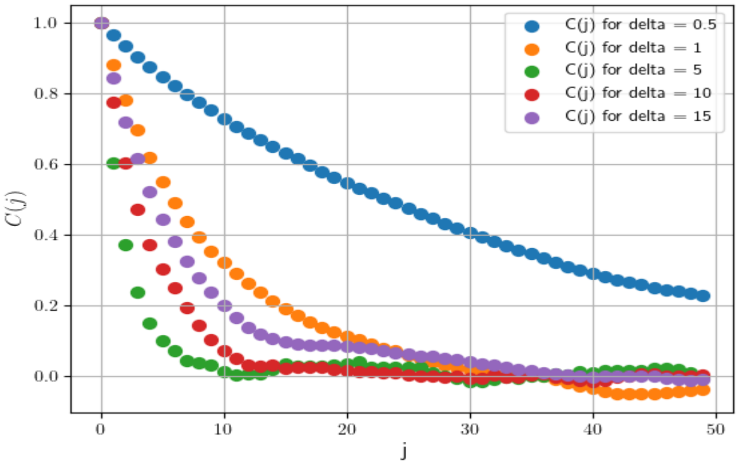

Fig 2.1 Correlation vs j for different values of δ/σ ($n = 10^5$)

To conclude this section, we are going to further explore this peculiar characteristic of MCMC methods. In *Fig 2.2* is displayed the correlation behaviour for two set of random points whose asymptotic distribution is the same, but sampled with two distinct Monte Carlo algorithms: the MCMC and Box-Muller. Because of its easier structure, the Box-Muller technique does not show any relevant correlation of any kind, making it preferable to use in this case.

Hence, as a rule of thumb, if for single-dimensional distributions there are usually other MC methods that

[4] $\langle ... \rangle$ represents the average over the generated random sequence

[5] i.e. during the error estimation with block averages, it does not make any sense using blocks of that size or less!

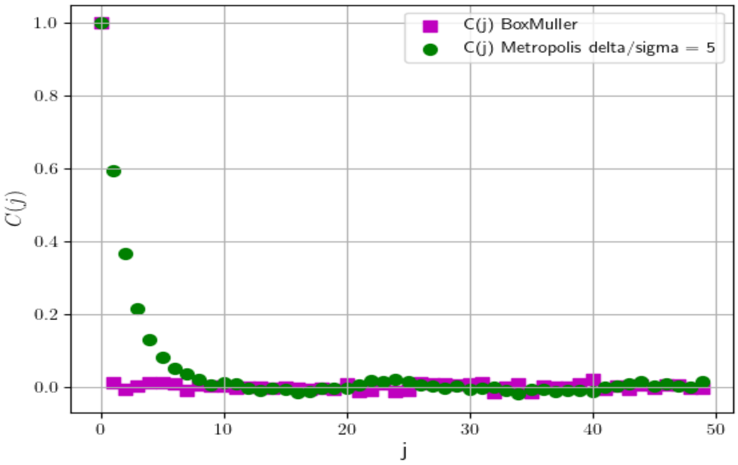

Fig 2.2 Comparison between the correlations of two different Monte Carlo sampling algorithms: Box-Muller and Metropolis ($n = 10^5$)

can directly return independent samples free from the problem of autocorrelation, MCMC algorithms are generally best used for sampling from multi-dimensional distributions.

3 Sampling physical quantities: direct sampling and Metropolis sampling (optional)

In this section and in the following one, we are going to show another fundamental application of the Metropolis algorithm, that is the numerical evaluation of integrals. The aim will be to generate a distribution $p(x)$ so to efficiently sample integrals of the form:

$$\langle f \rangle = \frac{\int f(x)p(x)dx}{\int p(x)dx} \tag{2}$$

In particular, in this section we focus on finding *expectation values* of physical quantities such as kinetic, potential and total energy in the framework of the quantum theory, and more specifically in the case of quantum harmonic oscillator.

Given the Hamiltonian $\mathcal{H} = -\frac{1}{2}\nabla^2 + \frac{1}{2}x^2$ and the ground state wave function $\phi(x) = Ae^{-x^2/4\sigma^2}$ it follows:

$$\langle E_{Pot} \rangle = \frac{\langle \phi | \frac{1}{2}x^2 | \phi \rangle}{\langle \phi | \phi \rangle} = \frac{\int \frac{1}{2}x^2 |\phi(x)|^2 dx}{\int |\phi(x)|^2 dx} = \frac{\sigma^2}{2} \tag{3}$$

$$\langle E_{Kin} \rangle = \frac{\langle \phi | -\frac{1}{2}\nabla^2 | \phi \rangle}{\langle \phi | \phi \rangle} = \frac{\int (\frac{1}{4\sigma^2} - \frac{x^2}{8\sigma^4}) |\phi(x)|^2 dx}{\int |\phi(x)|^2 dx} = \frac{1}{8\sigma^2} \tag{4}$$

This easy example not only provides us a testing ground where analytical solution are well known, but also allows us to understand more clearly why the Metropolis algorithm is particular convenient in such cases.
We are thus going to study the *actual errors* as function of n for the examined quantities: since they all depend on the second momentum $\langle x^2 \rangle$ we expect the relative Δ_n to follow the same behaviour, a part from scaling factors.

3.1 Direct sampling

We start by evaluating the integrals with a *sample mean* algorithm (as shown in unit v), using instead of uniformly generated random points, a sequence of gaussian distributed points (as of "BoxMuller" algorithm in unit III).

As we learnt in unit V, the actual error of Monte Carlo integration methods decreases as $n^{-1/2}$ with n. In this case, since we need to deal with both the integrals at the numerator and denominator of equations 3 and 4, it is logical expecting a constant trend (a part for the different scaling factors). The plot in *Fig. 3.1* qualitatively confirms our guess.

3.2 Metropolis sampling

Let us now focus on the heart of this section. The goal is the same as declared in §3.1, but we are going to follow a slightly different strategy. The key idea is to use the *importance sampling* technique (always seen in unit v), exploiting the advantages of the Metropolis algorithm.
You may remember the importance sampling is a MC method use to evaluate integrals in the form:

$$\int_a^b f(x)dx = \int_a^b \frac{f(x)}{p(x)} p(x)dx = \langle \frac{f(x)}{p(x)} \rangle \int_a^b p(x)dx \tag{5}$$

$$\langle \frac{f(x)}{p(x)} \rangle \approx \frac{1}{n} \sum_{i=1}^{n} \frac{f(x_i)}{p(x_i)} \tag{6}$$

with x_i distributed according to an easy-integrable $p(x)$, as much as possible *similar* to $f(x)$ in the behaviour.

Now, what if we used Metropolis algorithm to generate a sequence of random points *exactly* distributed with $p(x)$ and concentrate on evaluating $\langle A(x) \rangle_{f(x)}$?

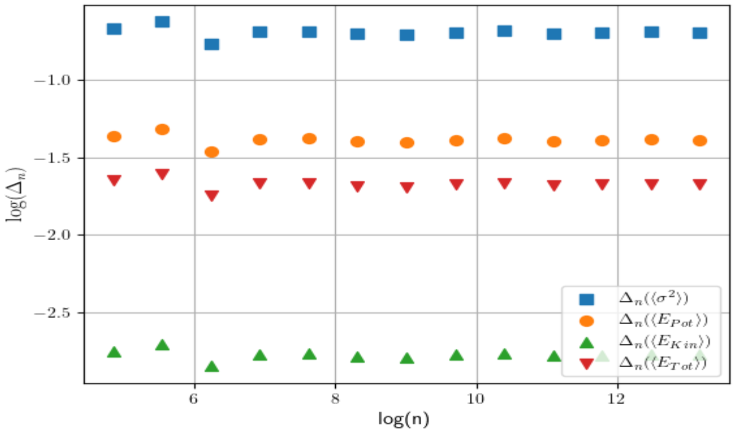

Fig 3.1 Actual error for variance, potential energy, kinetic energy and total energy of a quantum harmonic oscillator. Sample mean method

$$\langle A(x)\rangle_{f(x)} = \frac{\int A(x)f(x)dx}{\int f(x)dx} = \frac{\int A(x)\frac{f(x)}{p(x)}p(x)dx}{\int \frac{f(x)}{p(x)}p(x)dx} = \frac{(\int p(x)dx)\langle A(x)\frac{f(x)}{p(x)}\rangle}{(\int p(x)dx)\langle\frac{f(x)}{p(x)}\rangle} \tag{7}$$

which, according to *Eq.(6)* , leads to:

$$\langle A(x)\rangle_{f(x)} \approx \frac{1}{n}\sum_{i=1}^{n} A(x_i) \tag{8}$$

with x_i distributed according to $p(x) \equiv f(x)$

This method should provide a much more efficient way to calculate such integrals. Indeed, contrary to what happened before, since here there is just one Monte Carlo sampling involved, we expect the typical $\mathcal{O}(n^{-1/2})$ behaviour for the actual error.
If we get back on our quantum HO example, and set $p(x) = |\phi(x)|^2$, that is exactly what we find:

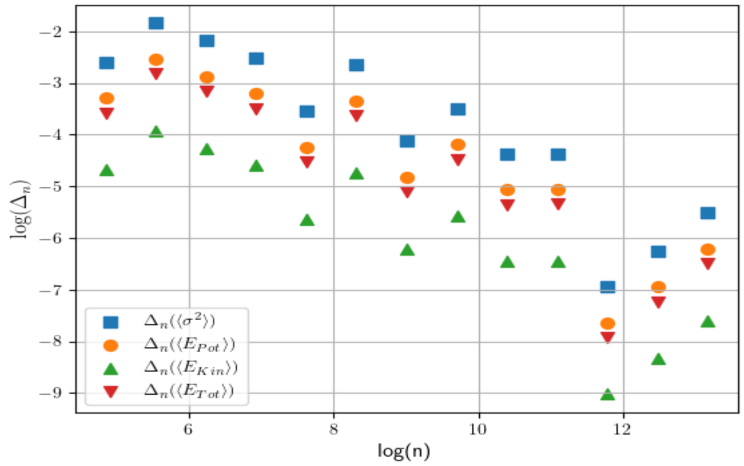

Fig 3.2 Actual error for variance, potential energy, kinetic energy and total energy of a quantum harmonic oscillator. Metropolis sampling method

4 Verification of the Boltzmann distribution: canonical ensemble

If the topic of the previous section was the quantum HO, here we are going to delve into the evaluation of *ensemble averages* in the canonical ensemble, using Metropolis algorithm to produce microsates with the typical Maxwell-Boltzmann distribution.

$$\langle A\rangle \approx A_n = \frac{\sum_{s=1}^{n} A_s e^{-\beta E_s}}{\sum_{s=1}^{n} e^{-\beta E_s}} \xrightarrow[\text{with Metropolis}]{\text{importance sampling}} \frac{1}{n}\sum_{s=1}^{n} A_s \tag{9}$$

More specifically, we are going to study first a single classical particle in 1D in thermal equilibrium with a heat bath, then an ideal gas of N particles in the same conditions.

4.1 Single classical particle in 1D

Let us consider a particle moving randomly in 1D: the distribution of its (one component) velocity v_x is $f(v_x)$:

$$f(v_x) = \sqrt{\frac{m}{2\pi k_B T}} e^{-mv_x^2/k_B T} \tag{10}$$

where $f(v_x)dv_x$ is the probability of finding the particle with a velocity in the infinitesimal element dv_x about v_x.
Then, since the system under exam is composed by only one free particle, in order to label all the different microstates of the canonical ensemble that can be generated by randomly varying the velocity, we can use velocity itself. For this reason, in this case, $f(v_x)$ could be interpreted as well as a "microstate distribution". The same goes for the energy E, a part a factor of 2 due to the redundancy of the sign in evaluating it: indeed $E = \frac{1}{2}mv_x^2$. As a consequence, defining the speed as $v \equiv |v_x|$ and indicating the energy distribution with the letter $P(E)$[6], it holds: $2f(v)dv = P(E)dE$.
This leads to:

$$P(E) = \frac{1}{\sqrt{\pi k_B T}} \frac{1}{\sqrt{E}} e^{-E/k_B T} \tag{11}$$

Figure 4.1 and 4.2 show respectively the distributions $f(v_x)$ and $P(E)$ generated with the Metropolis algorithm with $T = 1$, $v_0 = 0$, $\delta_v = 2$ and $n = 10^5$ ($m = 1$, $k_B = 1$ for convenience).

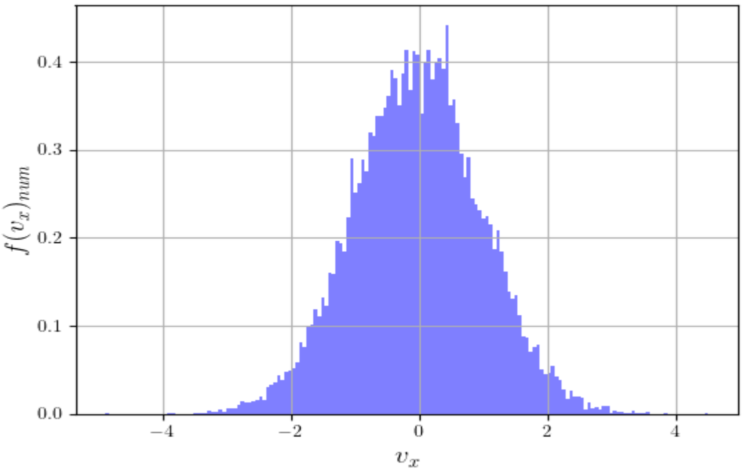

Fig 4.1 Velocity distribution for a single free particle in 1D. Metropolis sampling method

[6] To underline it is a different functional form from $f(v)$.

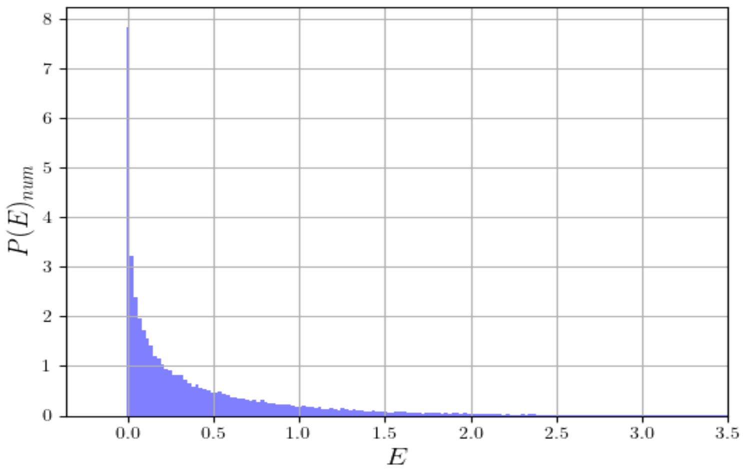

Fig 4.2 Energy distribution for a single free particle in 1D. Metropolis sampling method

The obtained shape of $P(E)$ is compared in figure 4.3 with the theoretical expected one (eq. 10) for a qualitative check: the agreement is well satisfying.

What about $\langle v_x \rangle$ and $\langle E \rangle = \frac{m}{2}\langle v_x^2 \rangle$? A couple of quite straightforward analytical calculations bring to:

$$\langle v_x \rangle = \frac{\int_{-\infty}^{+\infty} v_x f(v_x) dv_x}{\int_{-\infty}^{+\infty} f(v_x) dv_x} = 0 \tag{12}$$

$$\langle E \rangle = \frac{\frac{1}{2}\int_{-\infty}^{+\infty} v_x^2 f(v_x) dv_x}{\int_{-\infty}^{+\infty} f(v_x) dv_x} = \frac{k_b T}{m} \tag{13}$$

We are going to numerically evaluate these two quantities and compare the result with the expected ones. Furthermore, to give a proper numerical error estimation, the *block average* method is used in order to avoid problems with sampling autocorrelation ($s = 100$ blocks here):

$$\langle v_x \rangle_{num} = 0 \pm 0.005 \quad m/s$$
$$\langle E \rangle_{num} = 0.50 \pm 0.01 \quad J$$

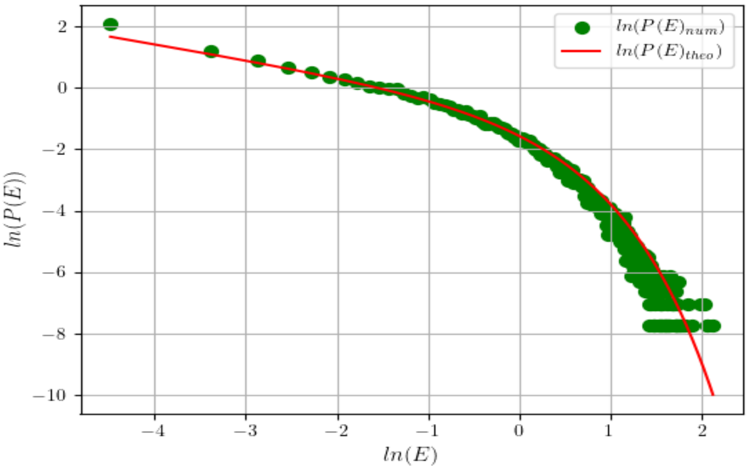

Fig 4.3 Comparison between numerical and theoretical behaviour of $P(E)$ (log-log scale)

4.2 Ideal gas of N particles in 1D

The same features investigated before, can be studied in the case of a system of N free particles in thermal equilibrium, with energy E and "energy per particle" $\epsilon = E/N$.
Each system configuration carries an energy $E = \sum_{i=1}^{N} \frac{1}{2} m v_{x,i}^2$, and since there are usually several microstates with the same energy, E cannot be used anymore to label them. As a result, the histograms of Metropolis samplings cannot be interpreted as a distribution of microstates: $P(E)$, or better, $P(\epsilon)$ is the *probability distribution* of the energy per particle.

We start by fixing $T = 100$K, $N = 200$, $v_0 = 10$m/s and $n_{MCsteps} = 10^5$, varying δ_v within the Metropolis algorithm until the acceptance ratio reaches the ideal value of $\approx 1/2$. For this particular case we find $\delta_v = 30$ (This parameter, however, depends on T, and this will be taken into account for future application).

Now, what are the *mean particle velocity* $\nu = \langle v_x \rangle$ and the *mean particle energy* $\langle E \rangle / N$? As before, we can provide a numerical estimation of these ensemble averages through the importance sampling integration technique. The error is given by the block averages ($s = 1000$ here):

$$\langle v_x \rangle_{num} = -0.003 \pm 0.003 \quad m/s$$
$$\langle E \rangle_{num} / N = 49.99 \pm 0.06 \quad J$$

These two values must be compared with the theoretical ones coming from the usual ensemble averages: $\langle v_x \rangle = 0$ and $\langle E \rangle = N \frac{k_B T}{2}$, where $\langle E \rangle$ is the average over all the possible microstate energies E. Beside calculus, the reason behind these results can be as well intuitively grasped: even though they are all initialized to the same value, since there are no preferential directions (possible if there was an external field, for instance), after a proper *equilibration time* the particle velocity will distribute symmetrically around the zero according to the the gaussian density function. As for the mean particle energy, this is a direct consequence of the *equipartition theorem*, that assigns the average kinetic energy of $k_B T/2$ to each particle degree of freedom in an equilibrium system.

Figure 4.4 shows $P(\epsilon)$ at fixed temperature for increasing N. We see that for $N = 1$ the histogram confirms what we found previously in the case of a single free particle. But what strikes more is that, for different Ns, the shape of the "energy per particle" distribution changes, passing from highly asymmetric profiles for lower values to the typical bell-shaped form at large N. Why does it happen? Is is compatible with theoretical expectations?

To answer the question, we should reconsider the physical quantity whose distribution we are focusing on, that is the "microstate energy per particle" $E/N = \frac{1}{N} \sum_{i=1}^{N} \frac{m}{2} v_{x,i}^2$.
Since we know v_x is normally distributed, the same will happen for the variable $\sqrt{\frac{m}{2}} v_x$. Now comes the heart of the story: by definition we know that[7]:

Given a standard normal deviate - that is a random sample from the standard normal distribution -, the Chi Square distribution is the distribution of the sum of squared standard normal deviates. The degrees of freedom of the distribution is equal to the number of standard normal deviates being summed. The χ^2 probability density function is

$$f(x;k) = \begin{cases} \frac{1}{2^{k/2}\Gamma(\frac{k}{2}} x^{\frac{k}{2}-1} e^{-x/2} & \text{if x>0} \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

It correspond exactly to the physical situation we are studying!
If we then compare the definition of $f(x; k = 1)$ with the well known $P(E)$ from the previous chapter, we are able to turn the expression in x into the probability density function of ϵ for an arbitrary N[8]:

$$P_N(\epsilon) \equiv P(N\epsilon) = \frac{N^{\frac{N}{2}-1}}{\Gamma(\frac{N}{2})} \frac{1}{(k_B T)^{N/2}} \epsilon^{\frac{N}{2}-1} e^{-N\epsilon/k_B T} \tag{15}$$

[7] See "Chi-squared distribution" on Wikipedia

[8] x correspond here to the normalized energy $E/\langle E \rangle = \frac{2E}{k_B T}$. Furthermore it must hold: $P(\epsilon N) d\epsilon = P(E) dE = f(x;k) dx$, with $k \mapsto N$

Asymptotically, for the χ^2 distribution it goes that:

By the central limit theorem, because the chi-squared distribution is the sum of N independent random variables with finite mean and variance, it converges to a normal distribution for large N.

Here it is finally explained what happens in Fig 4.4: as N grows, the energy per particle distribution $P(\epsilon) \sim \chi^2$ gets more and more symmetric, approaching a gaussian distribution centered in $\langle E \rangle / N$.

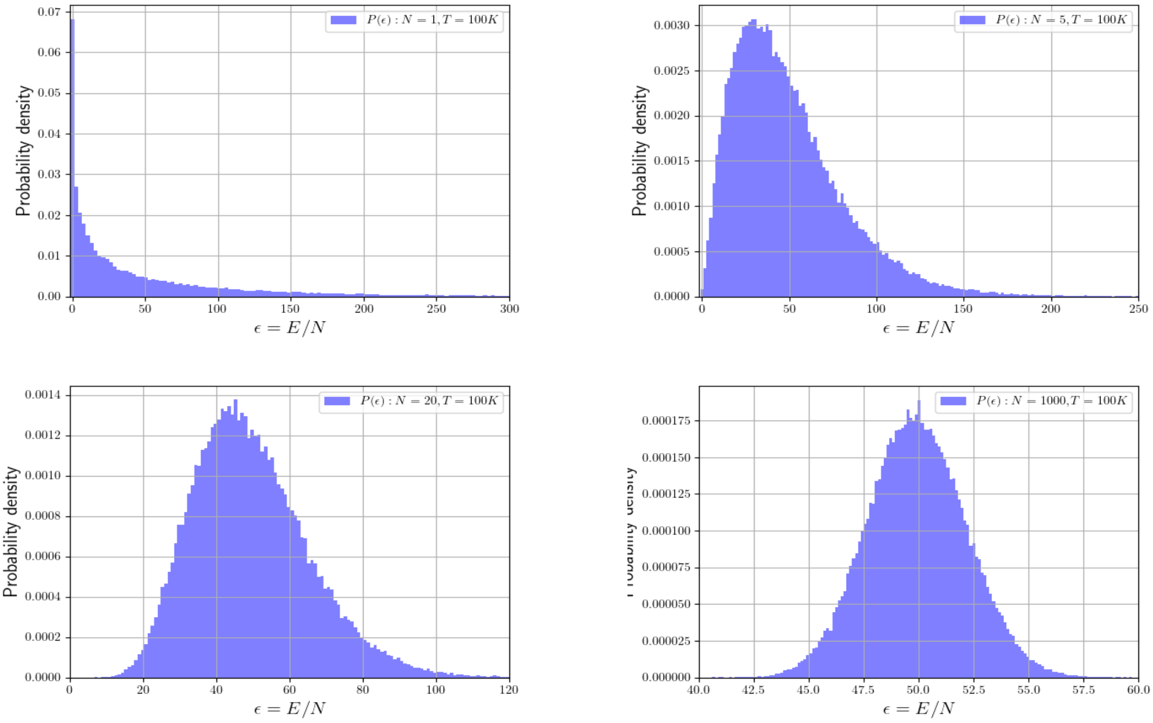

Fig. 4.4 Energy per particle probability distribution at $T = 100K$ for an ideal gas of a) $N = 1$, b) $N = 5$, c) $N = 20$, d) $N = 1000$ particles.

There is still an interesting observation we can make about the considered system, if we look at these histograms.
Since the particles do not interact with each other (and hence cannot exchange energy directly), we might expect the total energy of an ideal gas to remain constant. But if this is true, why we measure energies distributed according to a distribution of finite width? If we initialized all the particles with the same velocity of $v_0 = 10m/s$, how should we be able to measure energies per particle different from $\frac{1}{N}\sum i = 1^N \frac{m}{2} v_{x,i}^2 \equiv \frac{m}{2} v_{x,i}^2$?

The answer is quite simple: a constant temperature does not imply system's energy is constant. Since our particles are in contact with a thermal reservoir, energy is being exchanged all the time in such a way that the temperature and so the *average energy* (remember the equipartition theorem!) are constant. Furthermore, Statistical mechanics tells us that, within the *canonical ensemble*, the random energy fluctuations about $\langle E \rangle$ of a macroscopic system are so small that can be negligible. In particular, taking the "root mean square fluctuations" ΔE_{rms}:

$$\Delta E_{rms} \equiv \sqrt{\langle \Delta E^2 \rangle} = \sqrt{\langle E^2 \rangle - \langle E \rangle^2} \tag{16}$$

it can be demonstrated that:

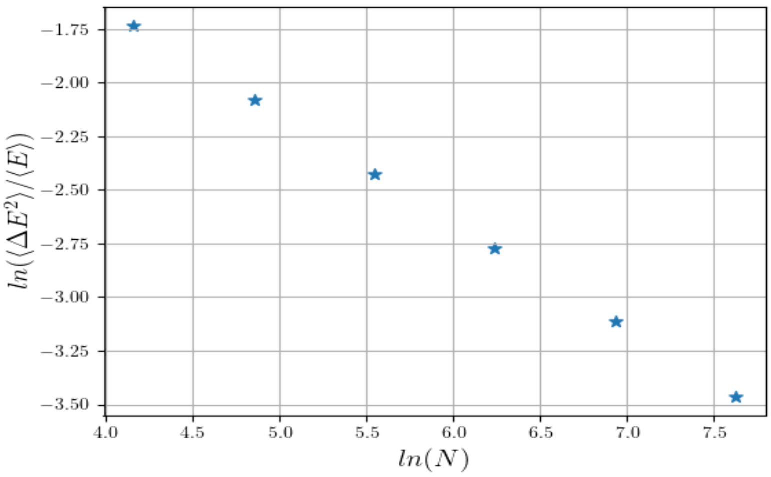

Fig 4.5 Relative energy fluctuations for an ideal gas of N particles (log-log scale).

$$\frac{\Delta E_{rms}}{\langle E \rangle} \sim \frac{1}{\sqrt{N}} \tag{17}$$

In Fig 4.5 we computed this quantity for different N and what we found is a striking qualitative accordance.

To conclude, we report a row numerical estimation of the *heat capacity* $C = \partial \langle E \rangle / \partial T$, that for an ideal gas we expect it to be constant.
Briefly, we fixed three equally spaced points in T quite close to each other and calculated the local value of the derivative:

$$C = \frac{\partial \langle E \rangle}{T} \approx \frac{\langle E \rangle (T + \Delta T) - \langle E \rangle (T - \Delta T)}{2 \Delta T} \tag{18}$$

We then fixed other points ($T = 10, 20, 30, 90, 100, 110$ K) in different regions and repeated the same operation, collecting these four values compatible with each other within the 5 *permille*:

Heat capacity
100.00
100.14
99.31
100.44