Random Walks simulation - Report

Giovanni Varutti

November 12, 2023

Contents

1 Exercize 1

The aim of this exercize is to write a code that simulates numerically a 1D random walk with a Monte Carlo approach and gives the final position x_N (and x_N^2) after N steps with fixed length l and probabilities $p_{\leftarrow}$ and $p_{\rightarrow}$ of moving left and right. Without any loss of generality, has been considered $x_0 = 0$ as starting position, $l = 1$ as step length, $p_{\leftarrow} = p_{\rightarrow} = 0.5$ for all the simulations performed. For this purpose, the code has been written using *FORTRAN90*, with the usage of *openMP* for parallelizing some loops and speeding the code up. In order to follow the evolution of a random walk with the number of steps, we can plot the instantaneous positions, i.e., x_i and x_i^2 vs. i, with i from 0 to N, for few different seeds.

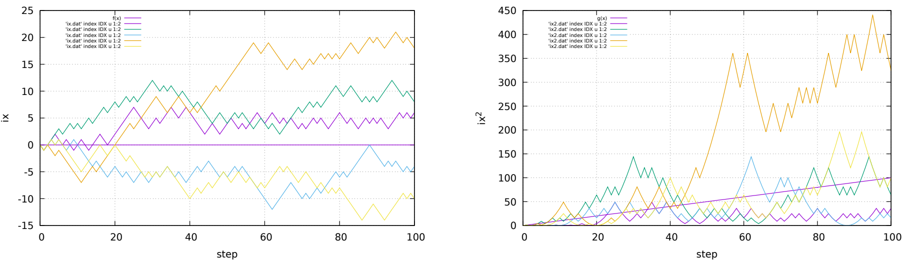

Figure 1: plot of instantaneous positions for five runs, with $N = 100$ steps. As expected, the lines diverge from the 0 line as the number of steps grow.

We obtain, of course, different walks for different seeds. In order to check if the simulation follows the theoretical behaviour, it is necessary to calculate the averages over many walkers of the instantaneous quantities $\langle x_i \rangle$, $\langle x_i^2 \rangle$ and $\langle \Delta x_i^2 \rangle$ (and their final values). We expect a constant behaviour (close to $y = 0$) for $\langle x_i \rangle$, and a linear behaviour ($\propto i$) for $\langle x_i^2 \rangle$ and $\langle \Delta x_i^2 \rangle$.

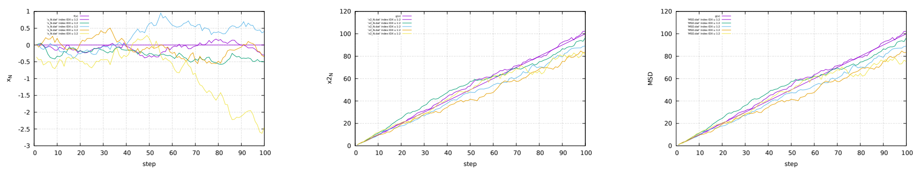

Figure 2: averages over many walkers: from 400 walkers to 80 walkers at steps of 50, with $N = 100$

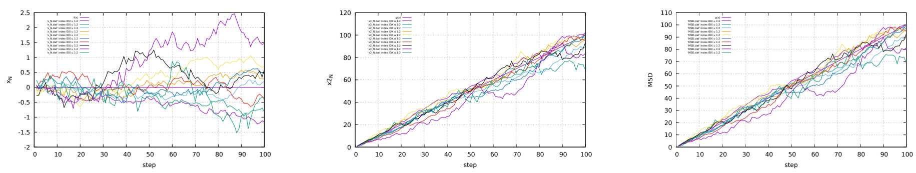

Figure 3: averages over many walkers: from 200 walkers to 20 walkers at steps of 20, with $N = 100$

As we can see, the simulations have the expected behaviour. In order to check the goodness of the simulation, we can calculate the accuracy Δ of the mean square displacement (after N steps). The following figures show the behaviour of Δ in function of the number walkers:

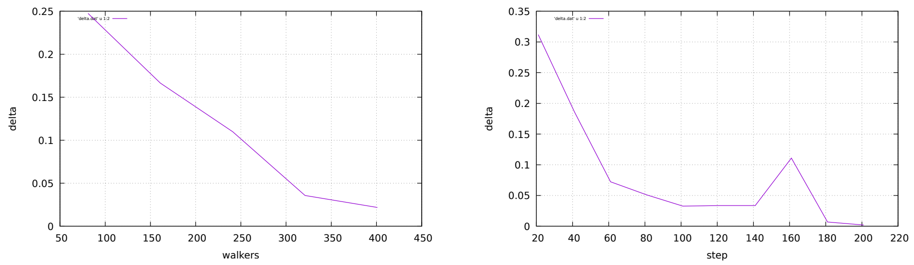

Figure 4: Δ function: from 400 walkers to 80 walkers at steps of 50, and from 200 walkers to 20 walkers at steps of 20, with $N = 100$ fixed

We can see that the larger is the number of walks for the average, the smaller is Δ. However, the Δ function has shown a variable behaviour in function of the number of walkers. For this reason, in order to obtain a relative accuracy $\Delta \leq 5\%$, we have to choose a value at least between 200 and 300 for the number of walkers. For the following points, a value of 200 for the number of walkers has been used. We can now determine the dependence of $\langle \Delta x_N^2 \rangle$ on N, both comparing analytical and numerical results for some values of N, and doing a linear fit using a *loglog* scale. In fact, we expect to see directly a linear behavior: $\langle \Delta x_N^2 \rangle = aN^{2\nu}$. So a linear fit in *loglog* form should give $\nu = 1/2$.

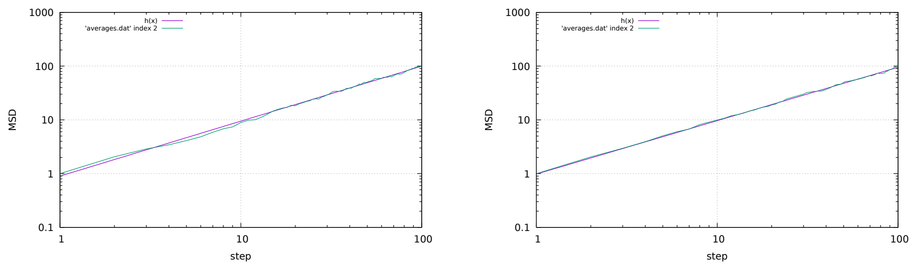

Figure 5: *loglog* fit of the Mean Square Displacement, using 200 and 300 walkers, $N = 100$. In the first case the fit provides an estimation for the parameters $a = 0.901$ and $\nu = 0.511$. In the second case the fit yields $a = 0.981$ and $\nu = 0.497$.

Now we can study the distribution $P_N(x)$ comparing the numerical simulation and the expected behaviour, with $p_{\leftarrow} = p_{\rightarrow} = 0.5$ and $N = 8$.

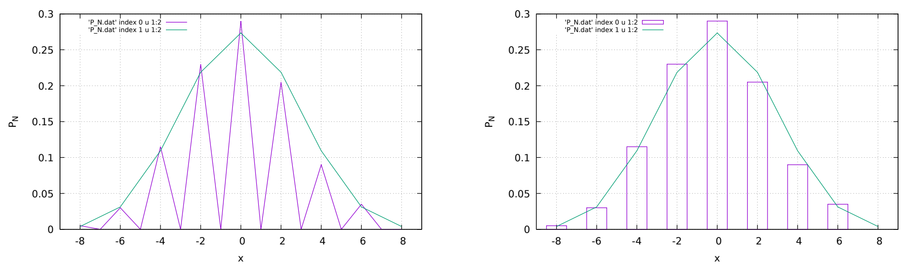

Figure 6: $P_N(x)$ and $P_N^{th}(x)$

As we can see the computed function $P_N(x)$ is not continuous. In particular, there are no cases in which the final position x_N occupies an odd number $\in [-8, 8]$. This is because the walker has to do exactly 8 movements, so there is no possibility to finish in an odd position. This is also confirmed by the expression of the theoretical function $P_N^{th}(x)$, in which the factorials allow only integer numbers (so $(N \pm x)/2$ have to be integer). For sufficiently large N , $P_N(x)$ can be approximated with the gaussian distribution, with $\mu = \langle x \rangle$ and $\sigma = \langle \Delta x^2 \rangle$. This can be verified by calculating numerically and plotting $P_N(x)$ for $N = 8, 16, 32, 64$.

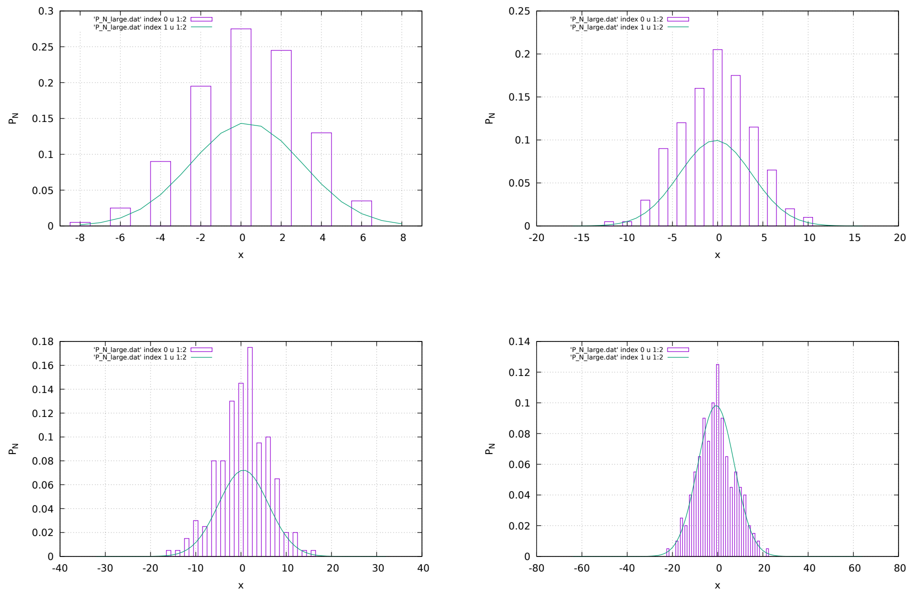

Figure 7: $P_N(x)$ (numerically computed) and $P(x)$ (theoretical gaussian distribution). In the last plot, the function $P(x)$ is multiplied by the factor 2, in order to coincide with the distribution deligned by the histogram.

2 Exercize 2

The aim of this second exercize is to write a program for the numerical simulation of 2D random walks, with equal probabilities of moving in each direction. Also in this case the code has been written using*FORTRAN90*, with the usage of *openMP*. The main task consist in calculating and plotting the Mean Square Displacement $\langle \Delta R_N^2 \rangle = \langle x_N^2 \rangle + \langle y_N^2 \rangle - \langle x_N \rangle^2 - \langle x_N \rangle^2$ in function of N, and do a linear *loglog* fit. This was done using three different algorithms to choose the displacements along $x$axis and $y$axis. The results are shown below (for all the simulations $N = 100$ and the number of walkers is 200):

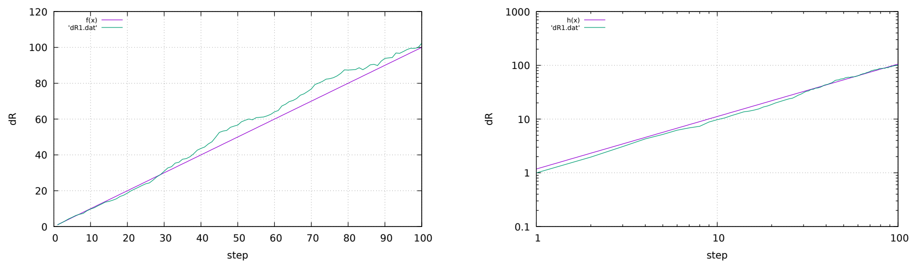

Figure 8: Plot and *loglog* fit of $\langle \Delta R_N^2 \rangle$ in function of $aN^{2\nu}$ with the first algorithm. The estimated parameters are $a = 1.179$ and $\nu = 0.488$. The first algorithm consist in choosing a random number $\theta \in [0, \pi]$ and then set $x = cos(\theta)$ and $y = sin(\theta)$

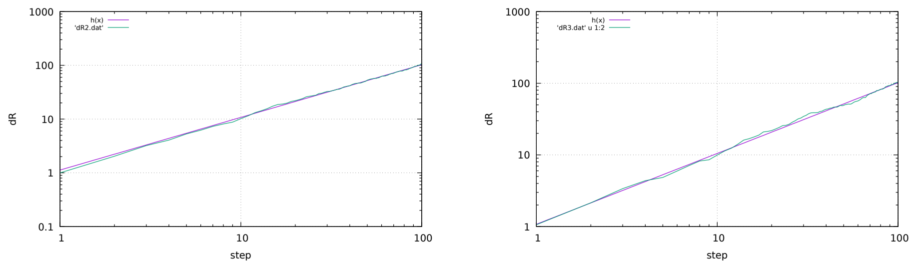

Figure 9: On the left: *loglog* fit of $\langle \Delta R_N^2 \rangle$ in function of $aN^{2\nu}$ with the second algorithm. The estimated parameters are $a = 1.126$ and $\nu = 0.490$. The second consist in choosing separate random values for Δx and Δy in the range $[-1, 1]$, then normalize them in order to obtain the step length $l = 1$.
On the right: *loglog* fit of $\langle \Delta R_N^2 \rangle$ in function of $aN^{2\nu}$ with the third algorithm. The estimated parameters are $a = 1.078$ and $\nu = 0.494$. The third consist in choosing separate random values for Δx and Δy in the range $[-3/2, 3/2]$ (in this case $l = 1$ on average).

As shown by the values of the estimaed parameters (and also the plots), the third algorithm seems to have the better behaviour. The last task for this exercize is to perfom the 2D-random-walks simulation on a discrete lattice grid. In this case, the steps (left, right, up or down) are chosen at random by taking a random number $rnd \in [0 - 1]$. Then, steps to the right are choosen if $0 < rnd < 0.25$, i.e. for $\lfloor rnd * 4 \rfloor = 0$ etc... The results are shown below:

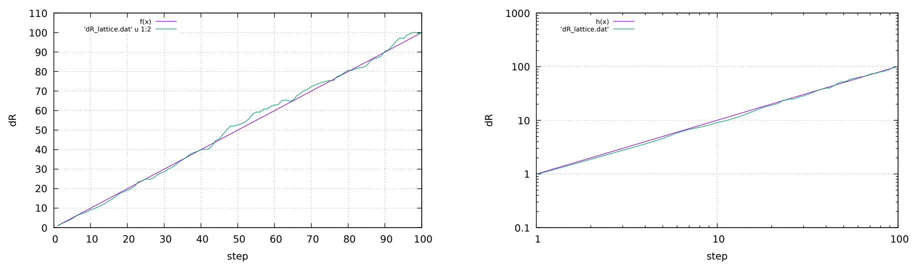

Figure 10: Plot and *loglog* fit of $\langle \Delta R_N^2 \rangle$ in function of $aN^{2\nu}$ in the discrete lattice grid. The estimated parameters are $a = 0.999$ and $\nu = 0.501$.