Homework 1

Francesco Andreuzzi (andreuzzi.francesco@gmail.com)
Data Science and Scientific Computing – Computational Science
Academic Year 2022-2023

April 10, 2023

1 1D random walks

1.1 Simulation strategy

We used the following algorithm to simulate a single RW for N time steps:

```
call random_number(moves)
x(:) = sign(l_vec, p_right - moves)
do i = 2, n
   x(i) = x(i) + x(i - 1)
end do
```

We leverage vectorized operations to achieve some computational speed-up. `l_vec` is simply $[\ell \dots \ell]$. Each $x(i)$ receives the value $\pm\ell$, depending on the sign of the right-most parameter of `sign`, which is positive with probability $p_{\rightarrow}$. The evolution of the body position is then recovered via a cumulative sum.

1.2 Instantaneous position (a)

We take $p_{\rightarrow} = p_{\leftarrow} = 0.5$, $\ell = 1$, and we simulate $N = 1000$ time instants. The instantaneous position x_i computed during the RW is plotted for several bodies with different seeds in Figure 1 (left), along with the theoretical expectation. The same is done for x_i^2 (right). There's clearly an important difference between the theory and the realization of the system, the first three bodies show oscillating behavior around $\langle x_i \rangle$, however the fourth is clearly an outlier.

1.3 Averages (b)

We now consider the averages of x_i and x_i^2 with respect to 10 independent bodies. We chose by purpose a small number of walks, in order to show that the measured averages approach the theoretical expectation as we increase the number of bodies. The result are shown in Fig. 2. The adherence to the theory is pretty poor as expected. The situation improves drastically if we consider 10 times more bodies.

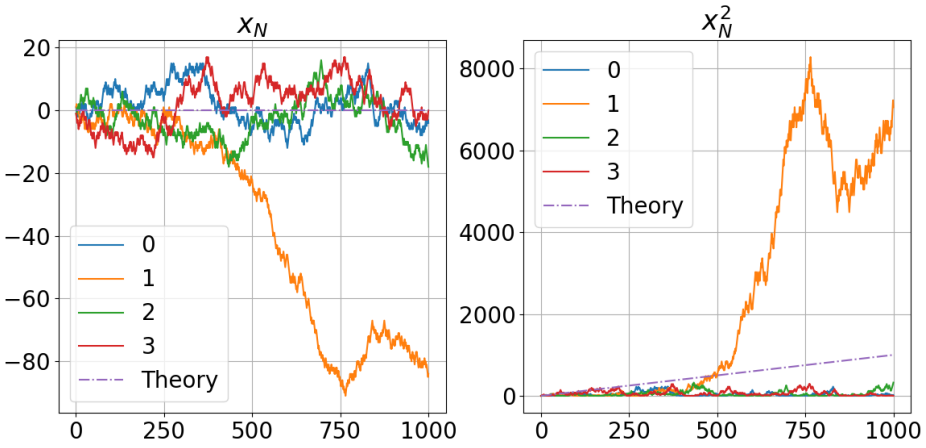

Figure 1: Instantaneous position of several bodies during a RW, and the theoretical expectations.

1.4 Accuracy (c and d)

In Figure 3 we visualize the adherence of the mean square displacement $\langle(\Delta x_N)^2\rangle$ to the theoretical expectation, where N is the last simulated time instant of the RW. The quantity which varies in the x-axis is the number of simulated walks, in order to visualize its dependency on the number of independent bodies. For

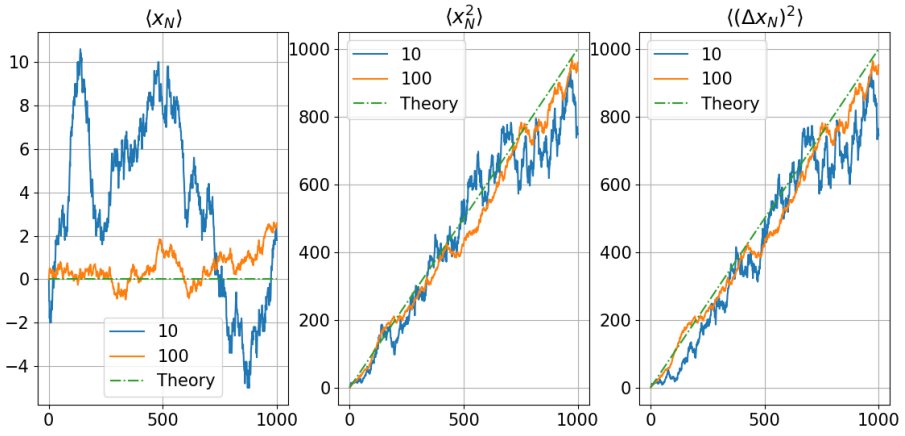

Figure 2: Experimental averages compared with the theoretical expectations for RW of 10 independent bodies (blue) and 100 (orange).

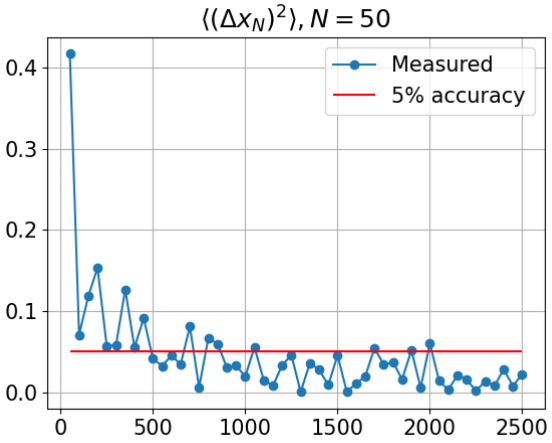

Figure 3: Relative accuracy measured for the mean square displacement.

more than 2000 RW averaged we obtain quite consistently a relative accuracy greater than 5

In Figure 4 we tried to find a correlation between N (the last simulated step) and the improved accuracy obtained by increasing the number of independent runs. At first sight it may seem that taking more steps (e.g. 128 in the plot) results in more stable and faster convergence. This is also somewhat plausible, since we're averaging more sources regulated by the same kind of randomness.

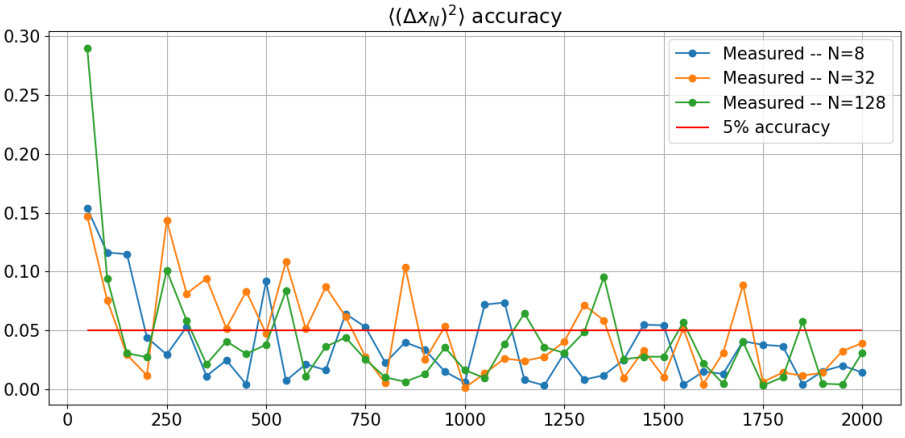

Figure 4: Relative accuracy measured for the mean square displacement, measured for different values of N.

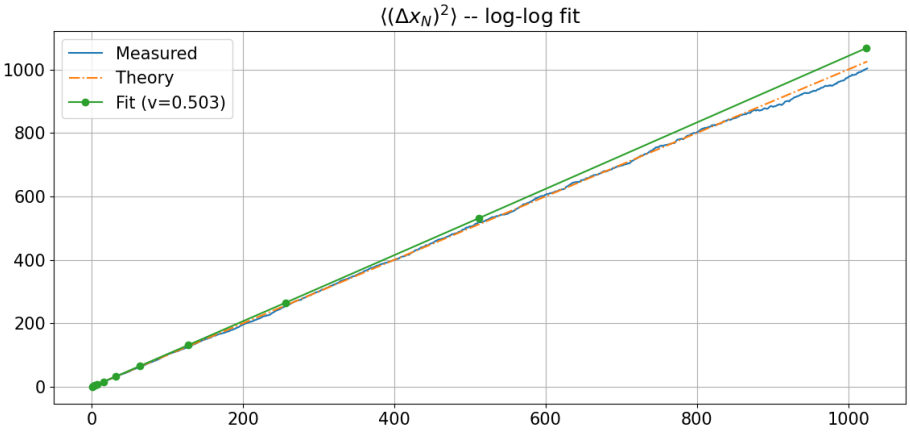

Figure 5: Comparison of the measured $\langle(\Delta x_N)^2\rangle$, the theoretical one and the fit against the expected functional dependency on N.

However the test is inconclusive, since we weren't able to identify a clear relation.

1.5 Fitting the dependency between $\langle(\Delta x_N)^2\rangle$ and N (e)

The theoretical expectation for $\langle(\Delta x_N)^2\rangle$ predicts a dependency whose functional form is aN^{2v}. By taking measurements of the simulated value of $\langle x^2\rangle - \langle x\rangle^2$ we could fit the dependency on a log-log plot, thus deducing the value of v which suits best the measurements. As we can see in Figure 5, the value is

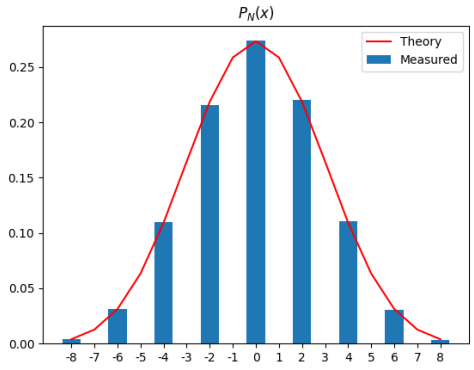

Figure 6: Comparison between the theoretical and the empirical position distribution of a 1D RW at $N = 8$.

pretty close to the theoretical expectation $v = \frac{1}{2}$.

1.6 Calculation of the distribution (f)

During the simulation we accumulate the final position of all the independent bodies taken into account with $N = 8$, and compute the empirical distirbution $P_N(x)$. We compare it against the theoretical one in Figure 6. The prediction is matched pretty accurately, expect for the fact that $P_N^{th}(x)$ is a continuous function, while its empirical counterpart as pretty evident holes coinciding with odd values of the position. This is due to the fact that for $N = 8$ it's impossible to end up in an odd spot.

For the computation of $P_N^{th}(x)$ it's more convenient to compute $\frac{N \pm x}{2}$ rather than $\frac{N}{2} \pm \frac{x}{2}$ because this way we can save one (vectorial) division.

1.7 Theoretical limit (g)

We compare the theoretical limit $P(x) = \sqrt{\frac{2}{\pi\sigma^2}} e^{-\frac{(x-\langle x \rangle)^2}{2\sigma^2}}$ to the empirical measurements of $P_N(x)$. We expect the empirical distribution to converge to the limit as N increases, since the larger N the better the estimates of $\langle x \rangle$ and $\sigma^2 = \langle (\Delta x)^2 \rangle$. This behavior can be observed in Figure 7.

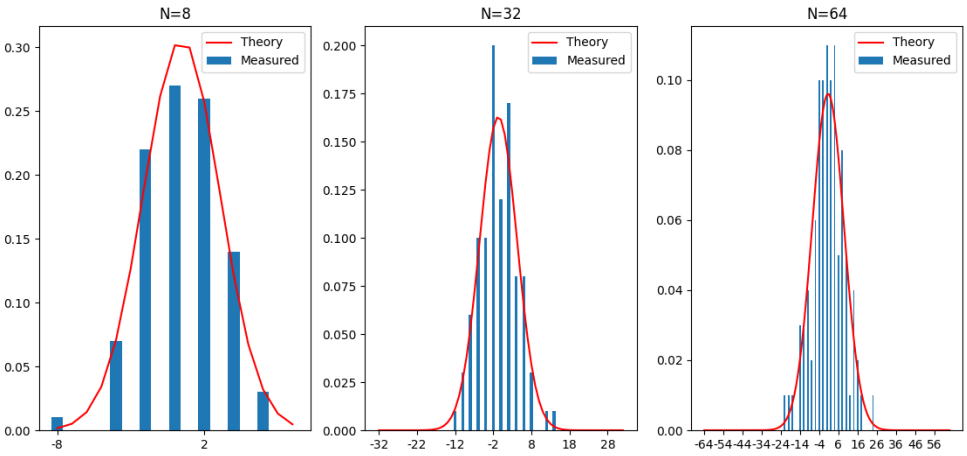

Figure 7: Comparison between the theoretical limit $P(x)$ and the empirical distribution $P_N(x)$, for several values of N.

2 Monte Carlo method — Sample mean and importance sampling

The implementation of the Monte Carlo method with importance sampling uses the inversion method to map a set of uniform random variables to an exponential RV in [0,1]. The results are plotted in Figure 8. In the plot on the left it's pretty clear the variance improvement which we get by using the IS variant of the Monte Carlo method. The plot on the left shows the empirical dependency of the two variants on the number of samples N.

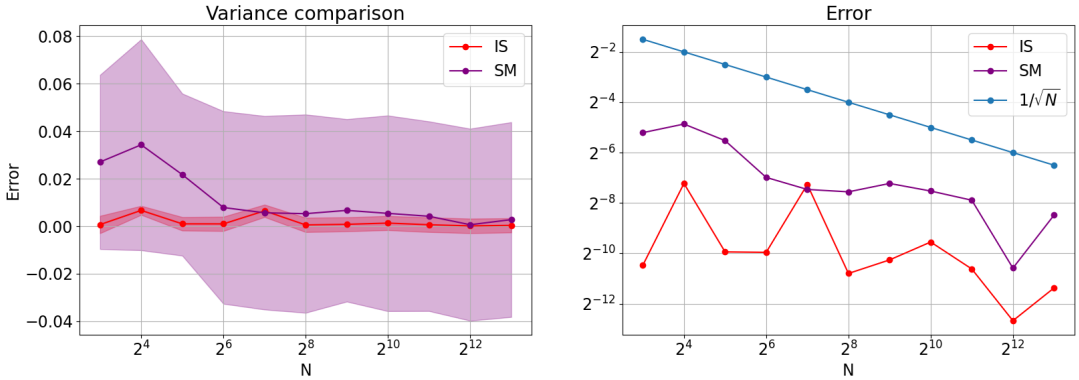

Figure 8: Comparison of the importance sampling (IS) and sample mean (SM) variants of the Monte Carlo method.

3 Monte Carlo method — Error analysis

In Figure 9 we plotted a few estimators of the error for the Monte Carlo integration method (sample mean variant) of $4\int_0^1 \sqrt{1-x^2}dx = \pi$. The expected evolution of the actual error Δ_n is clearly respected (left), since it follows quite nicely the line $1/\sqrt{N}$. It's also pretty evident how bad of an estimator δ_n is, since it's almost an order of magnitude larger than the empirical error.

In the plot on the right we depicted some better estimators of the error, namely $\sigma_s/\sqrt{s}$ (variance of block average), σ_m (variance of average of averages) and $\sigma_n/\sqrt{N}$. The three values are all pretty close, and they are all quite coherent with the actual value Δ_n.

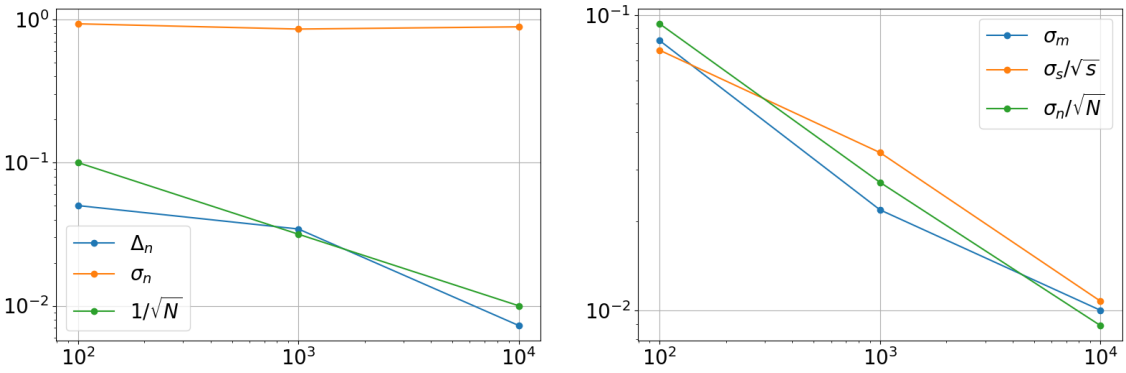

Figure 9: Comparison between several error estimators for the Monte Carlo sample mean integration method.

4 Software

All the simulations in this report were carried out using Fortran 90. Data analysis and visualization were carried out using Python 3 together with the libraries NumPy (for data collection and aggregation when necessary), Matplotlib and scikit-learn. All script files not provided with the delivered homework are available on my GitHub repository for the course *Computational Physics Lab*.