

WEEK 2: RANDOM WALKS

FRANCISCO GARCÍA FLÓREZ

1 Introduction

In this assignment random walks will be studied, both in two and three dimensions. Moreover, we will compute $\langle R^2(N) \rangle$, that is, the distance squared after each step from the origin averaged over a number of walks and study its functional dependence.

2 Program usage

Following there is a brief guide explaining how to use the program:

- **week2 -N int**: computes Pi using **int** points in a square of size $[-1, 1] \times [-1, 1]$, counting how many lie inside a circle of radius one.
- **week2 -e float**: calls the previous function with increasingly larger amounts of points until Pi is approximated with an error less than **float**.
- **week2 -walk dim steps file**: performs a random walk in **dim** dimensions with **steps** steps, and saves its results to file in json¹ format.
- **week2 -pwalk dim steps size file**: performs a random walk just in the previous case, including now periodic boundary conditions as a **dim**-dimensional box of size **size**.
- **week2 -rsquared dim steps nwalks file**: performs **nwalks** random walks computing $\langle R^2(N) \rangle$ and saving it to **file**, also in json format.
- **week2 -prsquared dim steps nwalks size file**: similar to the previous function, incorporating now periodic boundary conditions.

By running the script **week2.sh** these commands will be executed, showing their results. By executing **python python/graphs.py** a number of plots can be generated.

3 Random walks

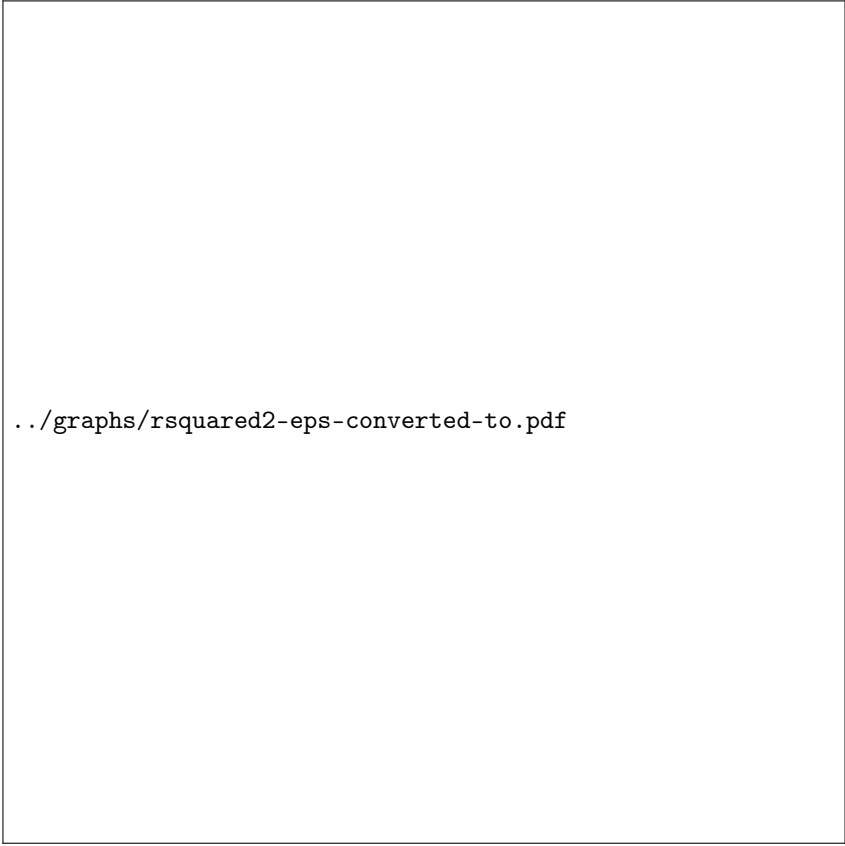
3.1 Functional dependence of $\langle R^2(N) \rangle$

Firstly, we compute the required data by executing the following commands:

- **./bin/week2 -rsquared 2 200 10000 data/rsquared2.json**
- **./bin/week2 -prsquared 2 200 10000 20.0 data/prsquared2.json**
- **./bin/week2 -rsquared 3 200 10000 data/rsquared3.json**
- **./bin/week2 -prsquared 3 200 10000 20.0 data/prsquared3.json**

And then, after plotting their results we get:

¹See <http://www.w3schools.com/json/> for more information.



../graphs/rsquared2-eps-converted-to.pdf

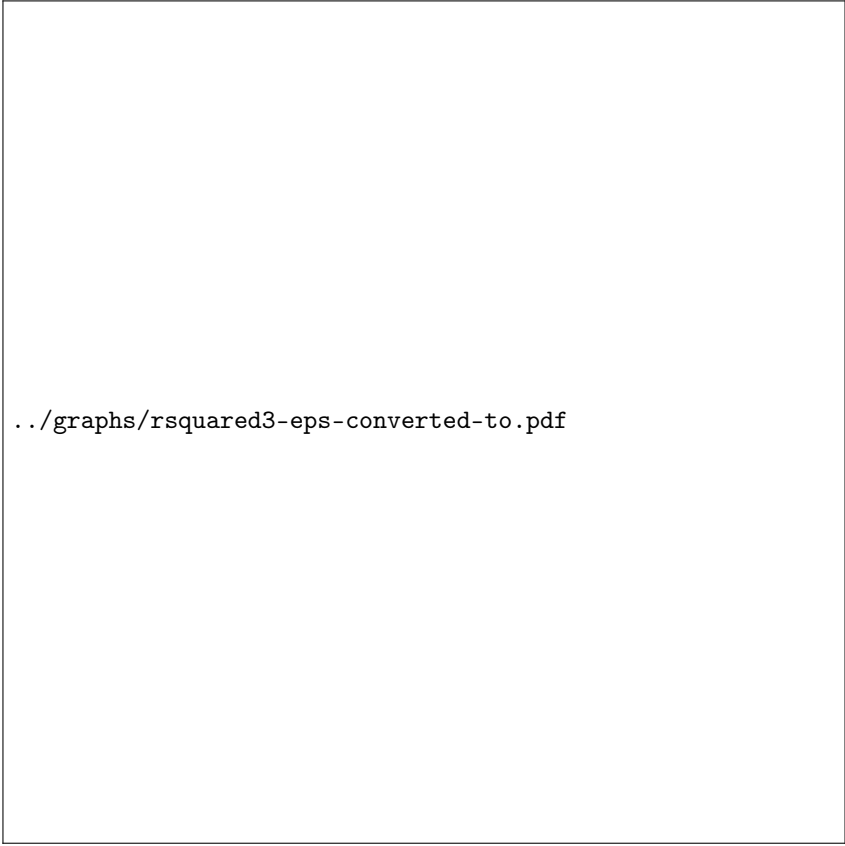
Figure 1: Graphical representation of $\langle R^2(N) \rangle$ in 2D.

It is easy to spot the linear dependence of $\langle R^2(N) \rangle$ when there are no periodic boundary conditions (“PBC” in the following), with a slope of 1.00 ± 0.01 and r-value larger than 0.9999. In the second case, with PBC, fitting the function doesn’t look that easy. At least, we can see that instead of increasing uncontrollably, it is bounded by some limit value.

If we fit the data to $f(x) = a(1 - e^{-bx})$, even if it’s not perfect, allows us to estimate $a \sim 70$ in this case. This behavior is expected, however, since the particle cannot escape the box. It is also expected that up to some number of steps N_* both averages should take almost the same value, namely, when the probability of reaching the boundary is not negligible.

A better fitting function may be a hyperbole, $ax^2 - by^2 = c$, rotated so the axis match with the unbounded line and the limit, blue and magenta lines respectively.

Now, repeating this procedure for three dimensions we get the following results:



../graphs/rsquared3-eps-converted-to.pdf

Figure 2: Graphical representation of $\langle R^2(N) \rangle$ in 3D.

So, the linear dependence of $\langle R^2(N) \rangle$ in the unbounded case has not changed, however considering PBC the limit of $N \rightarrow \infty$ changes to $a \sim 110$. This increase makes sense after all, since in 3D the particle has more “freedom” to move around before getting to the borders, and therefore this average will be closer to the unbounded case. It is also possible to interpret this result by considering that the surface of a sphere in 3D is larger (4π against 2π) than its two dimensional counterpart, so the probability of concatenating several steps in the same direction is lower.

As an experiment, the following results were generated from random walks in 1D and 11D, so this dimension dependent behavior is more explicit:



Figure 3: Graphical representation of $\langle R^2(N) \rangle$ in 1D and 11D.

As expected, in 1D a is lower than in the previous cases, at it reaches the limit quicker. Although, we can easily see some unexpected behavior: “steps” in the function when imposing PBC. This may be explained by considering that in 1D one step can only be taken in two different directions, with 50% chance for each, so it may produce this effect. In 11D on the other hand, everything works as expected, with a larger value for a and smaller for b .

3.2 Generating random unit vectors

If we tried to generate uniformly distributed points on the surface of a sphere by choosing two random values for the angles, we wouldn’t find a uniform distribution on the surface. It’s easy to visualize by mapping uniformly chosen points from $[0, \pi] \times [0, 2\pi]$ to the sphere, as most of them will be located near the poles.

Instead of using a uniform probability distribution for generating random numbers in angle-space, it would be possible to choose a different one that compensates this spherical deformation. However, not easily done.

Since we cannot check uniformity by only looking at plots like the following, we have to follow a more indirect procedure.



Figure 4: Graphical representation of a random walk in (left) 2D and (right) 3D projected to a plane.

For example, considering only random walks of N_S steps, we can make a density plot of how many times each point has been visited. When there are no bounds, we expect that for N_S large enough, this density distribution will have rotational invariance. In the following plot the results are represented:



Figure 5: Density distribution of visited points in 2D. (left) Without boundary conditions, (right) with periodic boundary conditions.

It's clear, even more so in the right plot, that our vectors are indeed uniformly distributed in the sphere. In this case we have performed the random walk in 2D, but the algorithm for generating the next step is the same independently of dimensions.

On the other hand, we can also analyze a similar density distribution generated this time by only one walk after a large

number of steps:

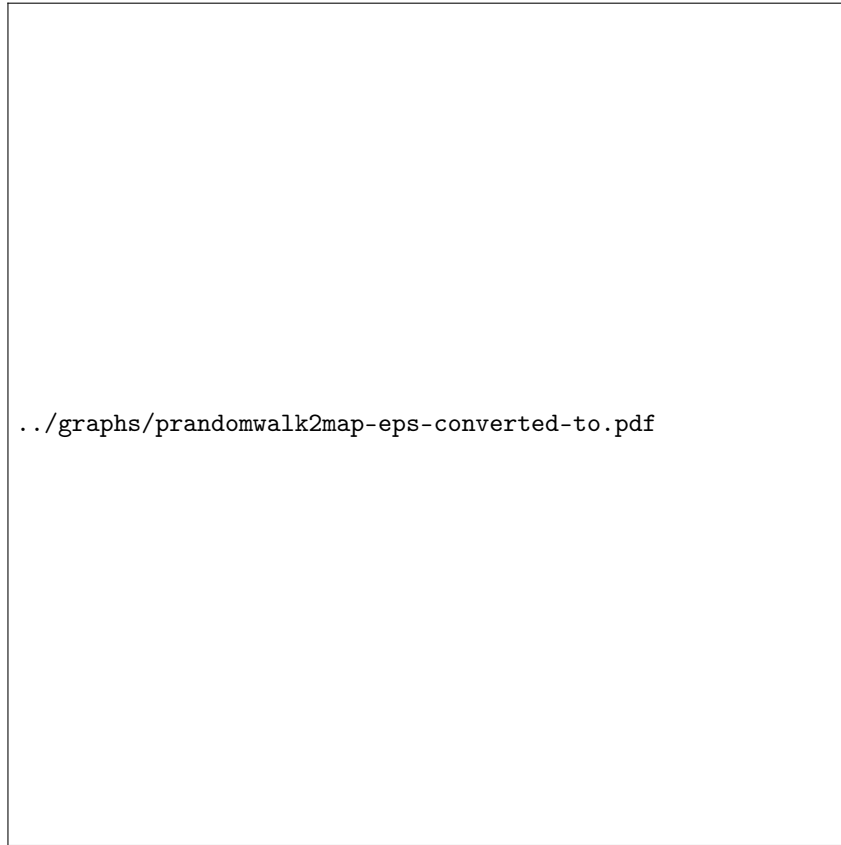
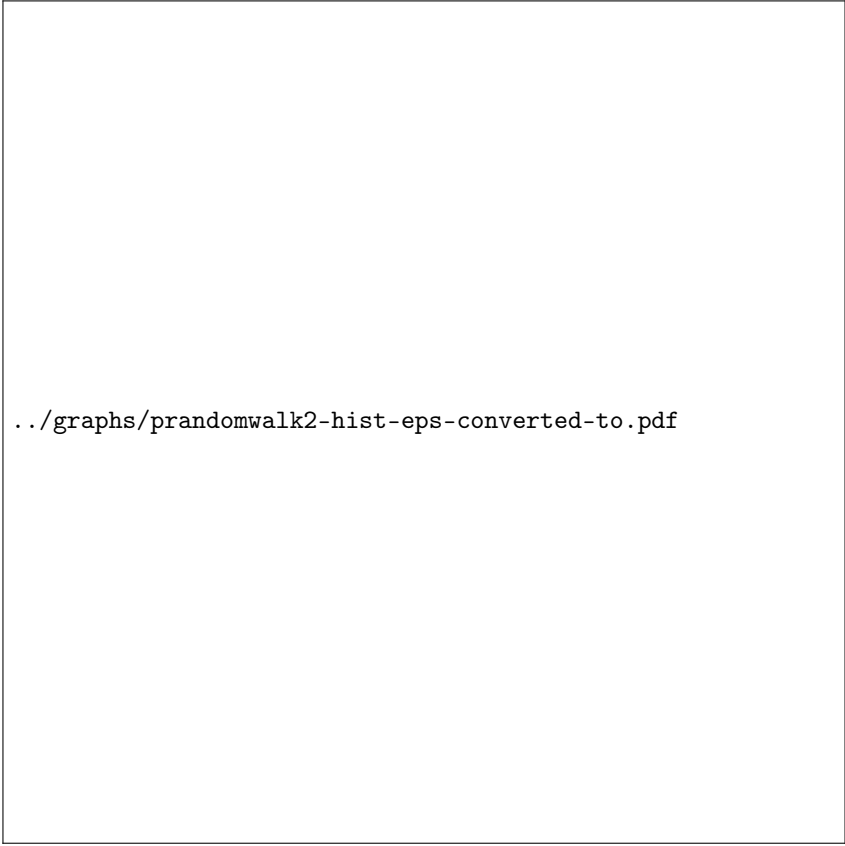


Figure 6: Density distribution of points visited by one random walk with periodic boundary conditions.

As expected this distribution is uniform, with small fluctuations around the average. Although, an histogram of the probability density around this average shows something a bit unexpected:



`../graphs/prandomwalk2-hist-eps-converted-to.pdf`

Figure 7: Distribution of values around the average.

Of course, those fluctuations are Gaussian around the average, but it seems like there are two different sources, generating two clearly different distributions. This behavior is not dependent on the box size, and as of now the possible reasons behind it are not clear.