Sebastian Straszak, s1728659

Modelling and Visualization[1]: Exam 2021-22

Contents

[1] Thanks for the course! Definitely the funnest one this year (putting aside the MSc which was equally fun!)

1 1

1.1 Numba Preface

Preface regarding the use of Numba in code

I am using Numba for most of the simulation code. This makes for loops/etc faster than default numpy functionality in some cases (i.e. rolls and other such functions.) I.e. more efficient.

1.2 Finite Difference Scheme

Going to be using Euler-Forward for temporal and Centred-Spatial for the laplacians.

$$a_{i,j}^{n+1} = \left[\frac{D}{\delta x^2} \left(a_{i+1,j}^n + a_{i-1,j}^n + a_{i,j+1}^n + a_{i,j-1}^n - 4a_{i,j}^n \right) + q a_{i,j}^n \left(1 - a - b - c\right)_{i,j}^n - p(ac)_{i,j}^n \right] \delta t + a_j^n \tag{1}$$

and so forth for the differencing (same style for the partials for b and c, just change the quantities around appropriately.)

I have defined the new matrix d via

$$d = 1 - (a + b + c) \tag{2}$$

for convenience.

1.3 Run Readme

To run this code, you will have to manually specify the values required (I've included documentation for the class necessary.) Simply open the file

```
question_1.py
```

and alter the necessary lines for the run that you wish to do. Once you've done this, simply do

```
python3 question_1.py
```

to run the code with the configuration you've set.

2 2

See Figure 1 for the oscillatory nature of the fractions over time (for $T \in [0, 200]$) with $\delta t = 0.01$. I've included a screenshot for the field defined by my implementation of Section 1.2 at $T = 50$.

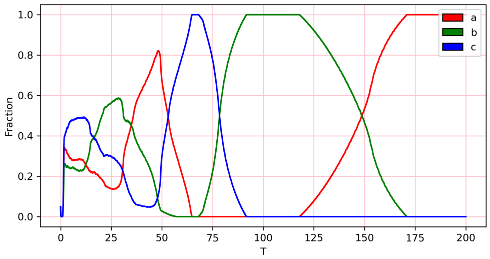

Figure 1: Fractions for species a,b,c as a function of time. The behaviour is clearly oscillatory (with a fairly defined periodicity) whereby the system explores alternates between states dominated by either of the chemical species fractions, in this case cycling from R-G-B (a-b-c.) It's interesting to see that despite seemingly reaching absorption occasionally, other fractions can still bounce back and continue the cycle (i.e. after b reaches unity and suddenly red spurs into action later.)

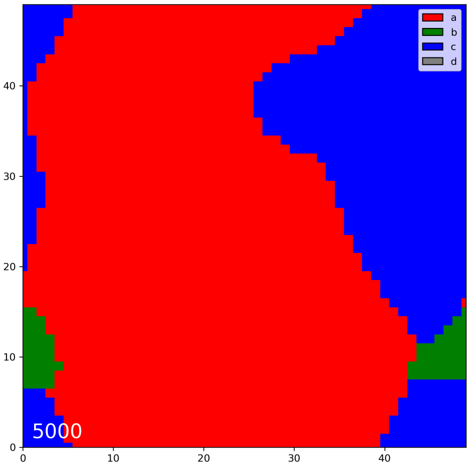

Figure 2: Example type field $\tau_{i,j}^{n}$, here for $n = 5000$, i.e. the 5000th sweep corresponding to $T = 50$ for $\delta t = 0.01$.

3 3

I've done this for 100 runs (gives cleaner looking histogram/graph.) See Figure 3. The functional form of ρ, the PDF for the time required to reach absorption, is certainly asymmetric- no doubt if you allowed T run to infinity as the limit (instead of setting a hard limit for T of 1,000 as described in the question sheet) the average would be skewed by the outliers, I think.

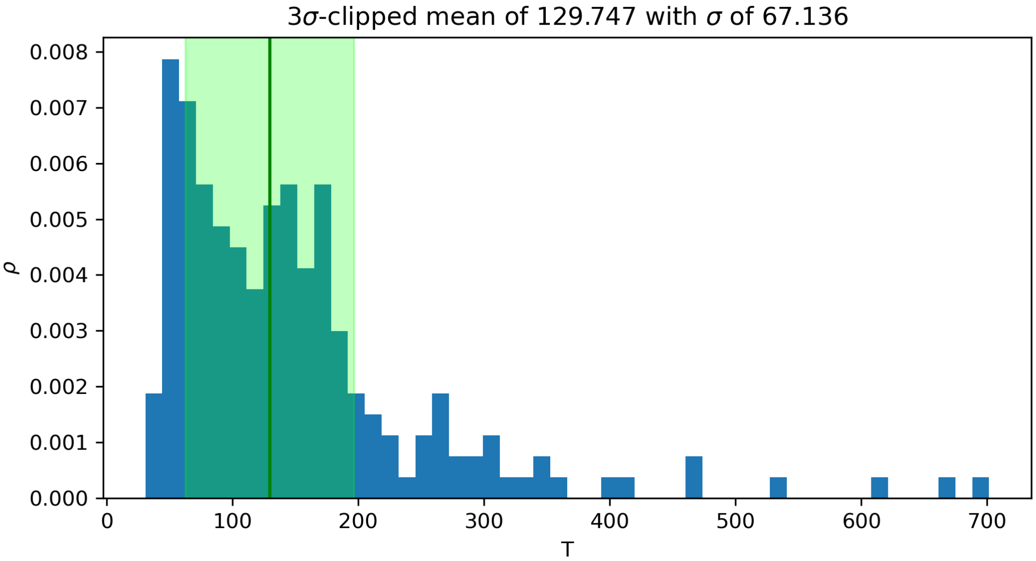

Figure 3: Histogram for the time required for absorption as requested for part (c). I've highlighted the sigma-clipped mean (clipped by 3σ) with a one-sigma radius region coloured in. Note that this is the *sigma-clipped mean* and not the regular mean. Here the probability density is illustrated by the histogram.

4 4

See Figure 5 for snapshot and explanation. See also Figure 4 to see oscillatory fraction-over-time.[2]

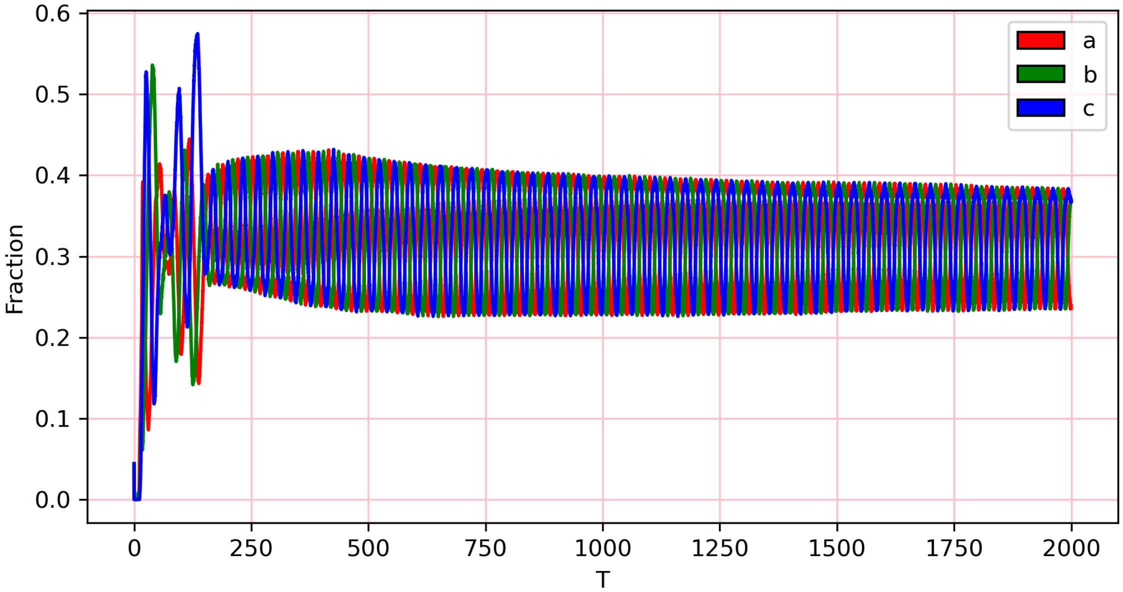

Figure 4: Same plot as for 2 but for this case. Fairly obvious that the fractions oscillate over time, with a fairly well-defined periodicity, about some equilibrium fraction with a fairly consistent amplitude. This is clearly elliptical stability. Given the periodic nature of the simulation (as mentioned in Figure 5, the swirling of the $\tau \in d$ points) one would expect the fractions to oscillate consistently for any point in the grid (at least those not $\tau \in d$, which I have no clue about.)

[2] HOLY HELL THATS COOL TO WATCH!

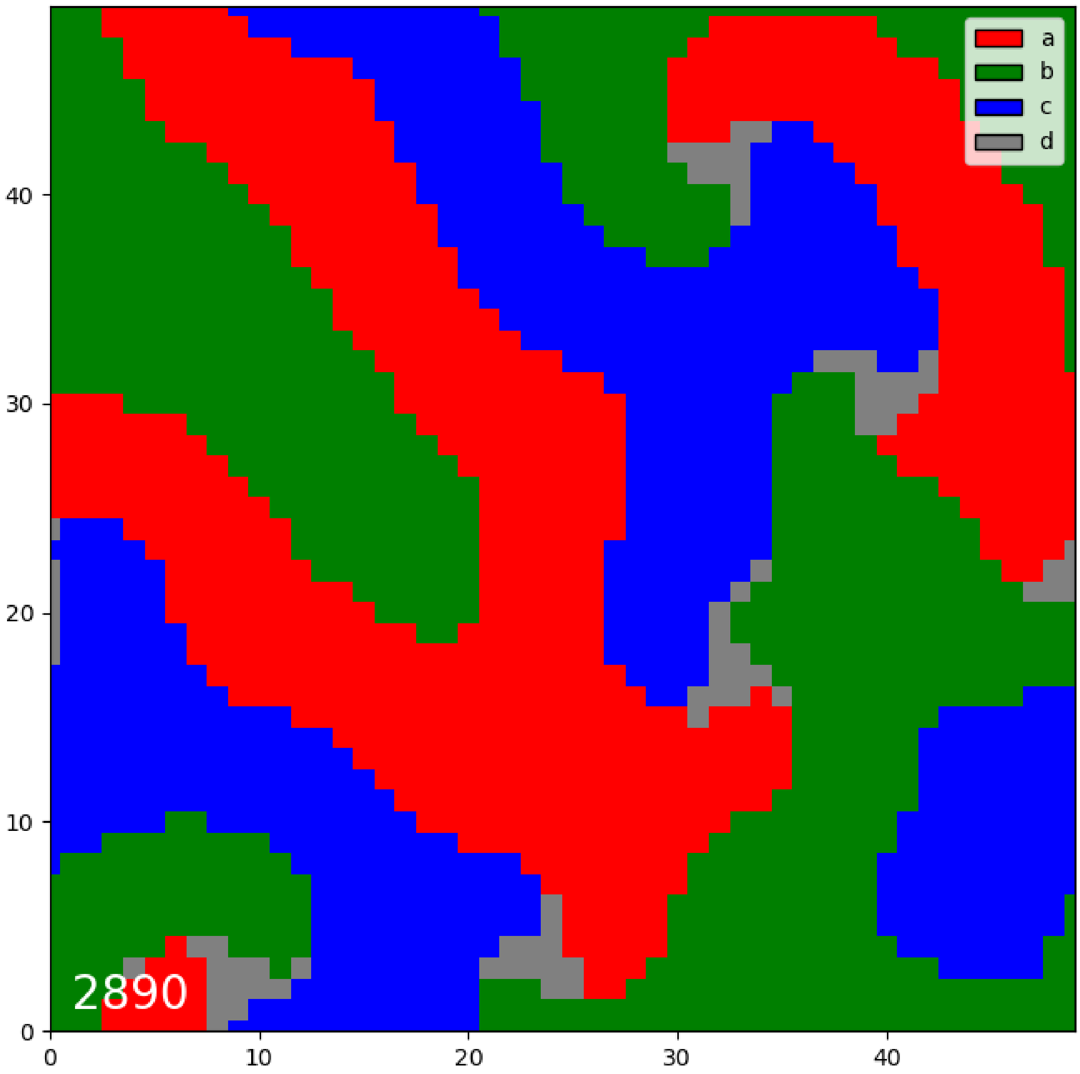

Figure 5: Example snapshot (with sweep illustrated) for $D = 0.5, q = 1, p = 2.5$. The gray regions (i.e. where $\tau \in d$) remain stable over time and appear to rotate with constant period, like paddles swirling a bucket of three non-mixing dyes. These gray points have three facets, with each chemical species (in this case, red-green-blue a-b-c) effectively clinging on to them, being dragged spirally as these points rotate over time. Similar behaviour to the vortices seen in the XY model (see here.) *I haven't checked to see if the total amount of a/b/c remains constant mind you- I was just making an analogy.*

5 5

See Figure 6 for my plot demonstrating oscillation over time. See the file

```
    question_5.py
```

for my scipy.curve_fit implementation on fitting generic sines to a subset of the data. The periods come out roughly near unity for both points (I chose to use points [10,10] and [20,20]). The code used to generate the periods is between lines 45 and 53, given by

```
def sine_function(t, a, b, c, d):
return a*np.sin(b*t + c) + d

fitpars = []
for type in types:
    popt = curve_fit(sine_function, times[int(max_sweeps/5):], type)[0]
    fitpars.append(popt)
fitpars = np.array(fitpars).T[1] # gives the frequencies
print(fitpars)
```

which returns, for an example run,

```
    [0.996291   0.99863988]
```

in the terminal. You can just run the code (question_5.py) if you want to verify this.

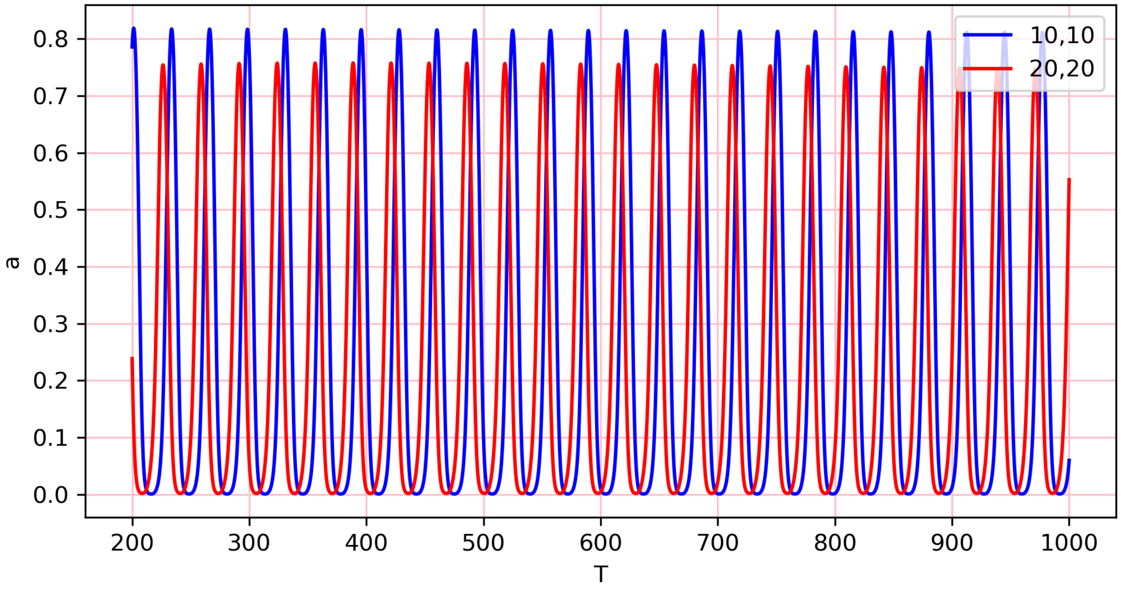

Figure 6: Value of *a* over time for two grid points, shown in the legend. They clearly oscillate over time, both with the same frequency.

6 6

I only have a few minutes left so this has been done very dirtily, you'll have to forgive that. The correlation function ρ has not been normalised: the rough idea to how I calculated it was to

```
For each point in lattice
Check distances to all other points
Select points within the r threshold chosen (euclidean distance) up to nx/2 = 25
Get the fraction of points that are the same as the point being considered
This is the probability
Do this for all points in lattice
Average of this is the two-point correlation function (averaged over all points in lattice)
Do this in 2D
Distance takes into account periodic boundary conditions
```

I really am short so I can't explain it better... sorry! (My code is there though!)

See Figures 7, 8, 9 for the plots for $D = 0.3, 0.4, 0.5$ respectively. Generally speaking, the smaller the value of D (for this particular set of parameters, which admits oscillatory behaviour) the faster the correlation function decays into the background (average correlation, i.e. $\rho \approx \frac{1}{3}$.) These aren't time averages (they were taken at $T = 200$) but I'm assuming that the correlation function remains steady over time, at least somewhat.

My belief here is that given that D is a diffusion coefficient, the lower D is, the less likely the chemical species are to diffuse from their centres over time (even oscillator-ily they'd still diffuse over the period of their swirling motions as discussed earlier.) That gives the lower correlation. I am out of time, sorry!!!!!!!

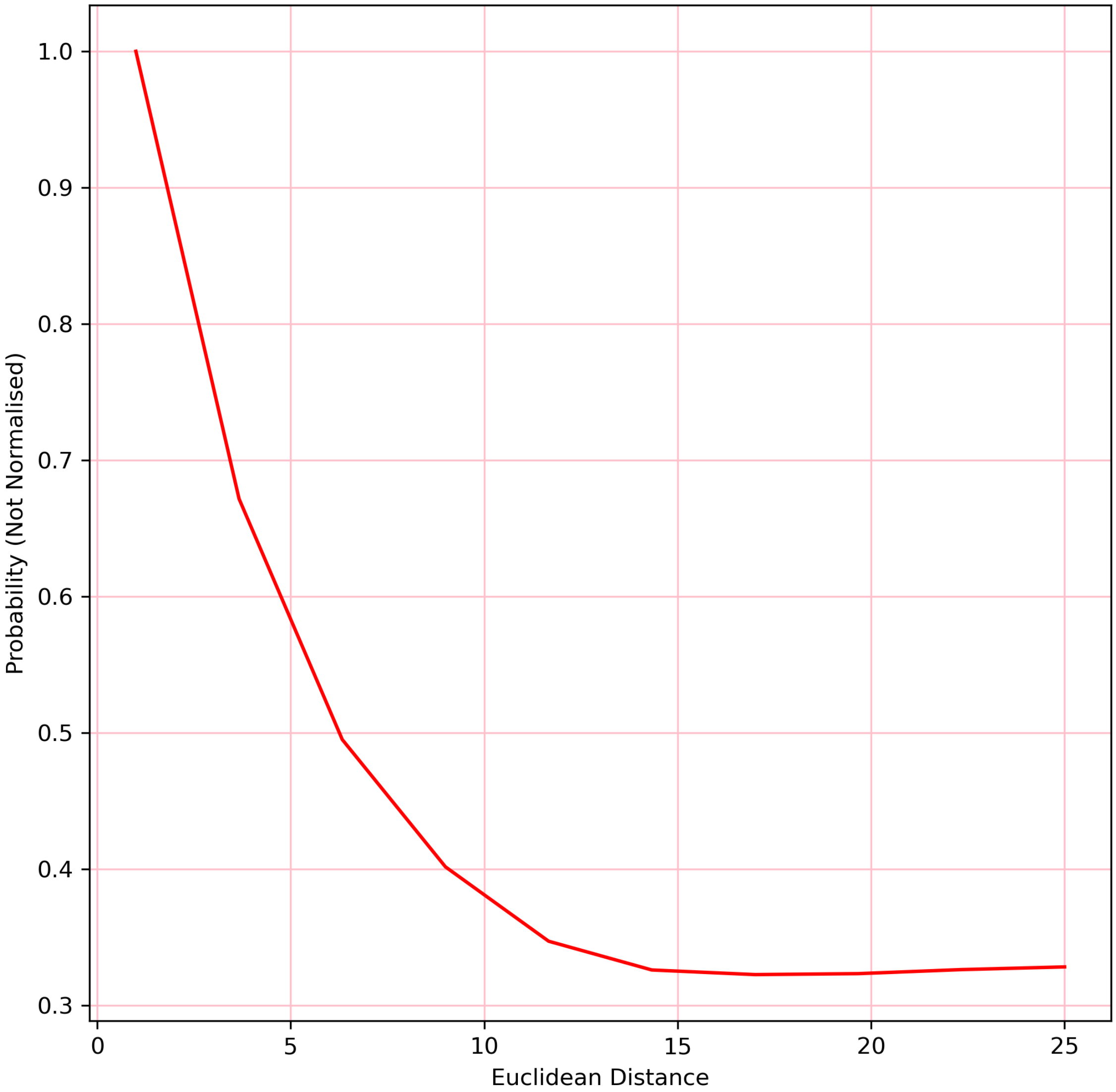

Figure 7: Correlation function as function of radius for D=0.3

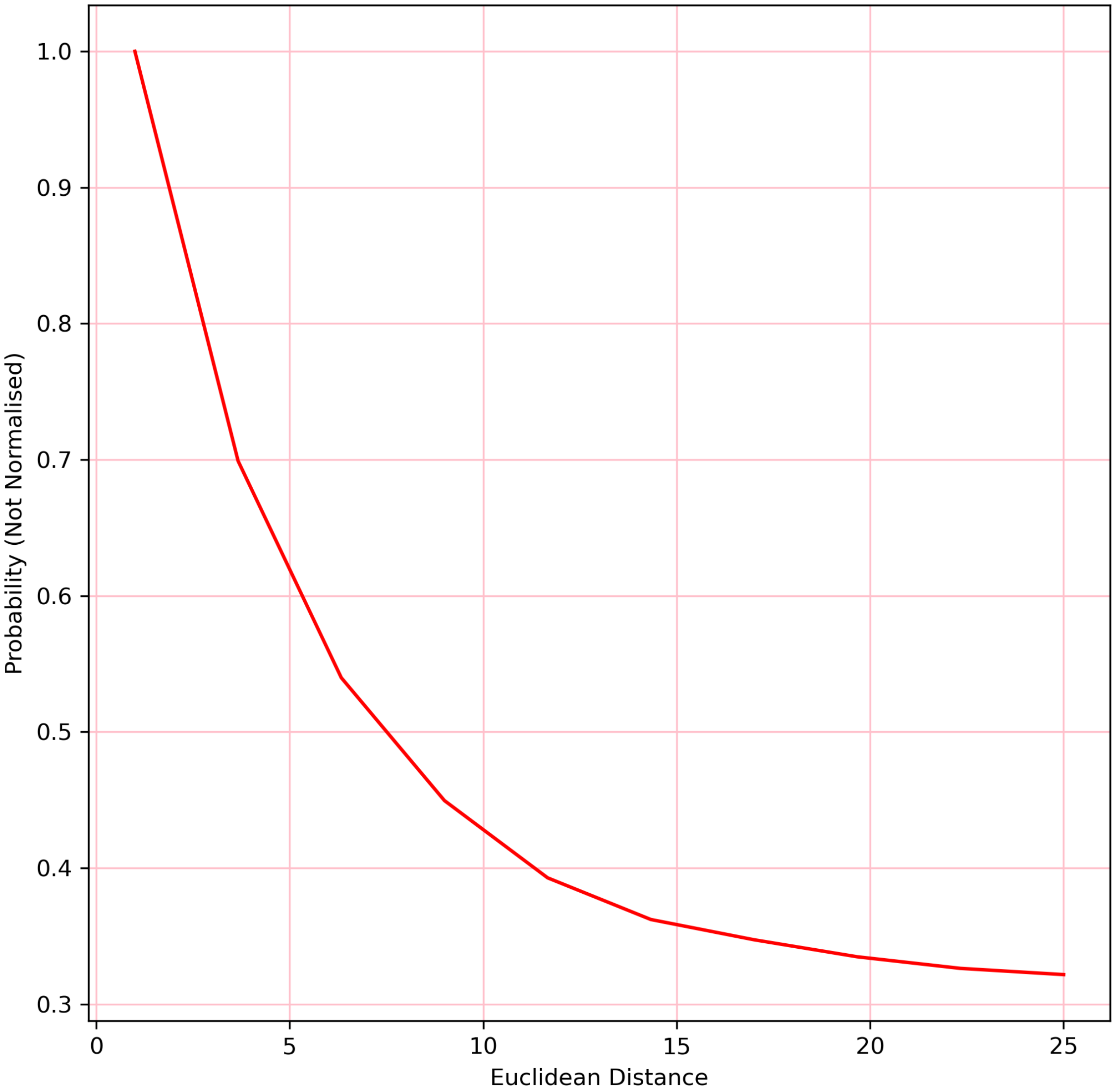

Figure 8: Correlation function as function of radius for D=0.4

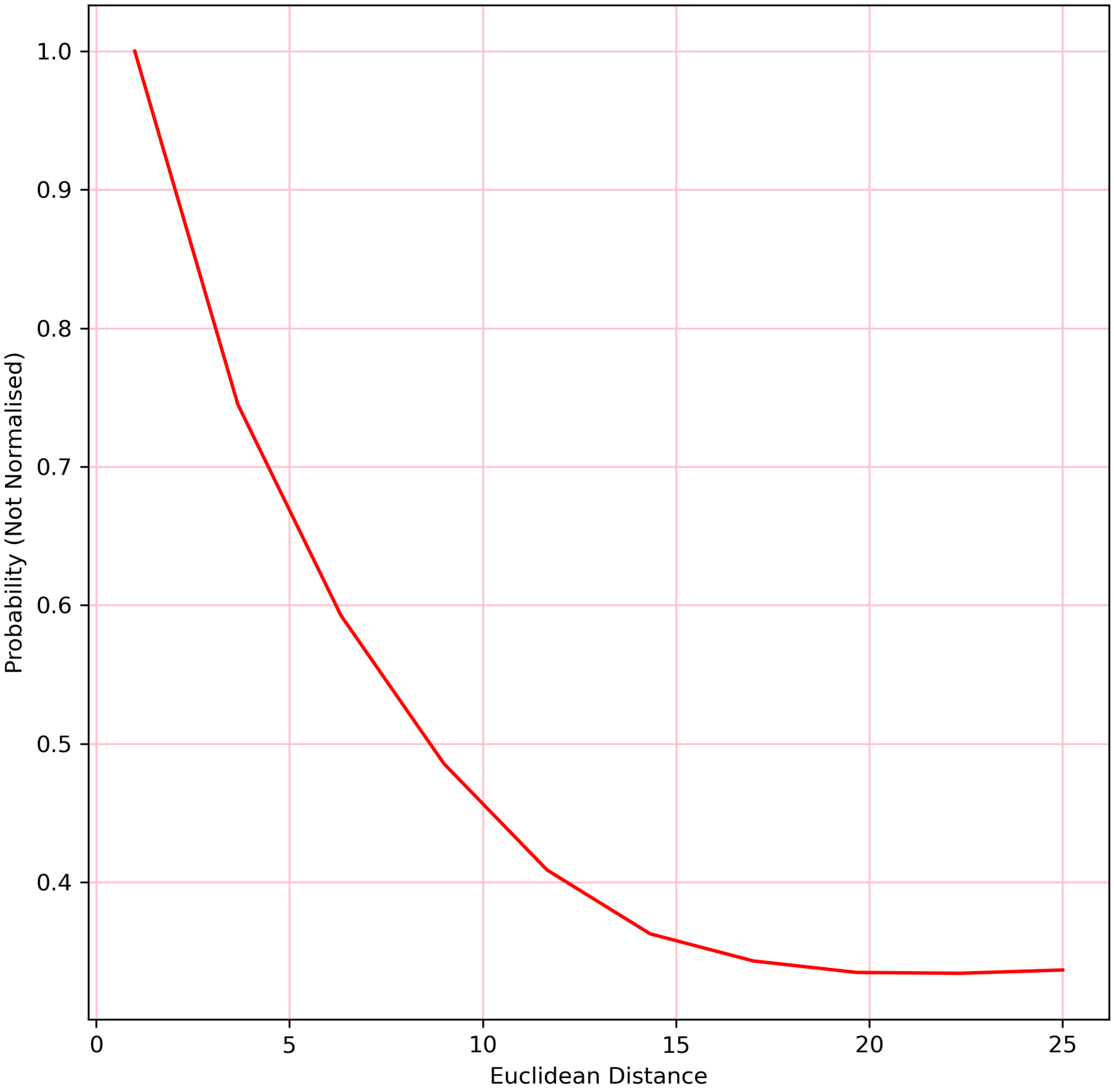

Figure 9: Correlation function as function of radius for D=0.5