Population Dynamics: Chaos

9. Sensitivity to Initial Condition

Run your program of the last class (8.) for $a = 3.7$. Compare the results $N(t)$ for the initial conditions $N_0 = 0.5$ and $N_0 = 0.500001$ by plotting their curves in the same graph.

10. Bifurcation Diagram

Next you will investigate the dependence of your results on $a$. To do so change your program such that it runs the same simulation for $a = 2.5, 2.51, 2.52, 2.53, \ldots, 3.98, 3.99, 4.0$. For each $a$ value run the program for 500 time steps but print for only the last 100 timesteps $a$ and $N(t)$. Pipe your result into xgraph. You therefore obtain a graph $N(t_{eq})$ as a function of $a$. Explain what you find.

Random Walks

1. Random Walk in One Dimension

Program the random walk in one dimension. Assume that a step to the left and a step to the right are equally likely and that the steps are of size 3. Use an integer for the position variable $x$. Start at $x=0$ and use $a=3$. Plot $x$ as a function of the time steps $n$.

Hint: Use a bit-random number generator, for which you can copy the sample file

`kvollmay/classes.dir/capstone_s2001.dir/fractal.dir/bit_rand.cc`

This sample file produces random numbers $r$ which are 0 and 1. An update is $x_{n+1} = x_n + (2 \times r - 1) \times a$.

2. Random Walk in Two Dimensions

2a. Write a program for the random walk in two dimensions. You will need the random number generator for numbers $0 < r < 1$. Use

`kvollmay/classes.dir/capstone_s2001.dir/fractal.dir/float_rand0-1.cc`

Use 10000 timesteps and start with $x = y = 0$. Plot $x$ and $y$ as a function of the step $n$. 
Random Walks

2a. The random walk in two dimensions consists of steps up, down, right or left with equal probability. Write a program of the random walk in two dimensions and plot $x(n)$ and $y(n)$.

Hint: For the random step you may want to draw a random number $0 \leq r < 1$ and go up if $r \leq 0.25$, down if $0.25 < r \leq 0.5$, etc..

2b. Next let’s make a movie of your random walk. Define a lattice of size 100x100 and initialize it with all sites equal to zero. Set $a = 1$. For each timestep print the lattice such that you can pipe your output into DynamicLattice. Set the current lattice site $(x, y)$ equal to 2 and all sites which were visited in the past equal to 1. You should obtain a white polymer with a red tail and blue background. Run your program for about 300 timesteps.

Fractals

3. Determine the fractal dimension of the pattern on the transparency (second pattern.) Use the same method as for the first pattern.

4a. Determine the fractal dimension of the quadratic Koch curve.

4b. You find a program which produces the quadratic Koch curve in ~ kvollmay/classes.dir/capstone_s2001.dir/fractal.dir/quad_koch.cc

Copy this program, have a close look at it and explain what exactly it does.
Fractals

3. Fractal Dimension
Determine the fractal dimension of the pattern on the transparency (second pattern.) Use the same method as for the first pattern.

4. Quadratic Koch curve
4a. Determine the fractal dimension of the quadratic Koch curve.

4b. You find a program which produces the quadratic Koch curve in
~ kvollmay/classes.dir/capstone_s2001.dir/fractal.dir/quad_koch.cc
Copy this program into your working directory. Compile the program and run it with:
executable | xgraph -m
The program is an example for “recursive programming,” which is tricky to understand. Explain how the program works: (i) Draw a sketch for the case of $n = 1$. The program prints out points. Label the corners of your sketched curve and assign the labels to the points $(x_1, y_1), (x_2, y_2)$ etc.. (ii) Draw a sketch for the case of $n = 2$ and explain the order in which the commands on the lines 47-51 and 54 are executed.
DLA Model

5. Flow Chart
Get together in groups of two and draw a detailed flow chart of the DLA model.

6. Initialization
Since the program for the DLA model will be a little bit longer, it will be useful to define functions for the different tasks. For testing purposes use a 10x10 lattice (but define the length of the lattice sides as a global constant so that we can change it later easily) and set up the initial lattice with all sites empty (0) and the middle site with the seed particle (1). Write a function which prints the lattice and call it in your main function after the initialization.

7. Starting Point For Random Walker
Write a function that takes as input the radius $r$ and passes back the x- and y-values of a lattice site near a randomly picked point on the circle around the middle of your lattice and with radius $r$. Comment out the printing command of 6. and print instead 100 x- and y-values near the circle by calling your function 100 times. Use $r = 3$ and look at your resulting data points.

**Hint:** Use $x = r \cos \theta$ and $y = r \cos \theta$, where $\theta$ is a randomly picked angle $0 \leq \theta < 2\pi$. Define the constant $\pi$ as 2.0 * asin(1.).

8. Random Walk
Write a function which has as input x and y values, does a single random walk step and changes the x and y values accordingly. Now let the particle on the circle (of radius 3) do 5 random steps and check your program by using DynamicLattice.
**DLA Model**

Today we will continue programming the DLA model. Use your program of the last class and daily assignment (exercises 6. & 7.) and build your program following the steps below.  

8. **Random Walk**

Next we add the random walk to the program.

(a) Write a function which has as input x and y values, does a single random walk step and changes the x and y values accordingly. Now let the particle start on the circle (Initialize $R_{\text{clust}} = 3$, and use $R = R_{\text{clust}} + 2$ for the radius of the circle.) Use a 25x25 lattice, do 10 random steps and check your program by using DynamicLattice. Draw the seed and the random walking particle with red (1) and the rest with blue (0).  

(b) Replace your loop of 10 random steps such that the random walk continues until the random walk particle has from the center a distance larger than $2 \times R_{\text{clust}}$. To test your program use a 50x50 lattice, $R_{\text{clust}} = 6.0$ and DynamicLattice.

(c) Add to your loop of the random walk that a particle which hits the cluster of particles sticks to the cluster and the random walk stops. Test your program again with DynamicLattice and the parameters in (8b) but start instead of with a single particle with a 15x15 square of particles in the middle of the lattice (i.e. borders $L/2 \pm 7$).

9. **Complete DLA Model**

Now you are ready for the final additions to your program and you will have a DLA model program!

Add a loop over particles and determine $R_{\text{clust}}$, which is the maximal distance of the stuck particles from the center. Keep in mind that the program stops either when you reached the intended number of particles or when your cluster outgrows the lattice. Start with a single particle as seed, with $R_{\text{clust}} = 3.0$ and use a 50x50 lattice and draw maximal 50 particles. Test your program with DynamicLattice for which you may want to take out the pause command of (8a).

10.

---

1 You find a solution of 6. & 7. in \`kvollmay/classes,dir/capstone,s2001,dir/fractal,dir/DLA_ex6,cc\.

2 Reminder: In DynamicLattice you can slow down the movie with \texttt{#pause 1}.  

DLA Model

9. Complete DLA Model

Today you are ready for the final additions to your program and you will have a DLA model program! Use your program of the last class and daily assignment (exercises 8a-c.)\(^3\) Add a loop over particles and determine \(R_{\text{clust}}\), which is the maximal distance of the stuck particles from the center. Keep in mind that the program stops either when it reaches a maximum number of particles \(N_{\text{max}}\) or when your cluster outgrows the lattice, i.e. when \(2 \times R_{\text{clust}} \geq \left( \frac{L}{2} - 2 \right) \) where \(L\) is the length of your lattice. Start with a single particle as seed, with \(R_{\text{clust}} = 3.0\), \(N_{\text{max}} = 50\) and use a 50x50 lattice. Test your program with DynamicLattice as in 8a but now you may want to take out the pause command.

10. Picture of DLA Cluster

(a) Now let’s have fun with with your program. To look at a larger cluster set \(L = 200\) and \(N_{\text{max}} = 1000\). This lattice is too large to make a movie of the growth but instead print out only the last configuration. Look at it with DynamicLattice.

(b) To be able to increase the number of particles we need to increase the lattice size \(L\). Use \(L = 500\) and \(N_{\text{max}} = 3000\). Since DynamicLattice takes already a fairly long time to draw the 200x200 lattice, print not the complete lattice but instead only a square in the middle of size 201x201, i.e. \(\frac{L}{2} - 100 \leq x, y \leq \frac{L}{2} + 100\).

11. History of DLA Particles

To understand the shape of the DLA clusters, let us now indicate the history of the particles with their color: Print “5” for the first 500 particles, “6” for the next 500 particles, “7” for the following 500 particles etc.. Look at the final result with DynamicLattice.

**Hint:** Adjust your check if the random walker is next to a particle of the cluster.

\(^3\)You find solutions of 8a-c in `kvollmay/classes/dir/capstone_2001/dir/fractal/dir/DLA_2x8a.cc`
`kvollmay/classes/dir/capstone_2001/dir/fractal/dir/DLA_2x8b.cc` and
`kvollmay/classes/dir/capstone_2001/dir/fractal/dir/DLA_2x8c.cc`. 
DLA Model and Variations of It

Today you will use your own model of today’s Wednesday project and you will do some analysis with it. We will share our results in class.

11. History of DLA Particles

Run your model with about 3000 particles. To understand the shape of the resulting cluster of your model, let us indicate the history of the particles with their color: Print “5” for the first 500 particles, “6” for the next 500 particles, “7” for the following 500 particles etc.. Look at the final result with DynamicLattice.

**Hint:** Adjust your check if the random walker is next to a particle of the cluster.

12. Fractal Dimension

Next we will determine the fractal dimension of your clusters. Since $\ln N = d_f \ln R$, we will make a plot of $\ln N(\ln R)$ and the slope of a linear fit to your data will be $d_f$.

**(a)** To determine $N$ as a function of $R$ define in your program an array `radiifrequency[Ld2]` which keeps track of the frequency of each radius. During the growth of your cluster update this array whenever a particle sticks to the cluster. We keep track of the rounded radii, so for an actual distance `rtemp` of a particle of the center you would update the array with `radiifrequency[floor(rtemp + 0.5)]++;`. At the end of the run you know how often each rounded radius occured. Now you are ready to calculate $N$, namely you sum the frequencies of each radius $r \leq R$ up, so

$$N = \sum_{r=1}^{r=R} \text{radiifrequency}[r].$$

Print out $\ln R$ and $\ln N$ and plot $\ln N(\ln R)$ with gnuplot.

**(b)** Next we make a linear fit to the data. Within gnuplot type

$$f(x)=a + b*x$$

then do the fit by typing

`fit [1.5:4] f(x) ’outputfile’ via a,b` where `outputfile` is the filename with your data. Have a look at your fit with `plot f(x),’outputfile’`. 