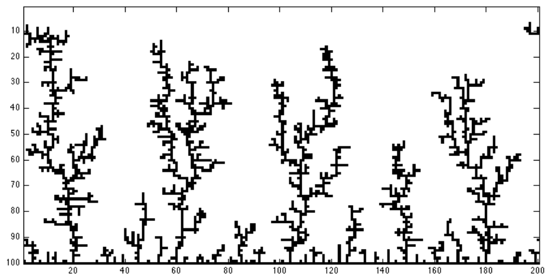
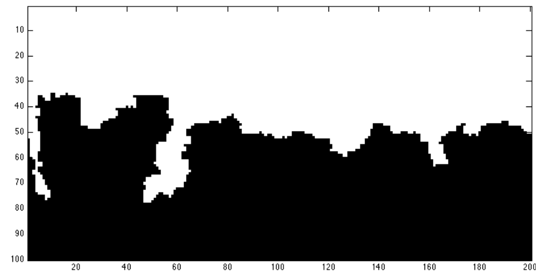
Biomedical Engineering

Diffusion Limited Aggregation

In this lab, you will edit a code that simulates the 2D diffusion of particles and their aggregation. Each particle performs a random walk until it gets stuck on the substrate, or on one of the forming dendrites. You should understand why different growth mechanisms lead to two so different pictures shown here:





(1) Please download the `DLA.ipynb` notebook from bCourses and familiarize yourself with all parts. Please pay attention to the 2D array A(x,y) and understand what the ranges *maxX* and *maxY* mean. Part I is essentially above filling in the … blanks in order to obtain a working code.

(2) First please write the code for the diffusion step. Depending on the random value, *r*, please have the moving particle go left, right, up, or down each with a 25% probability. Add 7 to 10 lines of code here:

```
r = np.random.random(); # Random float:  0.0 <= r < 1.0
#based on the value of 'r', move the particle
#left, right, up, or down and change x and y accordingly
...
```

(3) Right below in the file apply periodic boundary conditions in *x* direction. If the particle has left the simulation cell on the left, let it reappear on the right, and vice versa.

(4) In the following section, your need to determine whether any of the four neighboring sites are filled. The variables *xp, xm, yp,* and *ym* are defined for your convenience. Again apply periodic boundary conditions to *xm* and *xp.* Can you think of a reason why we do not have to worry about periodic boundary conditions in Y direction?

(5) Run the code and see what happens.

(6) The code will stop earlier than anticipated because there is not enough space to accommodate 10,000 particles. Please increase *maxX* and *maxY* until there is sufficient space for all of them.

(7) The growing dendrites compete for new particles like trees compete for sunlight. Not all dendrites make it. So please make incremental adjustments to *maxX* and *maxY* so there is just one surviving tree when the last of the 10,000 particles has been attached. This completes part I.

Part II: Imagine a stream of particles through a pipe with a sticky lower surface. To simulate a flow in a very simple way, I suggest altering the diffusion probability in the X directions so that the particle moves to the right with 40% probability, with only 10% probability to the left. Leave the probability for the vertical motion unchanged at 25%. Reset *maxX* and *maxY* to your values from section (6). Run your code again and determine whether the dendrites grow with the stream (like stalactites) or against it (like stalagmites). Give an interpretation for your observations!

Part III: Finally we want to see if we can adapt the program to make more compact structures. The main problem in the existing code is that new particles get stuck too early before they can diffuse further down to fill gaps in the existing structures.

Main correction: A particle will not necessarily get stuck the first time it passes by a filled site. For such an event, we introduce a sticking probability p. If the diffusing particle "hits" an existing structure it only sticks there with probability p but otherwise keeps diffusing. In part I, the particle would always stick, which is equivalent to setting $p = 1$.

Around the existing *if* statement, where you check if any neighboring sites are filled, you need introduce another *if* statement where you check if a second random number of less than p before you attach the moving particle. y Now introduce another *if* statement around the existing *if* statement.

Once this is working, perform DLA simulations for different values of p. Choosing p very small should lead to the most compact structures.