2 Assignment Set 2

2.1 Diffusion Limited Aggregation

The diffusion-limited aggregation (DLA) is a growth model based on diffusing particles. The growth is started with a single seed (just a small square object in a single lattice point, at the bottom of the computational domain). As described in detail in the lecture notes, DLA can be modeled by solving the time independent diffusion equation (i.e. the Laplace equation), locating growth candidates around the cluster, and assigning a growth probability p_g for each growth candidate, as a function of the concentration of diffusing nutrients at that growth location. Next, a single growth candidate is added to the cluster with probability p_g. After this growth step, the diffusion equation is again solved, and the process is iterated for a large number of growth steps. You currently have almost all the tools available for a simulation of the DLA growth model. Let us take a closer look at the growth model itself, allowing you to insert this into your programs. Figure 3 shows a possible configuration of an object (the filled dots) and the growth candidates (the open circles). A growth candidate is basically a lattice site that is not part of the object, but whose north, east, south, or west neighbor is part of the object.

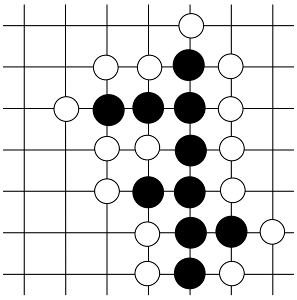

Figure 3: DLA cluster (filled circles) with growth candidate sites (open circles)

The probability for growth at each of the growth candidates is calculated by

$$p_g(i,j) = \frac{c_{i,j}^{\eta}}{\sum_{\text{growth candidates}} c_{i,j}^{\eta}}$$

The parameter η determines the shape of the object. For $\eta = 1$ we get the normal DLA cluster. For $\eta < 1$ the object becomes more compact (with $\eta = 0$ resulting in the Eden cluster), and for $\eta > 1$ the cluster becomes more open (and finally resembles say a lightning flash).

Modelling the growth is now a simple procedure. A set is created of all growth candidates with their associated weights, and a single candidate is chosen. *Hint: use numpy.choice with the p parameter for this*

A. *(4 points)* Implement the growth model, paying special attention to the calculation of the growth probabilities. Run a number of growth simulations. Try to do this for a domain of size 100×100 and investigate the influence of the η parameter. Can you still optimise by setting your ω parameter in the SOR iteration to a specific value?

Hint: the SOR iteration is run over and over again on a slowly growing object. As the growth step is constructed in such a way that on average only one lattice site is grown to the object, the concentration fields will hardly change. Therefore, it is advantageous to start a new SOR iteration with the solution of the previous growth step. Also, you can start the simulation with the analytical result for the empty system, the linear concentration gradient of eq. (5).

Optional: *(1 point)* Think of a way to reduce the time required to solve the diffusion equation. Compare your results for the 100×100 grid in question A and try larger grid sizes. Some suggestions: parallelize one of the iteration schemes in the previous exercise set, and possibly use a GPU to do the calculations.

2.2 Monte Carlo simulation of DLA

DLA can also be simulated by releasing random walkers on a grid, and letting them walk until they hit the cluster. When they hit, the walkers are stopped and become part of the cluster.

One random walker at a time is released in the system. It moves in steps, which are randomly chosen to be one lattice point up, down, left, or right. If the walker reaches a cell neighboring the cluster, the walker is stopped there, so that the cell with the walker becomes part of the cluster.

To simulate with the same boundary conditions as above, start the walkers on a randomly chosen point on the top boundary. If a walker walks out of the system on the top or bottom boundary it is removed and a new one created instead. If it walks across the left or right boundary, it should enter the system from the other side, as periodic boundary conditions were assumed in the horizontal direction.

B. *(2 points)* Implement the Monte Carlo version of DLA. Compare the resulting cluster to those obtained with the diffusion equation.

C. *(1 point)* In this model, the η parameter is no longer easily variable, it is fixed to 1. However, another parameter can be introduced, namely a sticking probability p_s. The sticking rue can then be stated in the following way: if the walker enters a cell which is a neighbor of the cluster, it stops there with probability p_s. If it does not stick, the walk continues as normal. The walker is however not allowed to move into a site belonging to the cluster. Run the simulation for different values of p_s, and plot the results. How does the cluster shape depend on p_s?

2.3 The Gray-Scott model - A reaction-diffusion system

The Gray-Scott model (J. E. Pearson, *Science*, Vol 261, 5118, 189-192 (1993)) describes a system of chemical reactions, involving the two chemicals U and V. Both chemicals diffuse in the system, and also react with each other. The reaction rate at any point in space, is determined by the *local* concentrations of U and V. The reactions are:

$$U + 2V \rightarrow 3V \tag{15}$$

$$V \rightarrow P \tag{16}$$

U is continuously fed into the system. It then reacts with V to produce more V. V spontaneously decays into P, a reaction product which is not interacting with U and V. The first reaction is said to be autocatalytic, since the reaction product V enhances the production of itself.

If we let u and v denote the concentrations of U and V, the following equations can be formulated.

$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + f(1-u), \tag{17}$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (f+k)v. \tag{18}$$

Here, f controls the rate at which U is supplied, and $f + k$ controls the rate at which V decays. For different values of f and k a large variety of behaviors can be observed. Some result in stable patterns, while others remain time-dependent.

D. *(3 points)* Implement the Gray-Scott model in two dimensions. Explain the discretization and how the equations and boundary conditions are implemented in the program (you may choose which boundary conditions to use). Plot the resulting concentrations of U and/or V for several choices of the parameters. The time-dependent diffusion program from Set 1 can be used as a base. In this case, there are two variables to keep track of. For parameters, start with $\delta t = 1, \delta x = 1, D_u = 0.16, D_v = 0.08, f = 0.035, k = 0.060$. For the initial conditions, you can take $u = 0.5$ everywhere in the system, and $v = 0.25$ in a small square in the center of the system, and 0 outside. Try adding a small amount of noise too.