PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022

Growth Models

Abstract: Growth models are applicable in a plethora of different areas. Ranging from economic to biological growth. Studies and experiments within the domain of growth analysis can give us a better understanding of how things accumulate. They may also predict outcomes of real world growth scenarios.

Introduction: For this experiment, we will explore the essence of the Eden cluster, DLA model, and percolation. We first investigated the Eden cluster, which describes the growth of specific types of clusters such as bacterial colonies and deposition of materials. These clusters grow by random accumulation of material on their boundary. Next we investigated Diffusion Limited Aggregation (DLA), which is usually studied in 2 dimensions as a model of fractal growth processes such as branching, lightning, snowflakes, mineral deposits, and coral. Finally we observed percolation probabilities, percolation can be described as the random distribution of any given substance “shotgunned” onto an area of interest. Percolation is said to take place if one of those clusters spans the entire area. Percolation is a physical phenomenon, for example the motion of ground water through soil, oil oozing through a porous rock, or burning of a forest.

Methods: To begin, the problems must first be initialized.

1. For the first exercise, we were tasked to create a two-dimensional Eden cluster model and grow systems of various sizes. For this code we first created a matrix of arbitrary dimensions. Values of zero were given to each index within the matrix, besides the very center of the matrix. Next, a second matrix was created with a star-like search element. The search element was created using the command “circshift”. This searcher was then used to fire approximately 35000 random growths. Growth would take place within the first matrix only if the search element was within the boundary of the growing cluster. This was achieved within the matlab code by “anding” the 2 matrices and summing the “anded” matrix. This “sum-and” operation only returns a growth if the searcher was within range of the previous cluster. This operation will also be referred to within the DLA model. Finally, a spy command was used for the matrix to plot the final results. Examples for the 100x100 and 50x50 Eden clusters can be observed within **Figure 1** and **Figure 2** respectively. The difference in model dimensionality can be observed by the first zero matrices.

PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022

Figure 1: Matlab Code for Eden Cluster of size 100x100

```
clc
clear all
close all
G=zeros(101);
G(50,50)=1;
for i=1:35000
    x=randi(101);
    y=randi(101);
    S=zeros(101);
    S(x,y)=1;
    S=S+circshift(S,[1 0])+circshift(S, [-1 0])+circshift(S, [0 1])+circshift(S, [0 -1]);
    T=and(G,S);
    if T(49,51)==1
        disp(T(51,51));
    end
    tstar=sum(sum(T));
    if tstar~=0
        G(x,y)=1;
    end
end
spy(G)
```

Figure 2: Matlab Code for Eden Cluster of size 50x50

```
clc
clear all
close all

G=zeros(51);
G(25,25)=1;

for i=1:35000
    x=randi(51);
    y=randi(51);
    S=zeros(51);
    S(x,y)=1;
    S=S+circshift(S,[1 0])+circshift(S, [-1 0])+circshift(S, [0 1])+circshift(S, [0 -1]);
    T=and(G,S);
    tstar=sum(sum(T));
    if tstar~=0
        G(x,y)=1;
    end
end
spy(G)
```

2. For the second exercise, we were tasked to create a DLA model and grow a cluster showing its shape along the way (for several different sizes at least). We were also asked to show the diffusing path for one walker. For this code, we first created 3 matrices of arbitrary dimensions. Values of zero were given to each index within the matrix, besides the very center of the spied matrix. The difference between the eden cluster and dla occurs whenever the program fires a random new element and the program will only continue if the sum of the “anded” matrices did not equal zero. That means that the randomly fired element landed within the range of the origin point. Going further, the 3rd matrix was then given the first 2 matrix elements. The “sum-and” operation was used for the first and 3rd matrices. This was used in the conditions of a while statement, stating that while the condition was equal to zero, the program would walk a new element based on a random number generator. Next, the original matrix was “ored” with the second matrix. This provided us with our growth for each iteration of the for loop. This entire operation was executed a total of 400 times. Finally, a spy command was used for the matrix to plot the final results.

PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022

Figure 3: Matlab Code for DLA Model

```
for i=1:400
    x=randi(51);
    y=randi(51);
    S=zeros(51);
    Sfriends=zeros(51);
    S(x,y)=1;
    if sum(sum(and(M,S)))~=0
        continue
    end
    i=0;
    Sfriends=S+circshift(S, [1,0])+circshift(S, [-1,0])+circshift(S, [0,1])+circshift(S, [0,-1]);
    while sum(sum(and(M,Sfriends)))==0
    move=rand;
        if move<0.25
            S=circshift(S, [1 0]);
            Sfriends=circshift(Sfriends, [1 0]);
        end
        if move>0.25 && move<=0.5
            S=circshift(S, [-1 0]);
            Sfriends=circshift(Sfriends, [-1 0]);
        end
        if move>0.5 && move<=0.75
            S=circshift(S, [0 1]);
            Sfriends=circshift(Sfriends, [0 1]);
        end
        if move>0.75
            S=circshift(S, [0 -1]);
            Sfriends=circshift(Sfriends, [0 -1]);
        end
        i=i+1;
    end
    M=or(M,S);
```

3. For the last exercise, we were tasked to create a grid of cells and find the percolation threshold for a number of trials. This was simulated for several sizes. Within the matlab code we used various for loops to iterate the percolation. Within the first for loop, the probability and trial indexes were placed to keep track of the amount of trials and increase probability over time. Within the second for loop contained a matrix of zeros that would use logical arguments to “shotgun” or fill the matrix with elements. The amount of points created was proportional to the probability in the previous for loop. A command “bwlabel” was also utilized within this for loop. Once the random points were filled, bwlabel would group the clusters and assign the same value for each given cluster. Within the final for loop, we checked whether percolation occurred for a given simulation. This was achieved by checking the value of the first column and final column. If the first and final columns had equivalent values assigned from bwlabel, then the trial was said to percolate. If the trial percolated then an if statement indexed the amount of percolations for each trial. Finally a plot was created with the percolation amount vs probability.

PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022

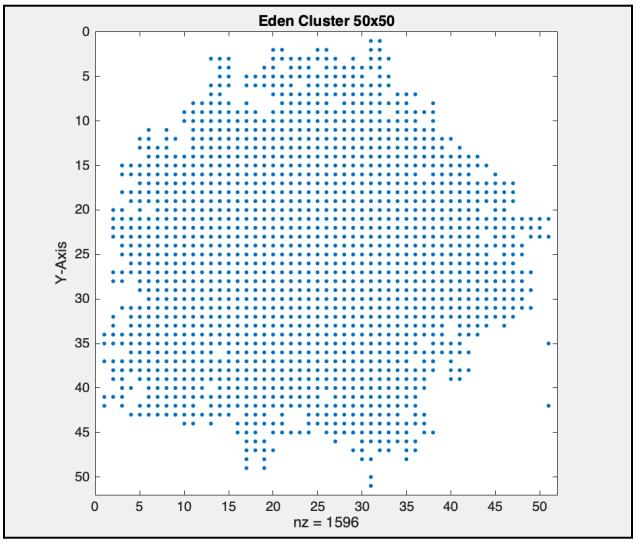
Figure 4: Matlab Code for Percolation Simulation

```
clc
clear all
close all
    for p=1:100
        prob(p)=p/100;
        ntrials(p)=0;
        for f=1:5000
            M=zeros(20);
            rfil=rand(20);
            M(rfil<=prob(p))=1;
            Try=bwlabel(M,4);
            for j=1:max(max(Try))
                [r,c]=find(Try==j);
                if min(c)==1 && max(c)==20
                    ntrials(p)=ntrials(p)+1;
                    break
                end
            end
        end
    end
plot(prob, ntrials/5000)
xlabel('Probability');
ylabel('Percolation');
```

Results:

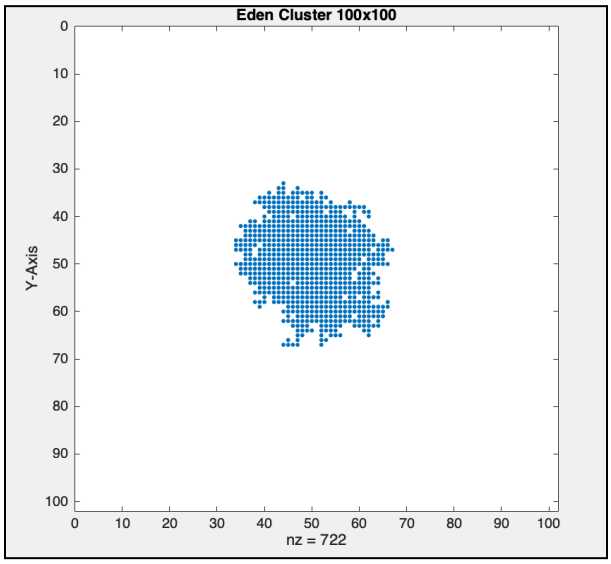
1. For exercise 1, the programs were executed and the results can be observed within **Graph 1 & Graph 2.** The results were as expected, eden clusters were created for each simulation. It could also be observed that increasing the overall area in which clusters can spawn, decreased the cluster size. This is most likely due to the fact that the searcher had a lower probability of touching the cluster due to a larger area.

Graph 1: Eden Cluster 50x50



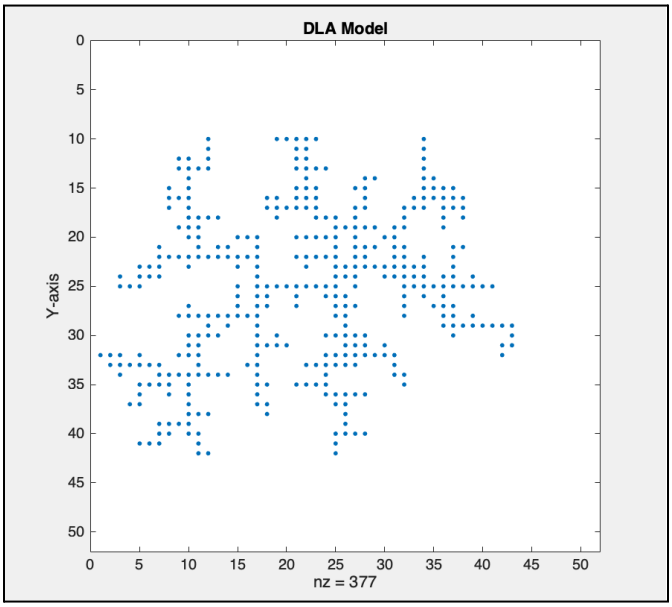
Graph 2: Eden Cluster 100x100

PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022



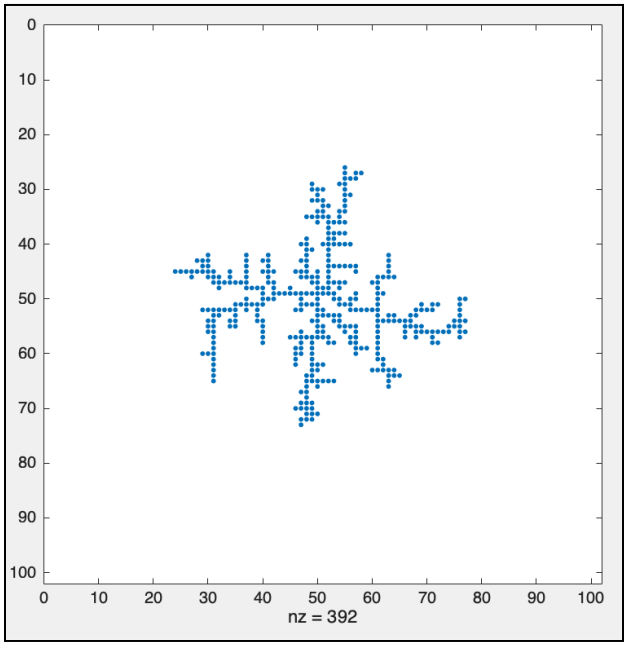
2. For exercise 2, the program was executed and the results can be observed within **Graph 3 and Graph 4.** The results were as expected, DLA models were created for each simulation. It was observed that increasing the size of the matrices would cause the program to run longer. This is mostly due to the while loop used within the program. Using the same logic as the eden cluster, due to the larger area, this decreased the probability of a growth occurring. Which would cause the program to get stuck in the while loop until a growth occurred.

Graph 3: DLA Model 50x50



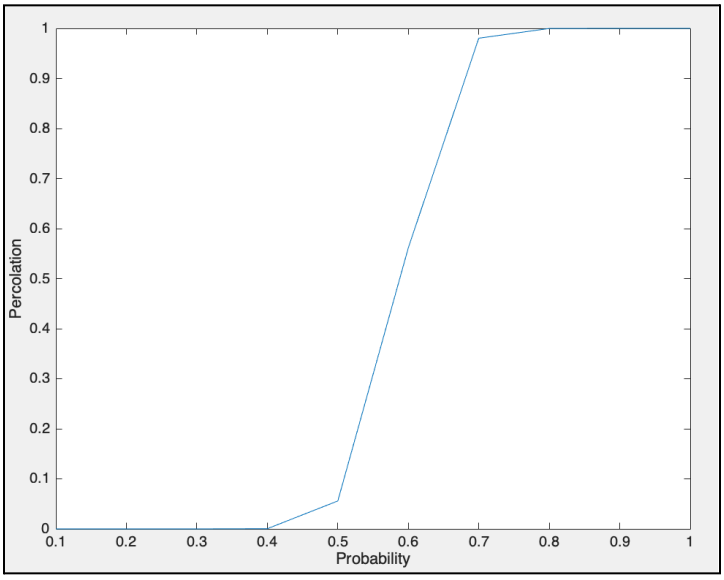
PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022

Graph 4: DLA Model 100x100



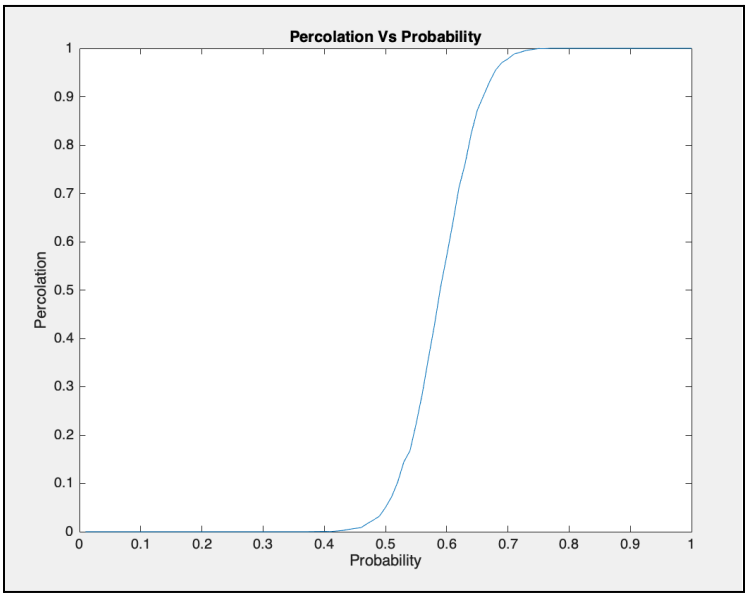
3. For exercise 3, the programs were executed and the results can be observed within **Graph 5 and Graph 6.** From these results, it can be noted that increasing the range of probability gave a smoother graph. This is mostly due to the allowance of more data points and trials. The percolation threshold can also be observed within these graphs. From the graphs we determined that the percolation threshold for these given simulations fell within the range of 0.45-0.6. Within this range of probability, the simulations are more likely to span the area and percolate.

Graph 5: Number of Spanning clusters that Percolate for probability 10



PHY 291- Computational Physics
Homework 6
Trenton Cathcart
05/04/2022

Graph 5: Number of Spanning clusters that Percolate for probability 100



Conclusion: This experiment allowed us to explore a few scientific models of growth. First, we learned how a randomly distributed growth model occurs(eden cluster). This model can be used in several different areas of study, but most notably cell growth such as tumors or other organisms. Next, we investigated the essence of a DLA model, which represented a more patterned growth distribution which usually returned a fractal like structure. Finally, we observed the percolation threshold, which for our given system seemed to percolate between 0.45-0.6. This model can be used to study many ecological areas, as well as anything that requires a gradual span across a given area. While this experiment has concluded, there are still endless amounts of growth models to be discovered and investigated.