# Homework2

Maedeh Karkhane Yousefi-98100991

October 20, 2021

# 1.Julia Set

For this code I tried creating a 2 dimensional array, filled with zero numbers. In the code I made a loop through an specific range of numbers, representing the real and the imaginary portion of the initial number. In the innermost loop, the function$f(z) = z^2 + c$ is operated on the point $z$, n times and meanwhile, there is condition checking whether the absolute value of $z$ is beyond the boundary (the given formula $\frac{1+\left|1+4*\sqrt{|c|}\right|}{2}$) or not. There are also two variables representing the x pixel and the y pixel in our resulting image, which are initially set to zero and increase by one after each loop, placing the the absolute value of the final point-if true in the mentioned condition before- in the these specified positions in the array.

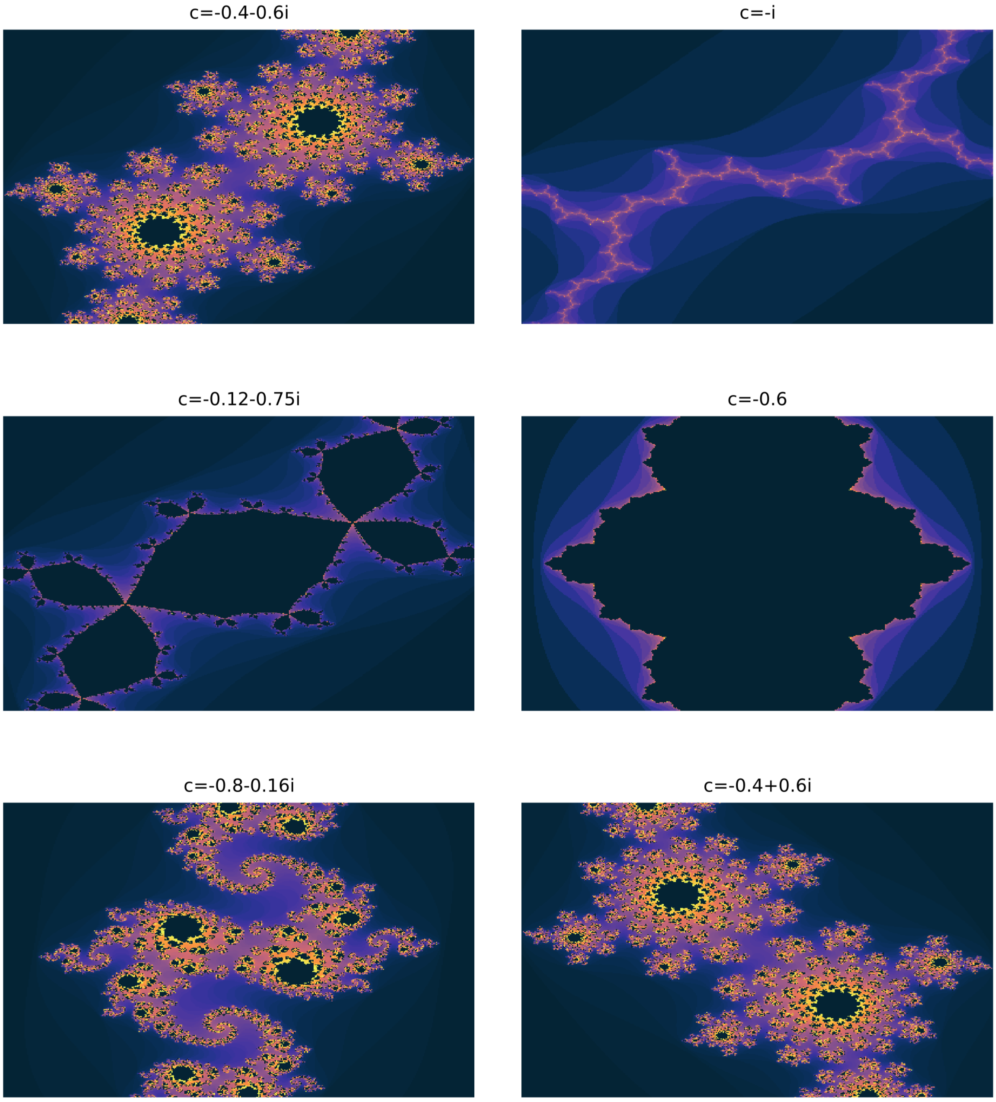

Figure 1: Julia Set is plotted for different values of c. Points are assumed to be between -1-i and 1+i with equal steps of 0.005 between each real and imaginary portion of the point. The function is $f(z) = z^2 + c$ and iteration on every point is set to be 10.

# 2.Random Ballistic Deposition

assuming $L$ to be the length of the line, I constructed a $L\times L$ *array* filled with zero numbers. In a loop through the total number of particles that we aim to sit on the line, in each step we introduce a random number which denotes the column that particle is. In the next loop we check all the rows in that column from bottom to top, for the first entry that is zero, or in the other words, empty. An specific number will be associated to that entry -other than zero-, confirming that pixel's color in our heatmap plot. The color is going to change after every $L\times 10$ step.
for plotting the mean hight of the layer in every time step I wrote a function, which counts all the entries in every column that are not empty, namely "are filled with particles ". Hence, the mean hight value can be calculated using the *mean() function* available in the *statistics package* in Julia.

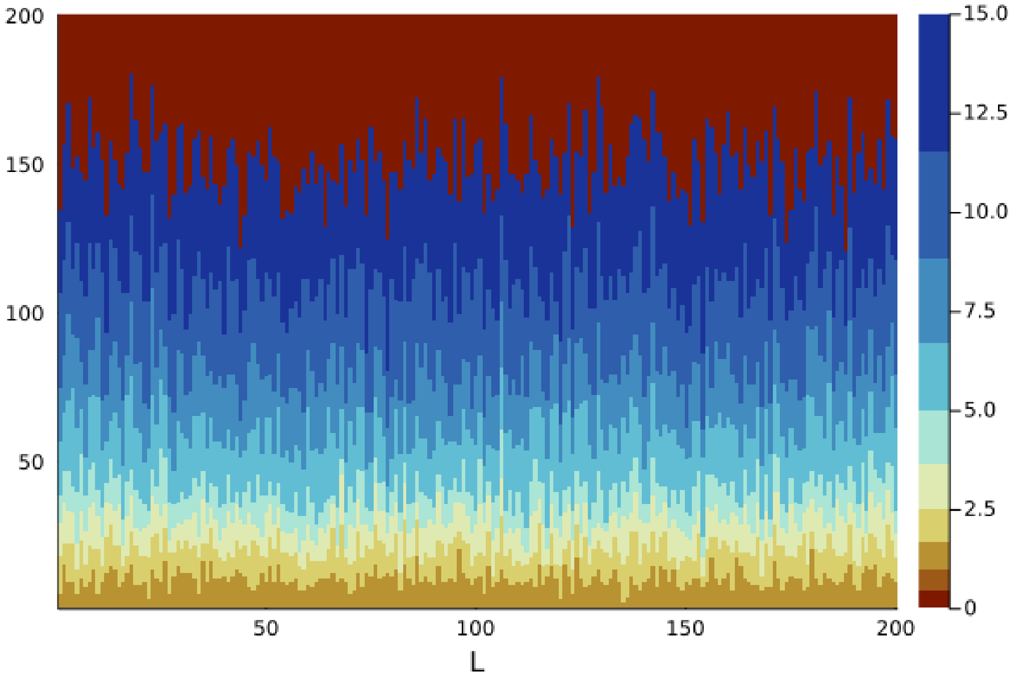

Figure 2: The model's dynamic for 30000 particles and a line of L=200 is shown in this plot.

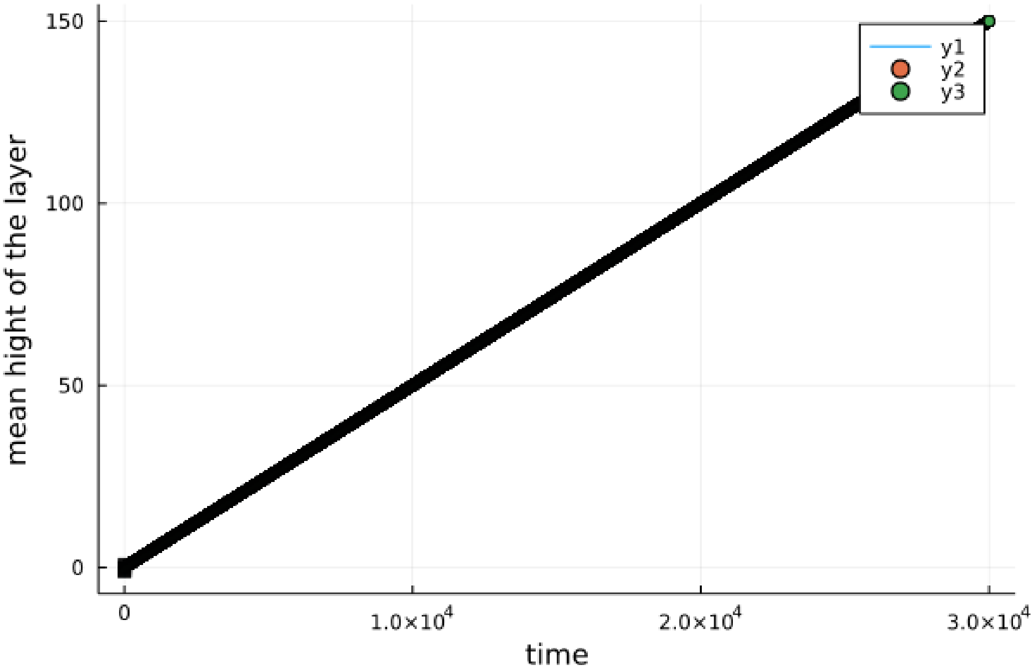

Figure 3: Mean Hight of layer over time for 30000 particles and a line of L=200 is shown in this plot.

For this part, we can no longer assume time to be continuous. We shall assume that the time intervals' increment are exponential and therefore, the amount of particles falling between each time interval is increasing exponentially, too. repeating the same procedure, but including this consideration into account, leads us to call the function I've written for calculating the The Standard Deviation at the end of the loop. The *STD function* works like the *mean_ calculater function* except that it returns a list of *std*s.

In order to show the the deviation I have set a *run_num*, by which the code loops through the whole procedure and it creates a list of multiple lists of *std*s. Getting the mean and the std of the same number of elements in every list, we are going to have relatable std list and mean list to plot the error bar, too.

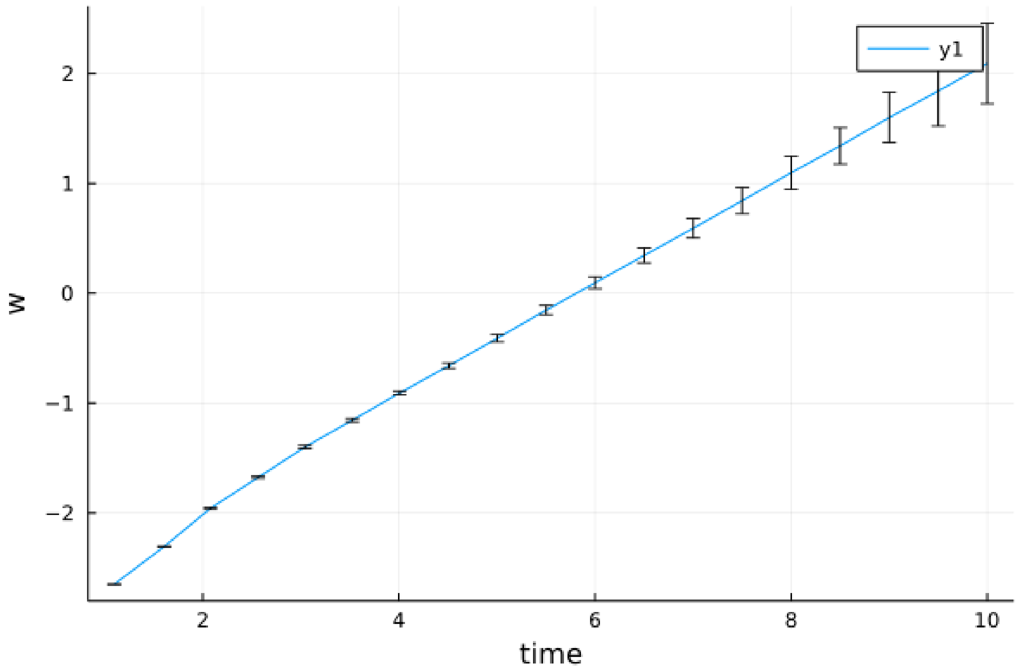

Figure 4: w over time for total particles of 1000 and a line of L=200 is shown in this plot. The initial number of particles that are going to fall in the first time-step=10.