PHYS 381 BONUS ASSIGNMENT - Julia Sets - February 21st, 2022

Author: David (Scott) Salmon - UCID (30093320)

A Julia set is a set defined by a specific function that under iteration, never approaches infinity. This function has the form: $f_c(Z) = Z^2 + C$, where Z is the number that is iterated every cycle (and is typically complex), and C is a constant. An example of a number in the Julia set is as follows:

$$\text{Suppose C} = 0\text{, and } Z_i = 0.5,\ \text{Then:}$$

$$f_c(Z) = Z^2 + C$$

$$f_c(0.5) = (0.5)^2 + 0$$
$$f_c(0.5) = 0.25$$

$$f_c(0.25) = (0.25)^2$$
$$f_c(0.25) = 0.0625$$

$$f_c(0.0625) = (0.0625)^2$$
$$f_c(0.0625) = 0.00390625$$

As is seen above, this function under iteration will never approach infinity, so Z = 0.5 is a part of the Julia set. This same logic applies when Z is initialized as a complex number. If by iteration, the complex function approaches infinity, then it is not a part of the Julia Set, and vice versa.

This process seems simple enough when using real numbers, however it quickly becomes extremely complicated once complex numbers are introduced… to the point that solving them analytically becomes practically impossible. However, computers can be used to solve them computationally.

The main process to solve a Julia set computationally is as follows:

1. Define a function that will iterate the Julia set function, f_c, repeatedly. The function needs to take a Z value and C value as an input. Inside the function, the iterative loop needs to be a while loop that stops when the magnitude of Z starts to become too large, or when the loop has run a maximum amount of times (based on recommendations from the lab manual, set the max magnitude of Z to 4.0, and the maximum amount of loops to 80.)
2. Inside this loop, the number of iterations of the loop needs to be tracked with a counter variable. This variable is what the function will return.

3. Make a zero square matrix (2D array) with whatever dimensions you desire. (based on the lab manual, set the matrix to 500x500).
4. Choose a value for C. This can either be pre-determined, or an input value from the user.
5. Next, a nested loop is required to track how many iterations are required to reach the magnitude of the Julia Set at each index of the matrix. A Z value is determined through taking the x and y values that are correlated to the current step in each of the nested loops, finding that specific index with these values in the matrix created in step 3 and then converting this number into a complex number.
6. Then the Julia Set function is called, and the Z value just determined is plugged into the function. The function returns how many iterations were required for the function to either diverge (i.e. magnitude becomes too large), or the maximum amount of loops occur. This number of iterations is then assigned to the matrix index.
7. This nested loop repeats this process for every x and y value in the matrix.
8. Finally, create a color map of the 2D matrix. This will make a visualization of the Julia set, as seen below.

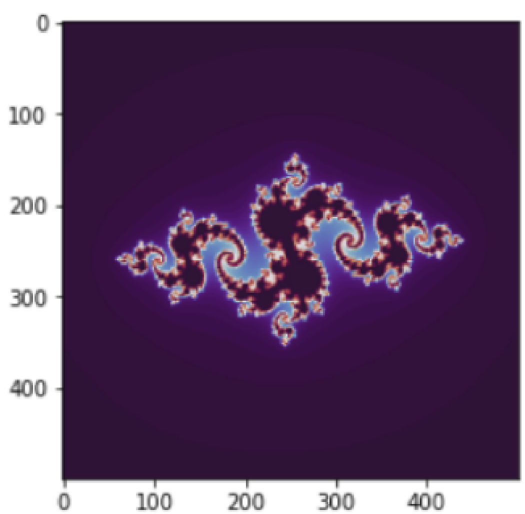

Figure 1: Julia Set with C = -0.8 + 0.156*i*

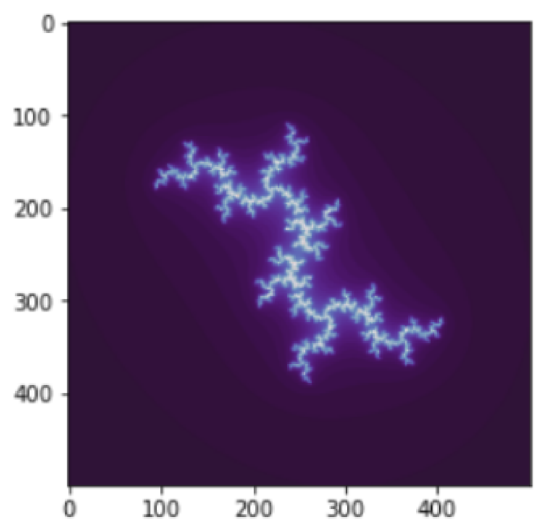

Figure 2: Julia Set with C = -0.8*i*