CS101 hw09: Fractal Plot

1. Preparation

In this section, you are to create the sine wave sin(x) pattern and plot it out. Given *X* in the range of [0, 5*pi], and *Y* in the range of [0, 1], use the matlab function *linspace* and *meshgrid* to create the x/y coordinates of a 1000x500 meshgrid in the range. Then create an image of the same size, each pixel corresponds to a coordinate in the meshgrid. In the code file hw9_prep.m, you are asked to fill in the missing code that defines the sine wave image. When you finish, you're supposed to have a plot exactly like the following:

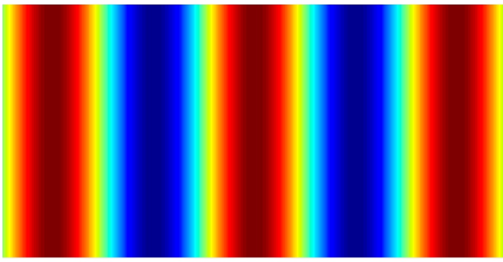

Figure 1. A sine wave pattern. Red for large values, blue for small values.

2. Fractals

Now, we should be almost ready to use the Matlab's plot tool to draw a particular type of graphic patterns called *fractals*. According to Wikipedia, a fractal is an abstract object in mathematics used to describe and simulate naturally occurring objects. It most typically exhibits the property of ***self-similarity at increasing scales***, like the left plot in Figure 2. Approximate fractal patterns are also commonplace in natural phenomena such as the pattern of coastlines, tree leaves, mountain ranges, blood vessels or DNA. In the right example of Figure 2, it is a picture of a broccoli that exhibits an interesting self-similar 3D pattern: the entire broccoli has many smaller copies of itself, and the smaller ones have even smaller versions in you zoom-in.

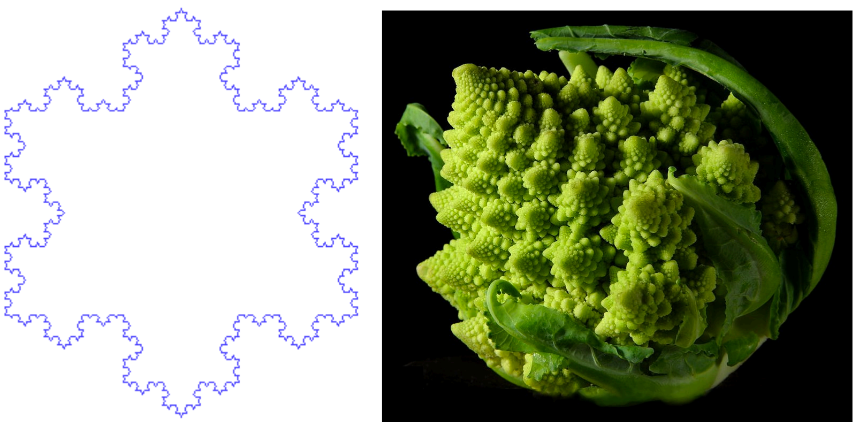

Figure 2. Left: Koch snowflake pattern. Right: a broccoli that exhibits self-similar form.

There are many fractal patterns in mathematics. The two fractal patterns we are going to implement and visualize are *Mandelbrot* set and *Julia* set – two closely related fractal sets. Their mathematical definitions are as follows.

2.1 Mandelbrot set

Starting with $z_0 = 0$, a complex number $c = (x + y * i)$ is in the *Mandelbrot set* if by applying the iteration:

$$z_{n+1} = z_n^2 + c$$

repeatedly, and the absolute value of z_n remains bounded however large n grows.
This can be formulated as:
$z_0 = 0$
$z_{n+1} = z_n^2 + c,$
$c \in M \Leftrightarrow \lim_{n\to\infty} |z_{n+1}| \leq 2$
where M is the Mandelbrot set.

2.2 Julia Set

Given a constant complex number c, a complex number $z = (x + y * i)$ is in the *Julia set* if by setting $z_0 = z$, and by applying the iteration:

$$z_{n+1} = z_n^2 + c$$

repeatedly, the absolute value of z_n remains bounded however large n grows.
This can be formulated as:
$z_0 = z$
$z_{n+1} = z_n^2 + c$
$z \in J \Leftrightarrow \lim_{n\to\infty} |z_{n+1}| \leq 2$
where J is the Julia set.

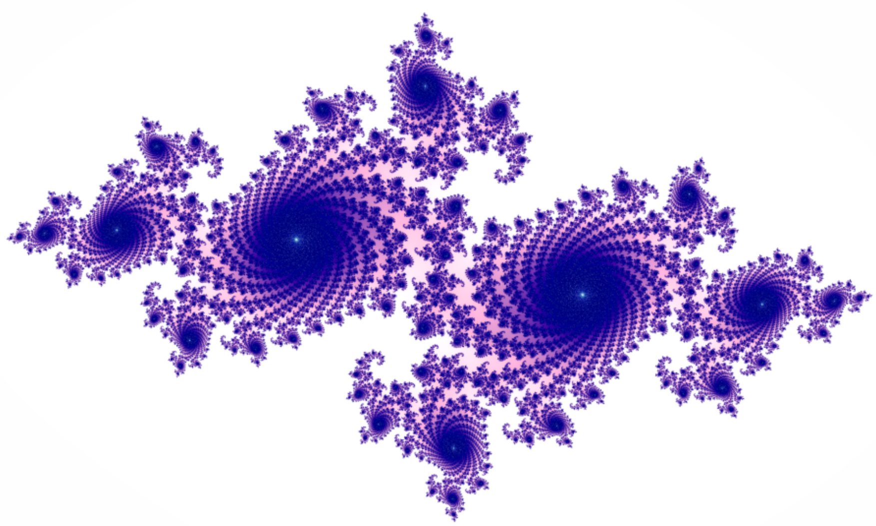

Figure 3. An exemplar Julia set plot.

3. Implementation Instructions

You are going to work in the matlab file fractal_plot.m that comes along with this homework assignment. The file has three function definitions: *fractal_plot*(), *Mandelbrot_plt*(Xr, Yr, k), and *Julia_plot*(c, Xr, Yr, k). fractal_plot() implements a simple interface that creates a window for plotting the Mondelbrot set and another window for plotting the Julia set. There is NOTHING you need to change or add to this function.

In the other two functions, Xr, Yr specify the range of X and Y over which the Mandelbrot/Julia set is defined. k specifies the number of points to sample in both Xr and Yr (which means you are going to have an image of the same height and width). What you are going to do is to fill in the missing code that assigns the value **n** to each point $c = (x + y * i)$ in the range when the absolute value of z_n exceeds the threshold 2 ($|z_n| \geq 2$). You should set the maximum n to 100. That is, if $|z_n| < 2$ for n = 1,2,…100, then stop the iteration and return the value 100. The set of points that has value 100 are taken as the ones in the Mandelbrot/Julia set. The other ones are not in the set. The two functions already provide the code for plotting (visualizing) the resultant sets. Figure 3 shows an exemplar Julia set plot.

The argument c in Julia_plot(…) is a point (a complex number) you choose from the Mandelbrot set plot. The interface provides a simple mouse click tool to pick a random c in the Mandelbrot set plot that defines a corresponding Julia set. Once a Julia set is defined and plotted. The interface allows you to zoom-in to finer scales and explore the magic in the fractal world. You can end the exploration by double-clicking on the Julia set plot and switch back to the Mandelbrot set to start over.

You may add new function definitions in fractal_plot.m for code clarity.

4. Submission

You should submit ONLY the completed fractal_plot.m file, plus 1 image of the Mandelbrot set plot, and 3 of your favorite Julia set plots. To save a figure plot, do File\Save As\... and save the result in .png format.

You MUST compress the above materials to .zip file, then submit it on http://others.zlcnup.com/cs101 before Dec 22nd, 6pm (midwinter). The system will check your information and rename your file. You should check out the information on the website. We will update on website if we have new announcements.