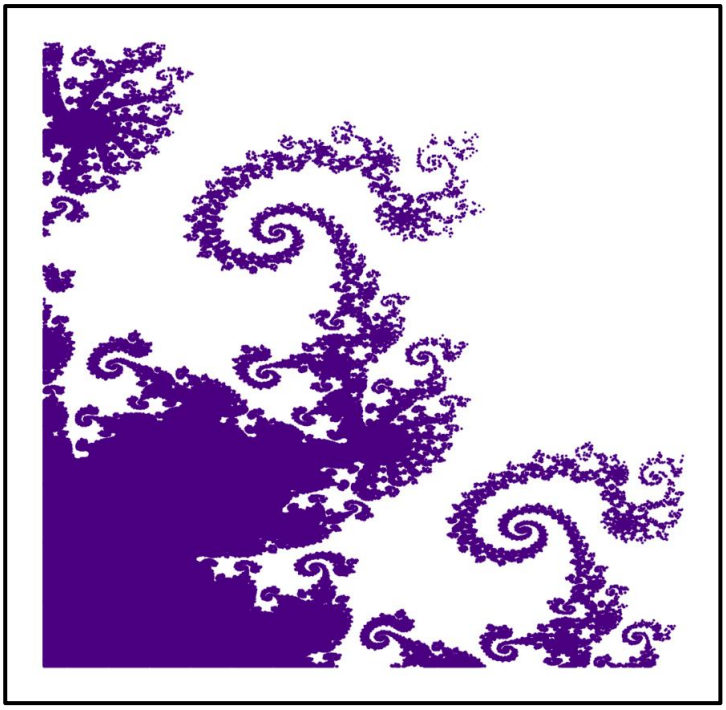
The seahorse valley



Fractals are shapes with repeated patterns on different scales. Simple fractals can be plotted using an iterative function system. This is simply the process of applying a function to itself many times. These systems are relatively easy to code and give really cool plots.

Example:
Take the function $x_{n+1} = 2x_n$. Starting with $x_0 = 1$, we get $x_1 = 2$, $x_2 = 4$, $x_3 = 8$, etc…
We can also have a parameter A: $x_{n+1} = 2x_n + A$
Taking A = 1 and $x_0 = 1$, we now have $x_1 = 3$, $x_2 = 7$, $x_3 = 15$, etc…

A famous example of a fractal with self-similarity is the Mandelbrot set.

The Mandelbrot Set **is the set of complex numbers c**, for which the iterative function

$$z_{n+1} = z_n^2 + c, \quad z_0 = 0$$

doesn't diverge to infinity. This set can be plotted as a surface in the complex plane. The boundary of the Mandelbrot set is a famous fractal curve: check this out!!
Points c for which z_n increases indefinitely are outside the surface. Points for which z_n converge are inside the surface. On the boundary are points c which diverge after a very large number of iterations. The different colours in the zoom represent the different number of iterations it took for z_n to diverge. There is also a good video by Numberphile that'll help you with the logic.

We want a plot of all points within domain $D = \{x + iy | -0.8 < x < -0.78, 0.14 < y < 0.17\}$. In this region of the Mandelbrot set, you should find patterns like in the plot above! This region of the Mandelbrot set is called the seahorse valley.
Around 100 iterations of the function will be enough. To check if z_n is diverging, use $|z|<2$ (the largest magnitude in the set is -2).

Some tips you might find useful:

- The i in python is 1j, for example $z = 2 + 3i$ is $z = 2 + 3 * 1j$
- If you want only the real or imaginary part of a complex number z, use the functions **z.real** and **z.imag**. For example, if $z = 2 + 3 * 1j$ then z.real return 2, and z.imag returns 3
- You will be doing a lot of iterations in your code so don't worry if it takes a while to run
- The bigger your figure size, the better it will look!

Please ask if you have any questions or email us!!