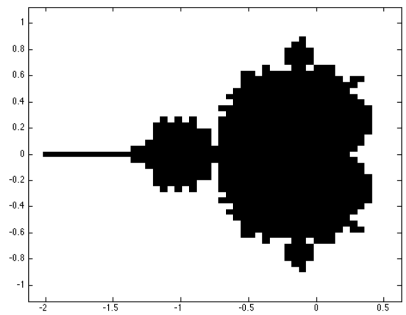
Mandelbrot Set

This lab is focused on the Mandelbrot set that was discussed in the lecture. The goal is to produce this image and to understand the algorithm for its generation:



(1) The Mandelbrot set emerges from the super simple iteration formula $z_{n+1} = z_n^2 + C$ where C is a constant and one starts from $z_0=0$. Let us assume all numbers are real for the moment. For what values of C does this series diverge? To find out, please write a simple loop the computes z_n up to $n_{max}=50$. Print all z_n. If $|z_n|$ exceeds $z_{limit}=50.0$ for any n, let us assume the series diverges. For the following C values, determine whether the series diverges or not. *(We tried this in class already.)*

C	Diverges Yes/No
–4	
–2.001	
–2	
–1.999	
–1	
0	
+0.5	
+1	
+2	

(2) Now we want to convey in the information in this table in graphical form. Generate a fine grid of C values from –4 to +2 and program a function f(C) that assume 1 or 0 depending on whether the series z_n diverging or not. The generate a plot f(C) versus C. *(You may not need to write a function f(C) as long as the plot is correct. Also this will be a somewhat boring graph but it is just a precursor to what is to come.)*

(3) Now make z and C complex variables and repeat step (1). Most likely you will only need to change a single line in your code where define C to `C=complex(–0.4,+0.2)`. The complex `abs(z)` function is defined to return $\sqrt{re^2 + im^2}$ and can be conveniently used to compare with z_{limit}.

C	Diverges Yes/No
$-0.4 + 0.2i$	
$0.4 + 0.6i$	
-1.7	
$-1.7 + 0.001i$	
-1.778	
$-1.778 + 0.001i$	
-1.25	
$-1.25 + 0.04i$	
$-0.125 - 0.9i$	
$-0.125 - 0.85i$	

(If you are not as confused as Mandelbrot was when he first noticed the peculiar results, then you may have done something wrong ;=)

(4) Finally, we want to calculate the real Mandelbrot set where we change the real and imaginary parts of $C=x+y\times i$ using a fine grid of N points in x and y directions ranging from $x=-2.0$ to $+0.5$ and $y=-1.25$ and $+1.25$. You may start with N=201. For every point C, start the iteration over z_n from $z_0=0$. Define a N×N matrix if fill it with 1 if the series has diverged within the first $n_{max}=50$ step. Otherwise set the matrix element to 0. Once every matrix element has been filled, use `imshow` to display it.

(5) Now we want to make the colors a bit more appealing. In cases, where the series z_n does not diverge, fill the matrix with n_{max} instead of 0. In cases where it diverges at some iteration $n<n_{max}$, fill the matrix with n. Does that look better?

(6) Zoom into the interval $-0.8<x<-0.7$ and $0.05<y<0.15$ and run the code again. Now increase the number of iterations n_{max} first to 75 and later to 100. What change do you see in the resulting image?

(7) Now zoom in further step by step until the size of the x and y intervals are only 0.001 or less. Adjust n_{max} as needed. Get lost in the neverworld of fractals. Save the image file and the parameters for the most beautiful image you obtained!

(8) Finally, run your code with only N=201 grid points and try these magic commands (Replace `A` by the name of your matrix):

```
%matplotlib osx
# On PCs, try %matplotlib qt
# On a Mac, could also try %matplotlib tk
# You probably need to restart Jupyter after every failed attempt!
plt.rcParams['figure.figsize'] = [5, 5]
from matplotlib import cm

min = A.min()
A -= min
max = A.max()
while (True):
    A = (A + 2) % max;
    plt.imshow(A,cmap='gist_ncar', interpolation='nearest',animated=True)
    plt.draw()
    plt.pause(0.2)
```

and see what happens.