

# Numerical Methods in Physics and Astrophysics

## Problem Set 2 : Fractals Through Newton-Raphson

### 1 Working Principle

To fulfill the necessary requirements for the assignment, the root finding program from the previous problem set is adapted to work with complex numbers and find the roots of a complex function. Two loops are then created to iterate what will be the pixels of a fractal image.

During the execution of the program, the y-values are calculated in parallel for different x-values and stores in a buffer. This buffer is then written to a file. Some brief testing indicates that using OpenMP and some primitive parallelization yielded a performance improvement of about six to ten fold.

Plotting the imaginary component of the root obtained or the number of iterations required to reach said root with a corresponding color map produces many wonderful images in the form of fractals.

### 2 Usage

A program called `calc_fractal` is created after running `make` in the project directory. Running `./calc_fractal` uses the values in `lib/constants.c` to generate the desired data in `data/fractal_data.csv` in the format  $x_0, y_0, \Re k(z), \Im k(z), \Re f(z), \Im f(z)$ , and  $\log_{10}(n)$  with  $n$  being the number of iterations needed. This counter and other parameters were set to 0 when a point did not converge.

Additionally, one can use `./calc_fractal` with two points defining a viewing window in which the fractal data is calculated (the default goes from (-2,-2) to (2,2)). In other words, one can write `./calc_fractal <p1_x> <p1_y> <p2_x> <p2_y>` with `p1` being the coordinates of the lower left corner and `p2` the coordinates of the top right corner.

Fractals are then plotted with `gnuplot` using the `.gp` file in `plots/`. Images of the fractals can be found in the same `plots/` directory with the format `fractal_plot_<ID>_<Img Vers>` with the ID corresponding to an equation and the second parameter the image version. Images with an ID of 1 were experimental.

Please note, one can change the `.gp` file in order to create images with different resolutions, output names, and dependent variables (either the  $\Im f(z)$  or  $\log_{10} n$ ).

### 3 Results

#### 3.1 First Function

We will begin by observing the results obtained from plotting the first function:

$$f(z) = z^3 - 1 = 0$$

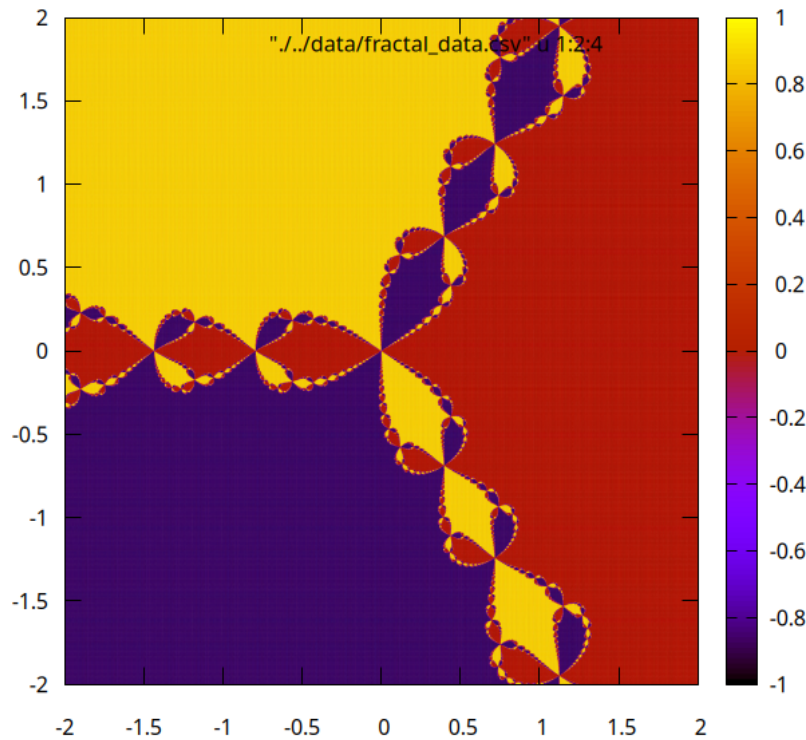


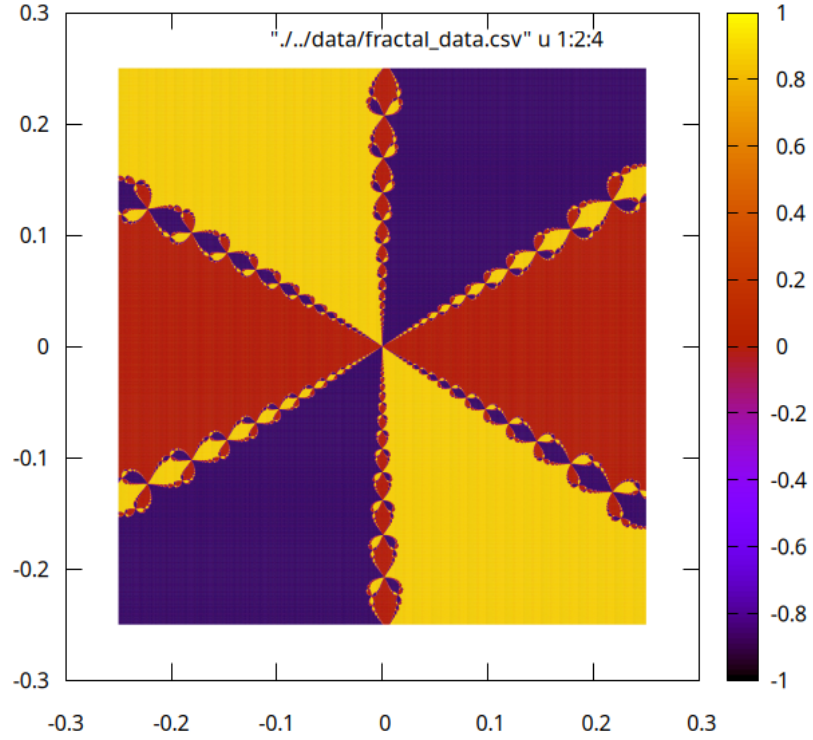
Figure 1:  $\Im f(z)$  plotted on the complex plane from  $-2 - 2i$  to  $2 + 2i$ .

Note that the color scale on the right side of the image shows  $\Im f(z)$ .

We then continue by choosing a region with complex behavior and plotting two more images:

After observing these interesting regions, we proceed by plotting the number of iterations on the complex plane, zooming into locations with interesting behavior.

As we can see upon closer examination of the complex areas in the plot, there are not only single points that need greater iteration times, but even the local areas require more computation to finding a converging answer. This can be seen by the shifting of the colors and even the scales as one zooms further into the image.

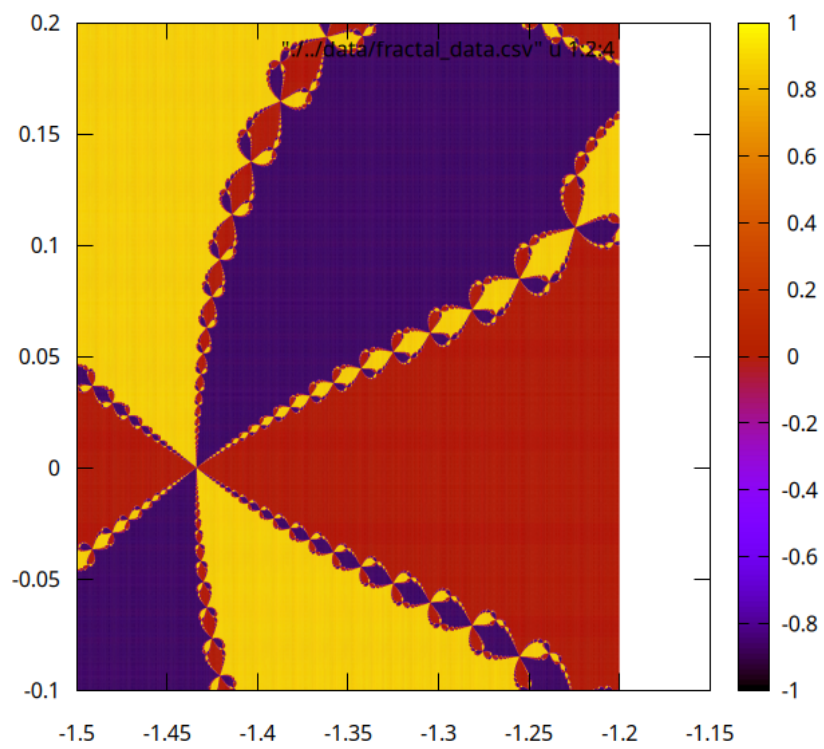


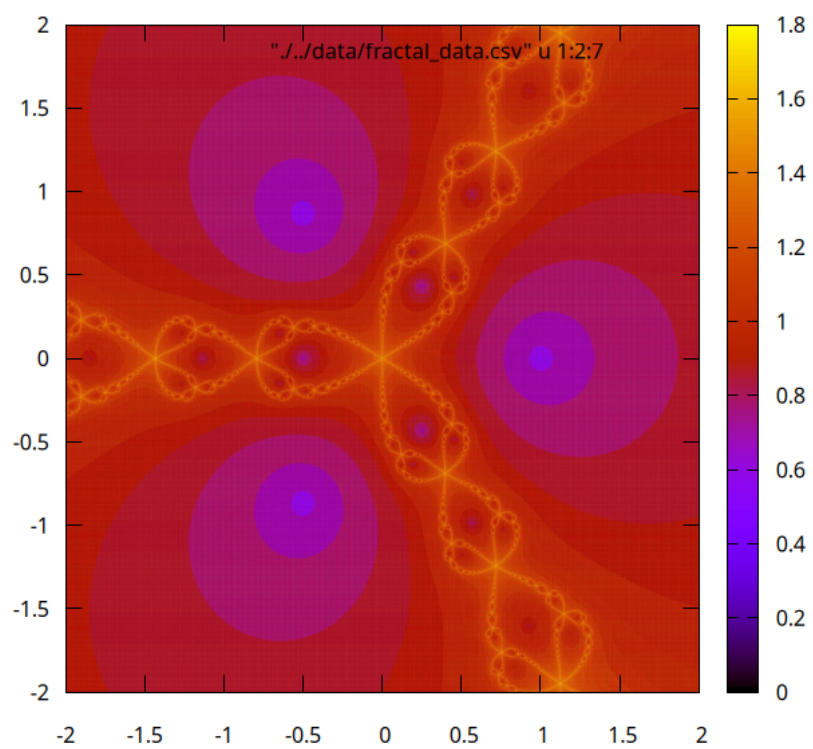
### 3.2 Second Function

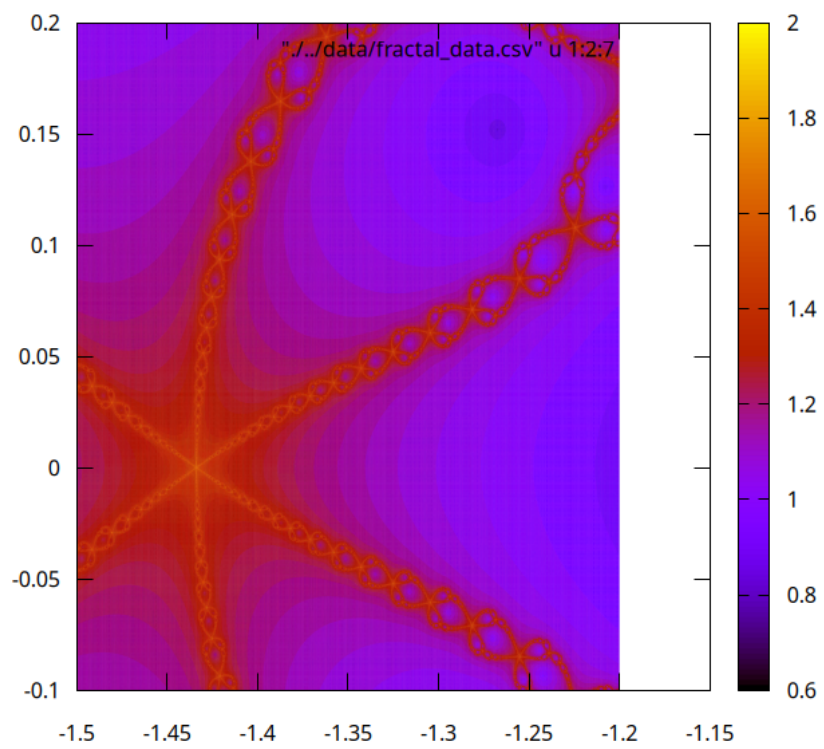
We further continue the exercise by examining a second function given as follows:

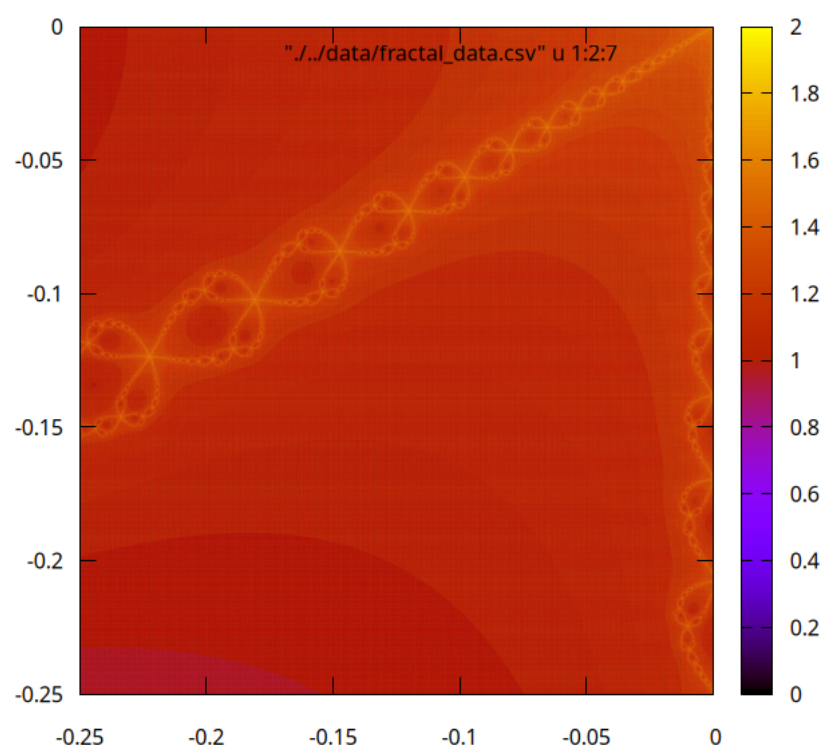
$$f(z) = 35z^9 - 180z^7 + 378z^5 - 420z^3 + 315z$$

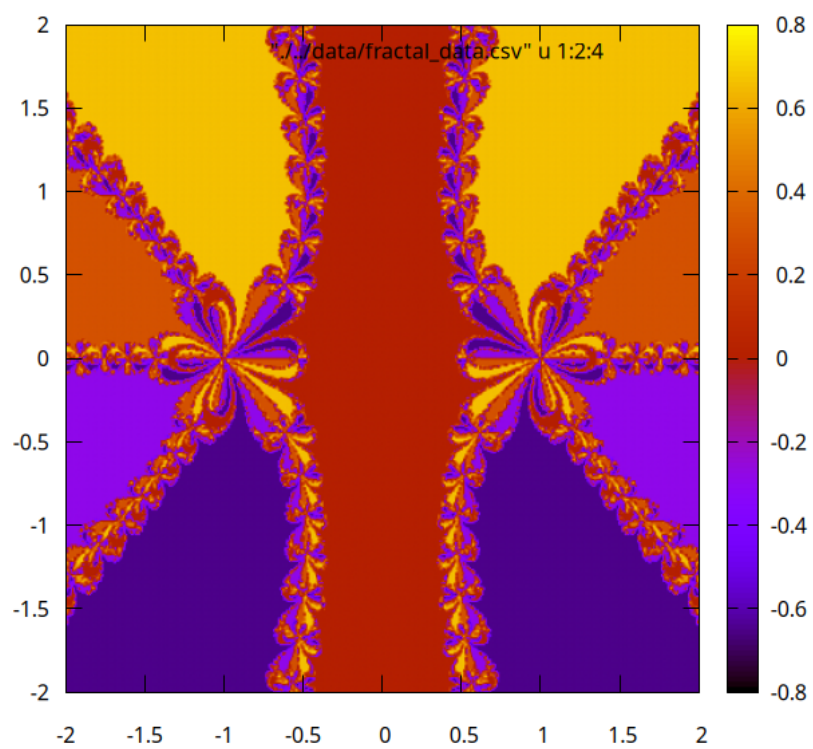
This function shows more complex behavior and even regions in which the program was not able to converge to a root (resulting in me troubleshooting overflows).



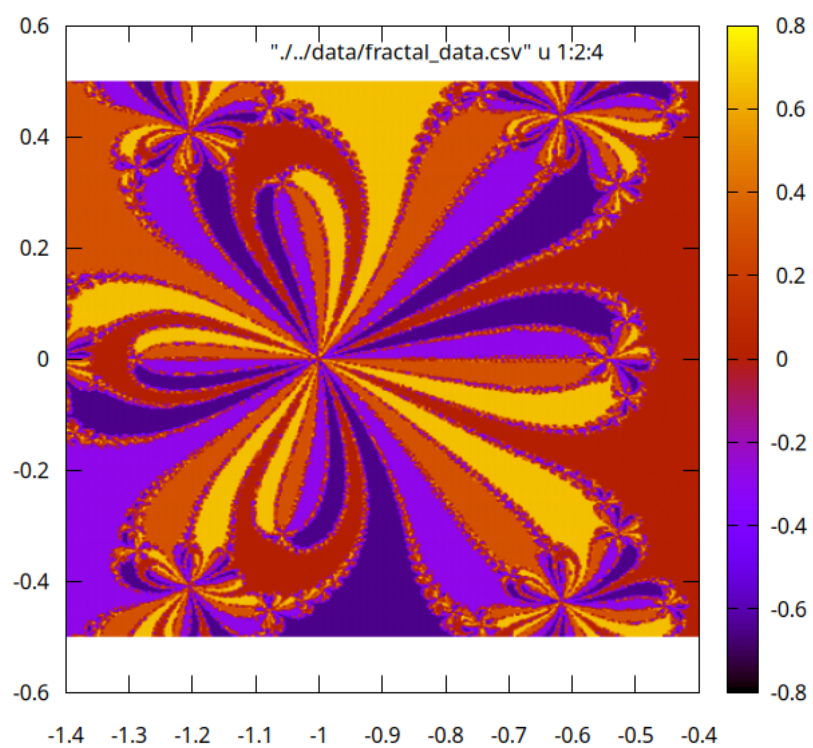












And we finish the assignment by examining the convergence plot of this function.

