Computational Physics 1

Jack Donahue

September 15, 2017

Abstract

In this assignment we begin trial projects of simulation and data analysis in scientific computing with Python. We create a numerical approximation to the Mandelbrot set, exploring ideas in simulation and data presentation. Finally, we reconstruct best-fit linear analysis used in the Millikan drop experiment to reproduce a historical calculation of Planck's constant.

Introduction

The Mandelbrot set is a fractal set that is defined by the recursion relation $z' = z^2 + c$ where $z, c \in \mathbb{C}$. We choose a point $c \in \mathbb{C}$ and, starting with $z = 0$, we repeatedly apply the recursion relation. If the point ever grows larger than 2 in absolute value, we say it is outside the set. If, for some number of iterations, it does not we say that point is in the set. The points left over form the Mandelbrot set. The Mandelbrot set was famously first visualized using computers, which could do the repetitive calculations needed to populate a graph. The study of fractals, especially with computers continues to this day and they are used in computer graphics and other fields.

The Millikan oil drop experiment, carried out in the early 1900s, was the most accurate measure of the electric charge, winning Millikan the Nobel Prize in physics. In this reconstruction, we will do a simple linear best-fit approximation to tease out the value of Planck's constant from data, using current values of the electric charge.

Method

First, the user who initiates the program must choose a resolution size of the picture to be generated, N and the maximum number of iteration the code should run for. This allows the user to create quick, rough sketches, as well as more detailed plots. The area to be investigated is always the 2 by 2 square in the complex plane centered around zero. This area is then broken up into N^2 points in a NxN array. The array is labelled spatially. This means that for our array a, $a_{0,0}$ corresponds to the upper left corner and $a_{N-1,N-1}$ to the bottom right corner.

The recursion relation $z' = z^2 + c$ is then carried out for each point c, in the array. If the magnitude of z ever exceeds 2, the loop is broken and the iteration number is returned. If it never exceeds 2, then the loop stops at the maximum iteration, which is returned. The log of the returned iteration number is then put in the corresponding cell in the array. Most numbers not in the set leave in a few iterations, so the log is used to more finely distinguish between the early departures. The data is then plotted in jet so that it's colorfully vibrant.

The Millikan oil drop analysis is much simpler. The data is imported and averages and variances of the data are calculated. These averages then are used to calculate the slope, m, and intercept, c, of the best-fit line. Planck's constant, h, is then calculated from this best-fit line's slope and the electric charge, e, related by $em = h$.

Results

The Mandelbrot code described in the methods section output Figure 1 for $N = 1000$ and a maximum of 100 iterations. We can we can see quite a high level of detail for this resolution, the qualitative fractal nature becoming easy to see.

The data and calculated best-fit line are shown in Figure 2. Visually, the line fits the data well. The slope from this best-fit line was used to calculate Planck's constant $h = 6.549340 * 10^{-34} J \cdot s$. This is 4.6% different from current quoted values of Planck's constant.

Discussion

Based on my knowledge of the Mandelbrot set and some image searches, the set produced here has no noticeable differences. The Mandelbrot set is a beautiful mathematical object and it's ease in computation speaks to the power of computing. Computers excel in the repetitive calculations needed to reproduce such an image, and many such applications of scientific computing.

The Millikan oil drop analysis exemplifies the other great power in computing, data analysis. The calculated value of Planck's constant came very close to current day quoted values, which is impressive considering the original data was taken in the first decade of the 1900s.

Conclusion

The Mandelbrot set is an example of simple mathematical laws creating some beautiful structures. Likewise, the coding of such laws emphasize the utility of scientific computing in aiding our understanding of the outcome of such laws. It would be interesting to see how the computation of the set could be improved. Ideally, one would want a finer mesh around the interesting borders of the set. We could imagine some possible solutions. One could implement a sparse set, then implement fine sets where the points are neither bounded nor always bounded. That way we would end up with more of our million points clumped around the boundary, revealing finer detail.

The Millikan oil drop analysis, though simple, is exponentially more useful as its methods are extended to more dimensions. Alternatively, we could try to implement a method that guesses a slope and intercept iterating towards a final solution. Although this may not converge nor be the quickest, we could imagine such iterative methods become useful as dimensions and data sets get larger.

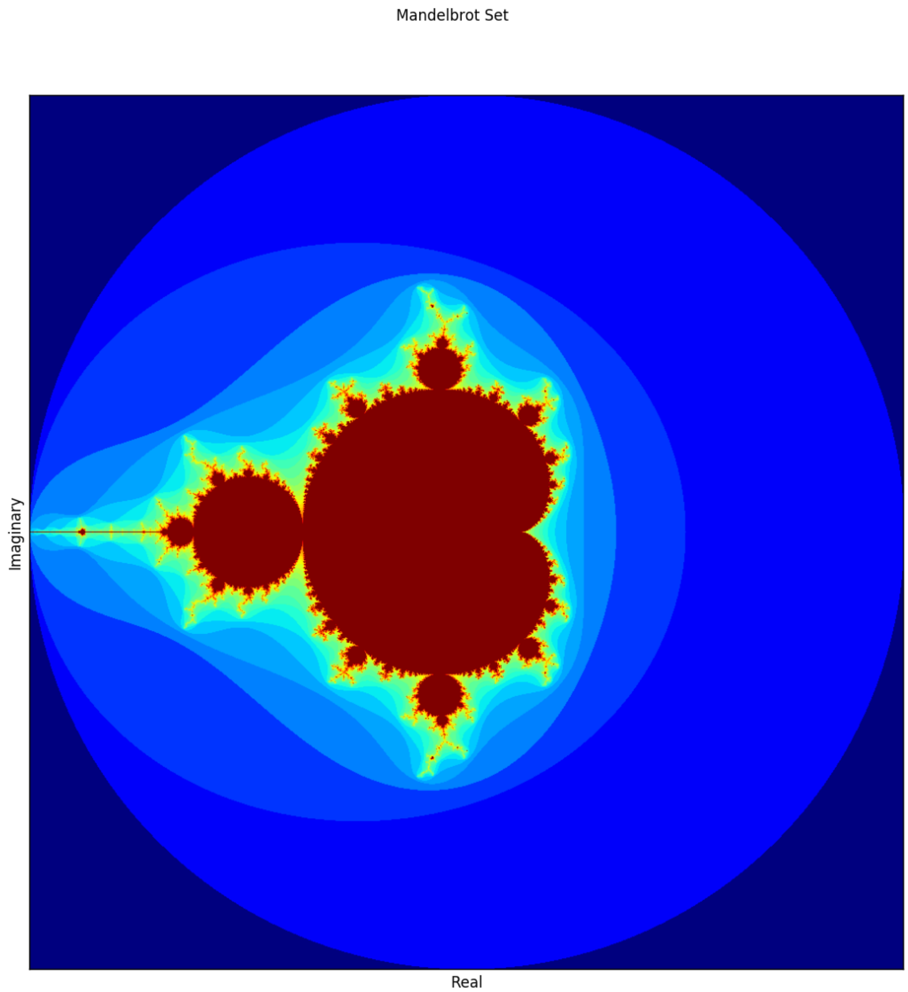

Figure 1: The Mandelbrot set calculated for a 1000 by 1000 mesh sitting on the 2 by 2 square in the complex plane. As the colors get more red, the more iterations before the solution became unbounded. Deep red maxed out at 100 iterations.

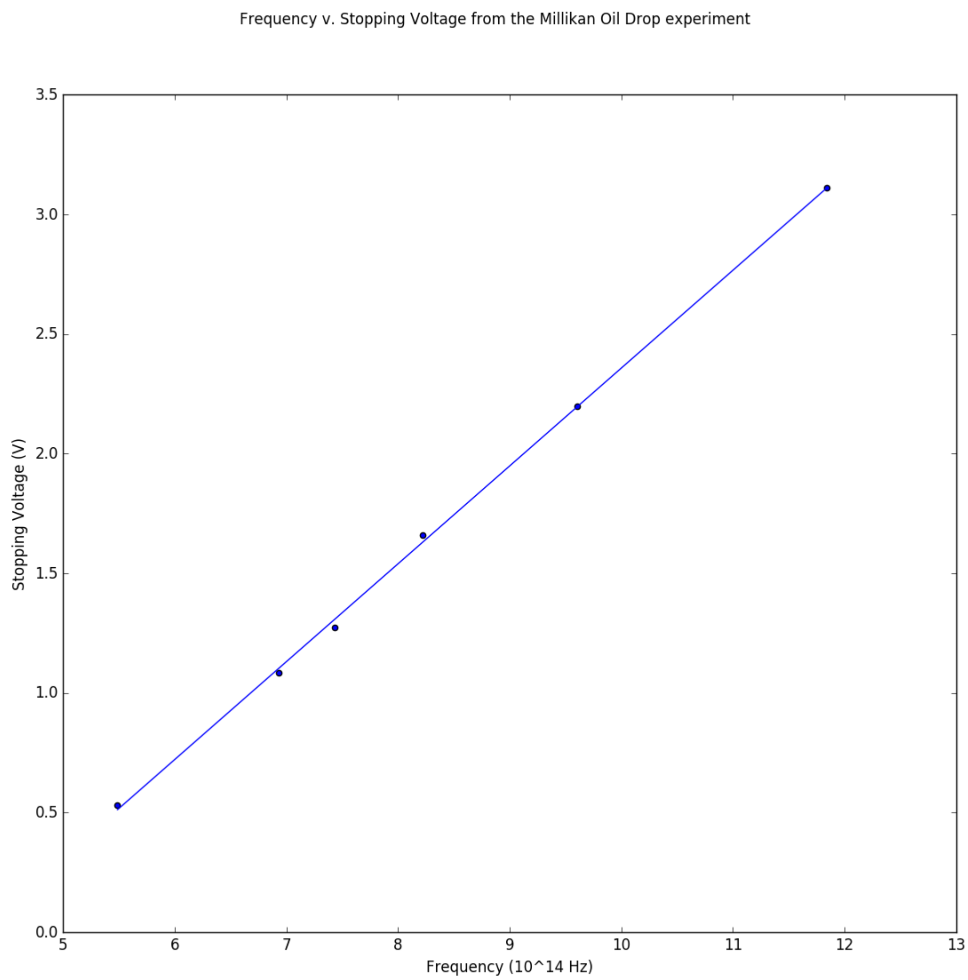

Figure 2: The data and best-fit line from the Millikan oil drop experiment. The slope of the best-fit line is $4.1 * 10^{-15}$ (V/Hz), with intercept -1.73 (V).