Project 2: Mandelbrot Fractal Border Length

Owen O'Reilly | 10/5/2024 | AMS-595

Summary

This script contains a procedure for approximating the length of the Mandelbrot fractal set boundary via 15th-order polynomial fitting.

Script 1: [Mandelbrot.m (via GitHub)]()

When run:

1. Generates 1000x1000 2D array of complex numbers ranging from -2 - *i* to 1 + *i*
2. Uses $f(z) = z^2 + c$ to determine whether every point converges or diverges after 100 iterations in *fractalIterationsToDivergence*, and creates a new grid of the same dimensions with values ranging from 1 to 100 based on that calculation for visualization
3. Uses a binary search-structured method *bisection* to scan each column of the complex number grid and find the divergence/convergence boundary, then stores the complex coefficient associated with the boundary in each column in a 1D array
4. Trims zero values off of the 1D array before fitting a 15th-order polynomial to the remaining points using *polyfit* native function
5. Displays a figure with the points of the fractal boundary as red x's and the fitted polynomial as a blue line (see Appendix)
6. Uses path length equation

$$l = \int_a^b \sqrt{(1 + (\frac{dx}{df})^2}$$

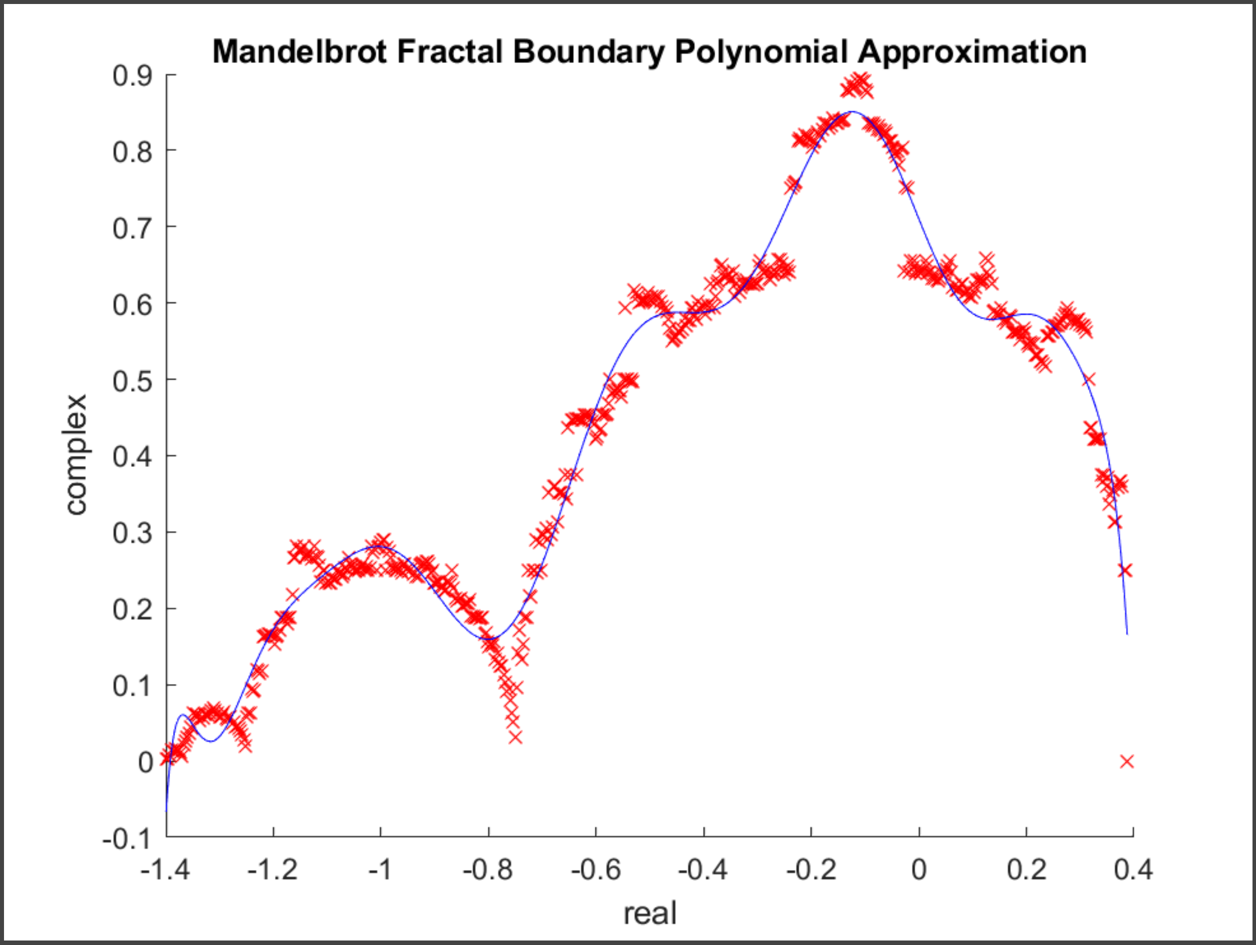
   to calculate the length of the fitted polynomial along the nonzero values using the *polyLen* custom function and prints the resulting length to the console

Discussion

The polynomial approximation of the Mandelbrot boundary captures the rough shape of the set, but by no means accurately captures the full shape of the fractal. As such, any length measurement based on this generated curve will be inaccurate to the actual fractal perimeter at the same resolution. The distinction of resolution is important, of course, because an infinite increase in the resolution would yield an infinite perimeter of the fractal. The accuracy of the polynomial representation could be improved by adding more orders to the polynomial fit, at the cost of computing time. Some small improvement might be obtained by increasing the number of points generated on the X and Y axes, but the polynomial already struggles to capture the curves at the current resolution, so there would need to be a large improvement to the polynomial accuracy before the resolution was increased.

Overall, this project is a good demonstration of the power of Matlab's native polynomial fitting function, as well as the elegant simplicity of the Mandelbrot fractal.

Appendix



Graph of the boundary points found for the Mandelbrot fractal, as well as the polynomial fit generated by Matlab. Note that although the polynomial captures the general shape of the fractal, several bumps in the curve are not accurately rendered in the line.