Project 4: Eclectic Mathematics

Owen O'Reilly | 11/2/2024 | AMS-595

Summary

There are three scripts contained in this assignment. The first is a simple visualization of a Mandelbrot set, the second is a demonstration of a Markov chain matrix, and the third allows for a Taylor Series approximation of any function.

Script 1: mandelbrot.py (on Github)

When run:

1. Generates a grid of complex numbers
2. Uses Mandelbrot test for convergence/divergence to create color values for each "coordinate" in the grid
3. Plots the resulting "image" of the fractal and saves it as a .png

Script 2: markov_chain.py (on Github)

When run:

1. Creates a 5x5 matrix of random numbers
2. Normalizes the rows of the matrix into *P*
3. Generates a random vector *p50*
4. Iteratively multiplies *p50* by P^T
5. Calculates eigenvalues of P^T
6. Prints *p50*, eigenvector corresponding to eigenvalue of 1, and summed difference between the eigenvector and *p50*

Script 3: taylor_series.py (on Github)

When run:

1. Generates 100th-degree Taylor approximation for function $x\sin^2(x) + \cos(x)$ and plots 100 points on interval [-1, 1] alongside reference points
2. Performs the same task for degrees 50 to 100 at 5 step intervals, measures the time required to do so, and outputs both the time and the error alongside the degree in a .csv file

Discussion

The Mandelbrot fractal produced by the program has good resolution and appears to be an accurate representation of the fractal shape itself. The program takes relatively little time to run.

The Markov chain demonstrates that the vector produced by the iterative matrix multiplication is very close to equivalent to the stationary distribution, with a summed difference on the order of 10^{-15}. The program is very quick to run.

The Taylor Series approximation is a very close fit at 100 degrees, as demonstrated by the generated plot. There is a significant drop in cumulative error as the degree magnitude increases from 50 to 100, from ~50 to 10^{-7}. This comes at a cost in terms of time to run, with an increase from 0.8s to 5s in a non-linear fashion.

All figures generated in these exercises can be viewed in the project Github.