Computational Physics Homework 2

Daniel Bateman

September 2024

1 Abstract

The purpose of this homework is to look deeper into how python works and explore some of its more complex functions.

2 Problem 1

In this problem we want to look at how NumPy takes a numerical input and represents it in a way allows it to perform computations, then compare NumPys representation to the actual value to gauge its' accuracy. NumPy uses 32 floating bit, where the first bit represents the sign, 0 for positive and 1 for negative, the next 8 bits represent the integer exponent, and the last 23 bits represent the mantissa. For the number 100.98763 NumPy gives:

$32bit : [0, 1, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1]$

Where,

$$Signbit : [0]$$

$$Exponent : [1, 0, 0, 0, 0, 1, 0, 1]$$

$$Mantissa : [1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1]$$

The actual difference in values comes out to be

$$2.7514648479609605e - 06$$

3 Problem 2

For this problem I initially started adding smaller and smaller numbers just by trial and error until I python stopped representing the added number however I discovered the "np.nextafter" and "np.inf" functions yielded much better results using ChatGPT. I found that the smallest number added to 1.0 in float32 to be

$$1.1920929e-07$$

and the smallest number added to 1.0 in float64 to be

$$2.220446049250313e-16$$

For part b of the question I resorted back to trial and error and it is evidently less elegant than the code written by ChatGPT.

minimum value represented by 32 floating bit:

$$1e-45$$

maximum value represented by 32 floating bit:

$$1e-45$$

minimum value represented by 32 floating bi:

$$1e-323$$

maximum value represented by 32 floating bit:

$$1e+308$$

4 Problem 3

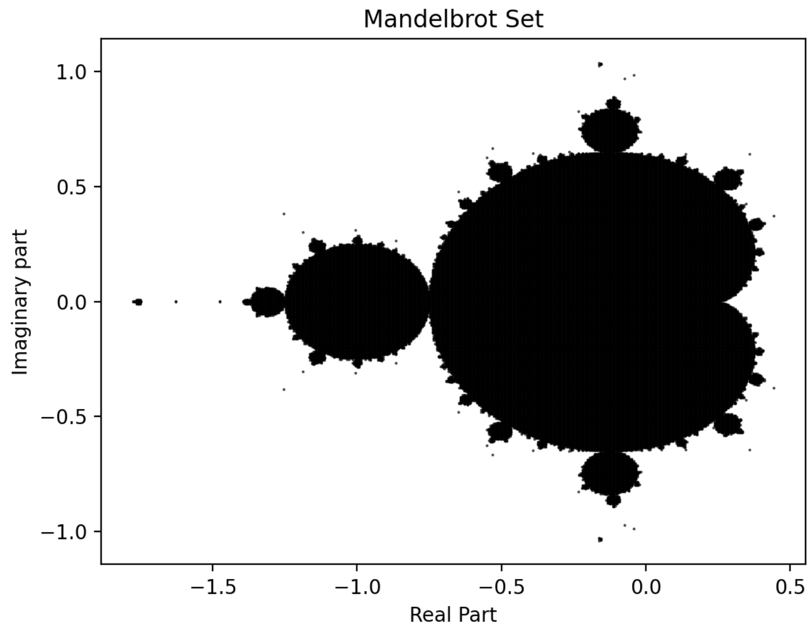
This problem demonstrates how different methods can be superior to others even if the answer is the same. Through calculating the Madelung constant firstly using a for loop and secondly using the meshgrid tool, we see that a for loop takes considerably longer. While both methods yielded the same result of , $M = -1.74182$, using the meshgrid tool was about 8 seconds faster on average than using the for loop method.

5 Problem 4

The Mandelbrot set is a fractal that exists sin the complex plane. The first step in this problem is discerning which values belong in the Mandelbrot set and which do not. We do this by iterating through the equation

$$z' = z^2 + c$$

with the condition that if $|z| > 2$ then the value does not belong to the set and a returned false. Once we know the values we can plot them and get the following fractal image.



6 Problem 5

In this problem we created a function that solves quadratics in 2 different ways, noticing that the different methods produce slightly different results. In this case the first method yields a slightly larger set of values. To fix this and to reduce error, we can express one solution as a function of the other. Calling a pytest yields no error showing the function works.