Computational Physics HW2

Siyuan Chen

September 26, 2023

P1: Numpy.float32 represents a real number with 32-digit of binary numbers. It approximates the number to be the form of $\pm(1+\sum_{i=0}^{22} m_i 2^{-(i-23)})2^{e-127}$ then represents it with the nearest mantissa m_i and exponent e from the format. In the 32-digit of binary numbers, the first digit on the left represents the positive or negative sign; the following 8 digit is the binary form of the exponent e; the last 23 digit are the mantissa ms. In this method, 100.98763 is represented by 01000010110010011111100110101011. The round-off error of such representation is about $2.7514648479609605 \times 10^{-6}$.

P2: From the equation given in Exercise 2.9, we can cancel the physical constants on both sides and get $M = \sum_{i,j,k=-L}^{L} (-1)^{i+j+k} \frac{1}{\sqrt{i^2+j^2+k^2}}$, not i=j=k=0. In the first method, we write a for loop to add the potentials from atoms at different positions on the grid. In the second method, we write coordinates of the grid and calculate potentials as an array-like object. Notice here we have to use floating point data type to record coordinates such that certain calculations could be allowed. For L = 100, the time to execute for loop is about 40s whereas for grid method, it's only 0.25s. Both methods agree the Madelung constant for sodium in sodium chloride to be -1.7418.

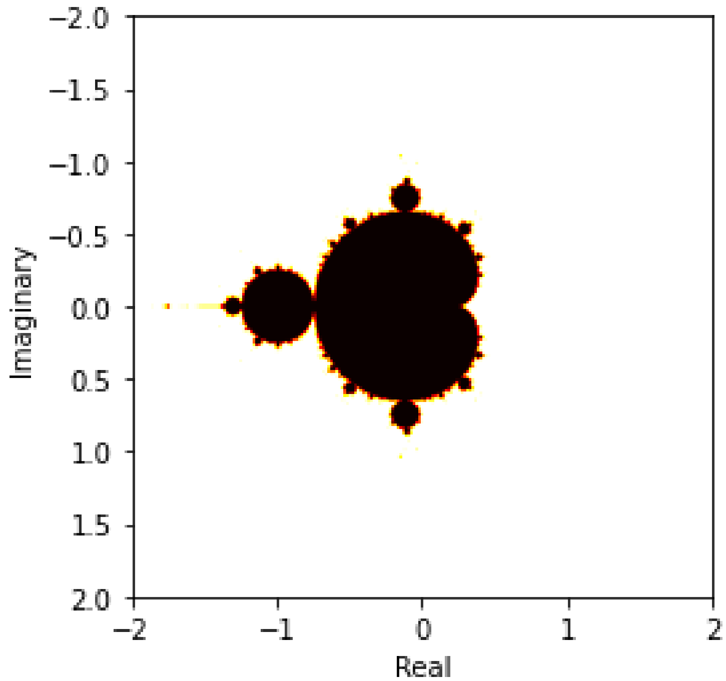

Figure 1: **P3: Mandelbrot**

P4: Part a gives solutions (-9.999894245993346e-07, -999999.999999). For part b,

$$x = \frac{-b\pm\sqrt{b^2-4ac}}{2a} = \frac{(-b\pm\sqrt{b^2-4ac})(-b\mp\sqrt{b^2-4ac})}{2a(-b\mp\sqrt{b^2-4ac})} = \frac{(-b)^2-(\sqrt{b^2-4ac})^2}{2a(-b\mp\sqrt{b^2-4ac})} = \frac{4ac}{2a(-b\mp\sqrt{b^2-4ac})} = \frac{2c}{-b\mp\sqrt{b^2-4ac}}.$$

In this form, we get the solutions to be (-1.000000000001e-06, -1000010.5755125057), which is a slight different from part a. This is because the difference of orders between a,c and b is of 10^6, then for $b^2 - 4ac$ this is of 10^{12}. This will cause round-off errors.