

PS-2 Solution

Nana Ama Darpaah

September 17, 2024

1 Question 1

Numpy uses the IEEE-754 standard to represent numbers in ones and zeros. Consisting of 4 bytes (32 bits), the decimal is converted to binary and divided into a total of 32 bits. The leading bit (bit number 31) represents the sign of number with 0 being positive and 1 being negative. The next 8 bits represent the exponent and the remaining 23 bits are the mantissa.

For the number 100.98763, its binary value is 01000010110010011111100110101011 and when converted back to decimal, is 100.98763275146484. The difference between the true value and the converted value is 0.000002751465.

2 Question 2

The minimum and maximum positive numbers that the 32-bit and 64-bit representation without an underflow or overflow are

- 32-bit : - Minimum: 1.1754943508222875e-38 - Maximum: 3.4028234663852886e+38
- 64-bit: - Minimum: 2.2250738585072014e-308 - Maximum: 1.7976931348623157e+308

3 Question 3

The Madelung constant (M) is used to determine the total electrostatic potential of an atom in a lattice by treating all other ions as point charges. At the origin (0,0,0), the potential felt is

$$V(i, j, k) = \pm e / (4\pi\epsilon_0^2 \sqrt{i^2 + j^2 + k^2}) \quad (1)$$

The potential can then be approximated as

$$V_i = e / (4\pi\epsilon_0 r_0) \times M_i \quad (2)$$

The Madelung constant obtained for L = 100 atoms of NaCl was determined to be -1.7418198158396005 , where the negative sign indicates stability of the lattice.

4 Question 5

For the equation of the line

$$ax^2 + bx + c = 0 \quad (3)$$

Its roots, the solution to this equation, are obtained by the well known

$$x = (-b \pm \sqrt{b^2 - 4ac})/2a \quad (4)$$

and multiplying both numerator and denominator by $-b \mp \sqrt{b^2 - 4ac}$, we get another formula

$$x = 2c/(-b \mp \sqrt{b^2 - 4ac}) \quad (5)$$

From the first equation, the values for x obtained were $-9.999894245993346e-07$ and -999999.999999 . However, solving the second equation leads to roots with values $-1.0000000000001e-06$ and -1000010.5755125057 , which are not equal.

This could be attributed to the error produced due to the numpy bit representation and those round off errors which come about after arithmetic operations were applied.