Assignment-3

PHY473/473A-Computational Physics

22nd Jan, 2023

Explain the algorithm you are using for each question. Use for loop and function to make your code efficient.

Question 1. Write a python program that computes the sum of all the elements of a matric C, where C = AB. Here A and B are two matrices of size $N \times N$. The value of N should be taken as input from the keyboard. The elements of the matrices are defined as:

$$A[i, j] = i \star j/100$$

$$B[i, j] = (i + 1) \star (j + 1)/100.$$

Round your answer up to three decimal places using np.round(x,3) function of python.

Question 2. The **Sieve of Eratosthenes** is an algorithm for finding all prime numbers less than or equal to a number N . Read about this algorithm on Wikipedia and implement it in a Python program.

Question 3. **Compute velocity and acceleration from 1D position data**
Suppose we have recorded GPS coordinates $x_0, ..., x_n$ at times $t_0, ..., t_n$ while running or driving along a straight road. We want to compute the velocity v_i and acceleration a_i from these position coordinates. Using finite difference approximations, one can establish the formulas

$$v_i \approx \frac{x_{i+1} - x_{i-1}}{t_{i+1} - t_{i-1}},$$

$$a_i \approx 2(t_{i+1} - t_{i-1})^{-1} \left(\frac{x_{i+1} - x_i}{t_{i+1} - t_i} - \frac{x_i - x_{i-1}}{t_i - t_{i-1}} \right),$$

for $i = 1, ..., n - 1$ (v_i and a_i correspond to the velocity and acceleration at point x_i at time t_i, respectively)

a) Write a Python function kinematics(i, x, t) for computing v_i and a_i, given the arrays x and t of position and time coordinates ($x_0, ..., x_n$ and t_0, ..., t_n).

b) Write a Python function test_kinematics() for testing the implementation in the case of constant velocity V. Set $t_0 = 0, t_1 = 0.5, t_2 = 1.5$ and $t_3 = 2.2$, and $x_i = V t_i$. Call the kinematics function for the legal i values.

Question 4. Find roots on an interval with the bisection method for $f(x) = x^2 - 4x + e^{-x}$. Use python's lambda function to create the function $f(x)$.

Question 5. a. Write a function count_pairs(dna, pair) that returns the number of occurrences of a pair of characters(pair) in a DNA string (dna). For example, calling the function with dna as 'ACTGCTATCCATT' and pair as 'AT' will return 2.

b. Count how many times a certain string appears in another string. For example, the function returns 3 when called with the DNA string ‘ACGTTACGGAACG’ and the substring ‘ACG’.

Hint: For each match of the first character of the substring in the mainstring,check if the next n characters in the main string matches the substring, where n is the length of the substring. Use slices like $s[3:9]$ to pick out a substring of s.