

Lab Assignment-3

PHY617/473-Computational Physics

23rd Jan, 2025

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 matrix 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 and take $N = 100$. The elements of the matrices are defined as:

$$A[i, j] = i * j / 100$$

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

Do the matrix multiplications using (a) the for loop and (b) using `numpy.dot` functions. [**4+4 marks**]

Round up your answer up to three decimal places using `numpy.round(x,3)` function of python for both cases. Which method [(a) or (b)] is taking less time and why? [**2 marks**]

Question 2. Compute velocity and acceleration from 1D position data

Suppose we have recorded GPS coordinates x_0, \dots, x_n at times t_0, \dots, 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 $i = 1, \dots, n-1$ (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, \dots, x_n and t_0, \dots, t_n). [**5 marks**]

b) Write a Python function `test_kinematics()` for testing the implementation in the case of constant velocity V . Take the dataset for position and time from the given datafile `GPS_data.txt` using `numpy.loadtxt()`. Call the `kinematics` function for the legal i values and calculate v_i and a_i . [**5 marks**]

Question 3. 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 'ACT-GCTATCCATT' and `pair` as 'AT' will return 2. [**5 marks**]

b. Count how many times a certain string appears in another string. For example, the function returns 3 when called with the DNA string 'ATCTTATCGAATC' and the substring 'ATC'. [**5 marks**]

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`.

Question 4. Look at the datafile *planet_data.txt* which has information about orbital period, semi major axis, radius and mass of a planet which is orbiting a star.

(a) Read them using numpy loadtxt and verify Kepler's third law ($T^2 \propto a^3$). [6 marks]

(b) Also plot mass and radius of the planet using python plot function. [4 marks]