Midterm Exam (part 1) - Computational Physics I

NAME: Alan Palma Tavez **SCORE:** 7/8

Date: Friday 18 October 2024 **Duration:** 45 minutes

Credits: 8 points (4 questions) **Type of evaluation:** LAB

Part 1 is closed-book, in-class, and contains short-answer questions. Please provide concise answers to the following items:

1. **(2 points) Programming Languages**
   a. Explain the difference between low-level and high-level programming languages.
   b. Provide 1 example of a low-level programming language and 1 of a high-level language.

a) • The low level programming language is more closer to the hardware world (machine code) while the high programming is closer to the user language. ✓

-0.25

• High-level programming language only needs an interpreter ~~compiler~~ to work while lowe level language need a compiler ~~interpreter~~. However, the high level language is slower than the high level langule since the last one is more closer to the machine language. ✓

• Most of programmers use hingh level langule because is more faster and easier to write whil low level language is used for more specific jobs. ✓

b) Lower level: Fortran / C ✓

High level: Python / Mathematica ✓

2. **(2 points) Systems of linear equations**
   a. Explain how the Gauss elimination method for solving systems of linear equations work.
   b. List the main steps for solving such systems via symbolic algebra with SymPy in python.

a) The Gauss elimination method consist on diagonalize our matrix of elements related with the system linear equations, this via file/column operations in the expanded matrix. Finally make a backward sustitution to get the x vector solution. ✓

Ex in 3x3:

$$Ax = b \Rightarrow A = \left(\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array}\right) \checkmark \text{ we should get } \left(\begin{array}{ccc|c} c_1 & c_2 & c_3 & b_{31} \\ 0 & c_4 & c_5 & b_{32} \\ 0 & 0 & c_6 & b_{33} \end{array}\right) \checkmark$$

b) 1) Import third party libliarie SymPy ✓

2) Declare as symbols our variables $x = (x_1, x_2, x_3)$ ✓ (Important).

3) Carry out the respective operation to solve the system. ✓

I don't remember but I think that there is some specific dependency to solve it. ✓

3. **(2 points) Interpolation Methods**
   a. When do we use data interpolation methods?
   b. Name 2 types of interpolation methods in python.

a) - We use data interpolation in specific cases as we don't want a project from the data or the data does not represent a physical behaviour to be modeled. ?
- "Join" points in interpolation would work to only see a trend graficolly on a specific data.
- when we don't have a physical motivated model.

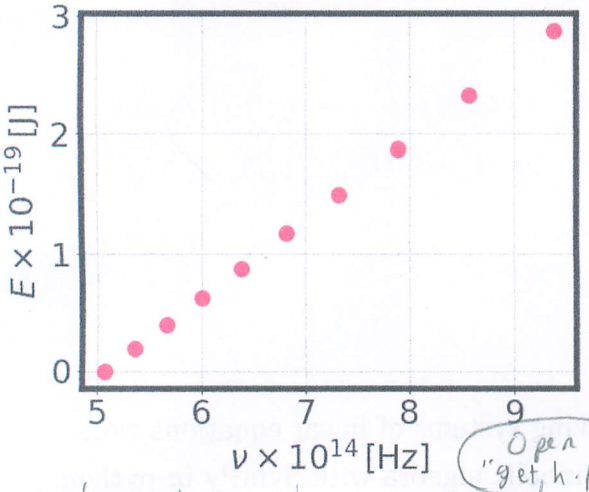
-0.5

But when do we use interpolation?
- Filling missing data
- Resampling data.

b) 1. Lagrange interpolation ✓

2. Using Scipy library, where we can setup the interpolation "degree", I mean linear, quadratic, etc. ✓

4. **(2 points) Regression algorithm**



Imagine you are given the energy-frequency data shown on the left figure from a photoelectric effect experiment using Caesium (Cs), and you are asked to carry out a regression to estimate the Planck constant. Design and sketch a suitable algorithm workflow to achieve this goal in python.

Photoelectric effect:
$E_\gamma = \phi + K_e$ ✓
$hf = \phi + K_e$ ✓
$\Rightarrow K_e = hf - \phi$ ✓
$E = hf - \phi$ ✓
Physical motivated model. ✓

Open "geth.py" ✓
Third party libraries ✓
Inspect the data ✓ (Through a fast plot should be.)
data ← Put the data into a pandas object ✓
Treat data and put in python objects:
f ← frequency ✓
E ← energy ✓
Create a function according to the physical motivated model: (E = hf - ϕ) → model() ✓
Carry out a regression with two free parameters using model().fit and E with scipy.optimize ✓
The model fit and the data should be presented in a plot ✓
One of the two free parameters should be the planck constant value (the slope). ✓
Present the result with its respective uncertainty) ✓
Close "geth.py" ✓
What about the goodness-of-fit test to assess the regression?

IO?

-0.25