Third exercise class

Class 5

Introduction to numerical programming and analysis

Asker Nygaard Christensen

Spring 2021

Plan

1. My thoughts on problem sets

2. Problem set 1

3. Recap and poll

My thoughts on problem sets

My thoughts on problem sets

They are located here (You can fork them, like I showed you in the last exercise class, or download them directly)

- A point that bears repeating: You're not required to start from scratch at every problem, finding something similar from the lecture notes or even the internet and then rewriting code to apply to the specific problem is encouraged
- Try and understand the harder problems conceptually first: what am I'm trying to do, which steps does this require, then you can start writing up the code to solve it
- It's fine to look at the solutions, but if you do, make sure you understand each line and how it all comes together
- The internet (and me) is your friend, when getting weird errors
- Also: experimentation is key, running something and getting an error allows you to learn something

Problem set 1

Projected time plan:

- 15:20-15:35: I'll introduce you to problems 2.1-2.4
- 15:35-16:00: You'll do problems 2.1-2.4 yourself
- 16:00-16:15: Break
- 16:15-16:20: You'll do problems 2.1-2.4 yourself
- 16:20-16:30: Introduction to problem 3
- 16:30:16:55: You'll do problem 3 yourself
- 16:55-17:00: Recap and poll about class structure

My notes on problem set 1

You should note that this problem set requires stuff you learned in lecture 3 and not Monday's lecture.

Notes on specific problems:

- 2.2 print: f-strings or formatted string literals is an excellent tool for printing your results. The basic syntax for printing a float is: f'{float:*width.digits*f}', where *width* is total width and *digits* is digits after the decimal point, but other letters than f after *digits*, leads to different options. A general guide can be found here (This also introduces other possible formatting options, but I recommend sticking to f-strings, so you can skip to that part) and a great table with number formatting options can be found here
- 2.3 matplotlib: Use point 4.4 in lecture note 3. There is no need to memorise all the plotting code. Also if you write *fig.tight_layout()* at the end it will look a little prettier

More notes on problem set 1

- 2.4: Numpy has the sine function. If you can't remember the syntax for using scipy optimize, you can can have a look at point 7 in lecture note 3 or even look at the documentation
- 3: It might look scary, but in essence it just a multi-dimensional version of the previous problem. The simple loop Jeppe has prepared is kind of slow and imprecise, beyond just using itertools as he suggest, a small amount of economic intuition can increase the speed significantly, I'll show you at the end of the class

If you find that you're having trouble understanding any of the problems conceptually (meaning that you can solve it, but you're not quite sure how you did so) write me, and I can explain it to you

Recap and poll

Recap and pol

An opinion poll:

I want to know your opinion on the classes via this poll.

Programmers looking at programming memes
Ah, humor based on my pain.
Ah, ha, ha.

You to-do

The deadline for the inaugural problem is 19th of March. So your priorities should be:

1. Finish the DataCamp courses (The deadline is this Sunday!)
2. Organise your group and register in group excel file (In MS Teams: UCPH_Lectures - Introduction to Programming → 'General'-channel → Files → Groups.xlsx).
3. Make sure you understand what we've covered today in PS1.
4. Prepare yourself for next exercise class by reading PS2. PS1 and PS2 contains all the tools you need for project.
5. Possibly review lecture 3 and 4, they cover the topics you need in order to solve the project.
6. If you have time, you can check out last year's inaugural project, it is very similar to this years. There is no official solution guide, but you can see the repositories of former groups. This group had a particularly elegant solution.