UNIVERSITY OF COPENHAGEN

Outroduction

Introduction to Programming and Numerical Analysis

Asker N. Christensen

Spring 2024

Table of contents

1. Introduction

2. Course overview

3. Exam project

4. Wrap up

5. 2023 exam examples

Introduction

Introduction

- **We are almost at the end!**
- We hope that you have enjoyed the course, and will use programming in your further economics education/work.
- **The final exam** is a portfolio exam
  - The 3 projects you have created during the course.
  - A new exam assignment, you'll have 48 hours to answer.

Course overview

Course overview

- **Fundamentals:** Print, plot, optimize, simulate, structure, document, work-flow.
- **Working with data:** Fetch, combine, split-apply-combine, visualize.
- **Algorithms:** Pseudo code, algorithms, complexity, solve, optimization, symbolic.
- **Further perspectives:** Comprehensions, generators, vectorization, parallelization, timing, numba.

While working with the material, you should hopefully also have gained the ability to **reformulate mathematical models into code.**

All model notebooks (ASAD, Labor supply, Consumer problem, Production economy, Random numbers example, OLG, Ramsey) are relevant to revisit.

Exam project

Problems

- **Structure:** 3 problems with 3-6 sub-questions. Most problems are on solving and simulating models and analyzing their implications graphically and numerically. There can also be problems on working with data, and on algorithms.
- **Examples of a model problems**
  1. Solve consumer or firm problems (with non-standard constraints)
  2. Solve and simulate an AS-AD model
  3. Solve for the Walras-equilibrium in an exchange economy
  4. Solve an extended Solow model
  5. Solve a two period dynamic optimization problem

  $\Rightarrow$ similar to the problems in the problem sets
- **Curriculum:** Lecture notebooks (except notebooks and sections marked with $+$)
- **Packages:** No new packages are required, and using non-standard packages are actively discouraged.

Answering

1. **Focus on answering the questions** - nothing more, nothing less
2. Explain your **method in words** (or with an algorithm)
3. **Structure and comment your code!**
4. Explain your **results in words**
5. **Partial answers, attempts and considerations** are also **awarded** (something on everything is better than a lot on a few questions)
   - If you think there is an error in your code, but you can't/don't have time to fix it, addressing this in words is a **huge** plus.

Disclaimer: Solving the full exam project in depth will be hard.

Use of AI is allowed. Guide to integrate with VS Code in the bottom of this page.

Hand-in

- **You should hand-in a single zip-file named with your groupname only.**
- The zip-file should contain:
  1. A general README.md for your portfolio
  2. Your inaugural project (in the folder /inauguralproject)
  3. Your data analysis project (in the folder /dataproject)
  4. Your model analysis project (in the folder /modelproject)
  5. Your exam project (in the folder /examproject)

Wrap up

To-do before the exam

(0.) Please answer the course evaluation you have gotten over mail.

1. Model project **peer feedback** before **May 19**.
2. **Polish up** your 3 projects so they are ready for the exam.
3. **Exam problems preparation.**
   Many ways to do that:
   - Look at earlier exams to get a sense of the kind of questions you'll get
   - Get an overview of the contents of the lecture and exercise notebooks. → Fill knowledge gaps.
   - Go through the lectures that cover solving economic models
     (ASAD, Labor supply, Consumer problem, Production economy, Random numbers example, OLG, Ramsey).
     Make sure you understand them, for example by changing them or rewriting the code to solve them yourself.
   - Resolve the exercises.
   - Try to solve earlier exams.

Questions

- **Any questions now?**
- **Online:** https://github.com/NumEconCopenhagen/IntroProg-lectures/issues

Amped for more programming? KU offers more programming courses: https://sites.google.com/view/numeconcph/home

2023 exam examples