

Lecture 01

Introduction to Programming and Numerical Analysis

Lecturers:

Christian Langholz Carstensen

Jeppe Druedahl



UNIVERSITY OF
COPENHAGEN

Plan for today

1. Online lectures and exercise classes
2. Intended learning outcomes
3. Numerical analysis in action
4. Your lecturers and you
5. Infrastructure
6. Work-flow and online learning
7. Projects and group work
8. A first example
9. Summing up

The online lectures and exercise classes

How we'll manage the lectures:

- **Today:** 2 hours. Some lectures will be **3 hours**.
- We are now officially back to normal, as you know.
- .. unless you have covid, of course. And many of you will probably get it.
- All classes are therefore streamed.
- They will also be recorded, but only released a week later.
- The institute's policy is that we prioritize the physically present aspect of teaching.

During the lectures:

- Ask a question by raising your hand if you are in the auditorium.
- If you want to ask a question from home, write a comment in the meeting feed, and tag me.
- If I seem to not notice, try and interrupt me.

Intended learning outcomes I

Get the fundamentals of programming right

- Be comfortable with working in a serious programming language, in our case Python.
- Understand some of the fundamental concepts that shapes Python and CS in general.

Apply numerical analysis to economic problems

- Use your computer skills to improve your understanding of well known economic models.
- Visualize solutions and simulations.
- Go further: solve models that cannot be solved analytically.
 - Empirically relevant models are often like that..
 - Models may include uncertainty, non-convexities and non-linear feedback mechanisms.
- Fetch data online and do programming based data manipulation and statistics (Excel is the mother of bad habits).

Intended learning outcomes II

Your new skills can be used when:

- Writing your bachelor or master thesis.
- Writing seminar projects.
- Working on assignments in other courses, including those in econometrics.
- At your job.
- Working on IT projects in general.

We *are* focusing on methods rather than theory, but:

- This does not mean that economic theory and math are irrelevant!
- In fact, it is super important to understand the math behind your methods.
- Otherwise, you'll be in the dark holding a black box.

Scientific programming

The three steps of numerical analysis

1. Mathematical model → create an algorithm
2. Algorithm → write the code
3. Code → present your results

Bugs are everywhere!

Is it the model itself? Or the algo? The code, perhaps? Or maybe in the presentation?

To figure out what went wrong, you need a **solid work-flow**:

1. Clear and understandable code *(so that you won't confuse yourself and others)*.
2. Comprehensive testing *(so that you can catch bugs in the making)*.
3. Documentation *(so that you won't forget and others will understand)*.

Point 2 and 3, not just 1, are **crucial** elements of creating useful software.

As is working actively with others and your future self - thus also an integral part of this course.

Active learning

We focus on active learning

- If you are not sweating, you are doing it wrong!
- It takes a lot of exercise and repetition to get *comfortable* with programming. Our brains are not designed to remember this stuff.
- We will show you lots of examples along the way.
- We will be helpful with answering all of your questions.
- You will also get the chance to help each other out.
- But most importantly, you need to spend time at the keyboard.

It's hard work, but

- The rewards are also **high**.
- You won't find such a course at most econ bachelor programs.

It is really pioneering work by Jeppe.

Please !!

You all have your laptops open, so:

No SoMe-time or news feeding during lectures.

You'll disturb your classmates sitting behind you!

Also:

You may be inclined to look up **relevant stuff** during lectures.

That too is annoying for others if there are attention grabbing ads in your browser

About your lecturers

Full name: Christian Langholz Carstensen

Position: Postdoc at Centre for Computational Economics

Contact: ch.carstensen@econ.ku.dk

Research interests:

- Dynamic models of urban economics (Phd) and education choice (Postdoc)
- Structural estimation
- Agent-based models

Full name: Jeppe Druedahl

Position: Assistant Professor at Department of Economics, Center for Economic Behavior and Inequality (CEBI)

Website: www.econ.ku.dk/druedahl

Research interests: Macro-questions and micro data. Numerical methods.

Numerical analysis in action

- We are working in **Python 3.9**
- **Suggested environment:**
 - The **Anaconda distribution** (which you can stick to)
 - Notebooks in **JupyterLab**
 - Editor for code is **Visual Studio Code**
- **Today:**
 - I'll give you a quick run through JupyterLab.
 - Solve a consumer problem.

Infrastructure I

Web-pages: The course is organized around our public website
www.numecon copenhagen.netlify.com

And the **GitHub** site
<https://github.com/NumEconCopenhagen>
where you can get all course material.

Links to **exercise class meetings** are available on Absalon.
Also, videos viewed during lectures will also be on Absalon.

Infrastructure II

About Git

- **Git** is a version-control system that virtually everyone in CS and scientific programming uses.
- It allows programmers to coordinate their work across computers without messing up.
- **GitHub** is the main online platform for sharing work through **Git**.
- You can get the latest version of all material by pulling it from our Github **repositories**.
- It is not easy to wrap your head around Git.
- Git is integrated in JupyterLab and VSCode, which makes it easier to start on.

You and Git

- You'll work with Git on **DataCamp** and during exercise classes.
- On DataCamp, you will use the command line.
- Looks scary at first, but is **easier** than point and click.

Infrastructure III

Download course material

1. Follow the [installation guide](#)
2. If you have not yet restarted JupyterLab after installations, do it now.
3. Make sure you are in the folder, where the course content should be located.
4. Press the tab **Git**.
5. Press `Clone a repository`
6. Paste in `https://github.com/NumEconCopenhagen/lectures-2022` and enter.
7. You now have all course material in the folder `lectures-2022`
8. Do the same thing for exercise class material, using the url `https://github.com/NumEconCopenhagen/exercises-2022`

Infrastructure IV

Updating your folder

1. Press the tab **Git**.
2. Press `Pull from remote`
3. If you get an error message, you need to discard all your changes. See guide on website.

Getting started

- **DataCamp**
 - You now have 6 months full access. Follow link in welcome mail or Absalon.
Go nuts!
 - DataCamp requires no installation.
- **Setting up your environment**
 - [Installing Python and VSCode](#)
 - [Running Python in JupyterLab](#)
 - [Running Python in VSCode](#)

Lectures and classes:

- Lectures: Monday 15-17(18)
- Classes: Tuesday/Wednesday/Thursday 15-17 (*Also in Easter week*)
- **Group work:** Strongly encouraged! All projects can be done in fixed groups (maximum of 3). You do not have to be in the same class.
- Classes need to be similar in size. Write me if you want to switch class. No promises!
- **Exam requirements:**
 1. Completing DataCamp courses (assignments on DataCamp.com)
 2. Inaugural project
 3. 2x Useful peer feedback on inaugural projects
 4. Data analysis project
 5. 2x Useful peer feedback on data analysis projects
 6. Model analysis project
- **Exam:** May 25th. Portfolio of projects + exam problem (48 hours). **Pass or fail.**

Lecture plan, tentative

1. Introduction
2. Fundamentals: Primitives
3. Fundamentals: Optimize, print and plot
4. Fundamentals: Random numbers and simulation
5. Fundamentals: Workflow and debugging
6. Fundamentals: Recap and overview
7. Working with Data: Load/save and structure data
8. Working with Data: Basic data analysis
9. Algorithms: Searching and sorting
10. Algorithms: Solving equations
11. Algorithms: Numerical optimization
12. Further perspectives: Canonical Economic Models
13. Further perspectives: Agent-based Models
14. Further Perspectives: The need for speed

Exercise plan

1. DataCamp
2. DataCamp
3. DataCamp
4. Problem Set 1: Solving the consumer problem
5. Problem Set 2: Finding the Walras equilibrium in a multi-agent economy
6. Work on your inaugural project
7. Problem Set 3: Loading and combining data from Denmark Statistics
8. Problem Set 4: Analyzing data
9. Work on your data project
10. Problem Set 5: Writing your own searching and sorting algorithms
11. Problem Set 6: Solving the Solow model
12. Problem Set 7: Solving the consumer problem with income risk
13. Work on your model analysis project
14. Work on your model analysis project
15. Feedback on model project

Work flow I

Lectures

1. Look through the notebook before the lecture starts.
2. Come in if you can and sit with your group.
3. Bring your laptop.
4. We'll have quizzes during lectures. Some of them may be more extensive.
5. Lectures are streamed over Teams.
6. You can make a channel for your own group, so that you can easily work together, even if some are only present online.

Work flow II

Exercise classes: Work on problem sets and your projects.

- **Solutions** are hidden but available from the outset - it's ok to peak when you're stuck. But write solution out yourself afterwards.
- **Organization:**
 - We meet physically, but use MS Teams for the online organization.
 - Each class has a team.
 - When you have formed a group, you'll get your own permanent channel in the team.
 - The feed of the class channel is projected during classes.
 - When you want to ask a question, just write a comment in the feed in order to get in line. Also works for people sitting at home.
 - You can thereby ask, and help, each other. And post tips + general questions.
- **Between class and lecture:**
 - Brush up the lecture notebook.
 - Finish problem set and make experiments.

Getting help

- **Observation:** Programming is the slow and painful removal of tiny errors in your code – one at a time.
- **Everybody often forgets the correct syntax** \Rightarrow trial-and-error and testing is central, never a single correct approach.
- **Ask questions!** In the following order:
 - Look in the documentation
 - Talk about it in your group
 - Search Google + Stackoverflow, see this [guide](#).
 - Ask question online using GitHub issues, see this [guide](#), [examples](#)
- **Help each other!** You will learn a lot. Remember to be constructive and polite!

Inaugral project

- **Objectives:**
 - Apply simple numerical solution methods
 - Structure a code project
 - Document code
 - Present results
 - Use GitHub
- **Content:**
 - Solution of pre-specified economic model
 - Visualization of solution
- **Structure:**
 - A self-contained single notebook presenting the analysis
 - Fully documented python files
- **Hand-in:** Create and commit folder called "inauguralproject" in your GitHub repository

Data analysis project

- **Objectives:**

- Apply data cleaning and data structuring methods
- Apply data analysis methods
- Structure a code project
- Document code
- Present results in text form and in figures

- **Content:**

- Import data from an online source
- Present the data visually (and perhaps interactively)
- Apply some method(s) from descriptive economics
(»samfundsbeskrivelse«)

- **Structure:**

- A self-contained single notebook presenting the analysis 2. Fully documented python files

- **Hand-in:** Create and commit folder called "dataproyekt" in your GitHub repository

Model analysis project

- **Objectives:**

- Apply model analysis methods
- Structure a code project
- Document code
- Present results in text form and in figures

- **Content:**

- Describe an algorithm on how to solve a simple economic model
- Solve (and perhaps simulate) a simple economic model
- Visualize results across e.g. parametrizations
- Analyze one or more extensions of the baseline model

- **Structure:**

- A self-contained single notebook presenting the analysis 2. Fully documented python files

- **Hand-in:** Create and commit folder called "modelproject" in your GitHub repository

Let's see JupyterLab and VSCode in action!

Your to-do list

1. **First priority:** Find a group! If you are looking for one, you can register on Absalon.
2. **Second priority:** Login to DataCamp (see info in e-mail)
3. **Third priority:**
 - A. Install Anaconda. See the guide: [Installing Python and VSCode](#)
 - B. Open JupyterLab and experiment. See the guide: [Running Python in JupyterLab](#)
4. **Fourth priority:** Download slides and exercises with Git [Downloading material using Git](#)
5. **Fifth priority:** Read the guides on:
 - A. [Searching for answers using Google and Stackoverflow](#)
 - B. [Asking questions using GitHub issues](#)
6. **Next time:** Introduction to the fundamentals of Python