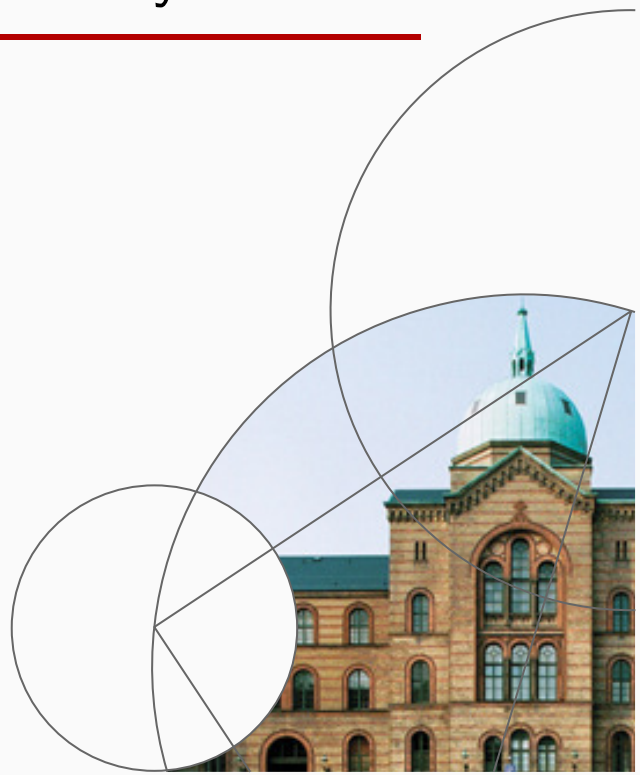
CENTER FOR
ECONOMIC
BEHAVIOR &
INEQUALITY

1. Introduction

Introduction to Programming and Numerical Analysis

Jeppe Druedahl

Spring 2019



Plan

1. Intended learning goals
2. Numerical analysis in action
3. Infrastructure
4. Work-flow
5. Projects
6. More examples
7. NumEcon
8. Summing Up

Intended learning goals

Intended learning goals

- **In a nutshell:** *Learn how to use numerical analysis to improve your understanding of economic problems*
  1. Visualize solutions and simulations of well-known models
  2. Explore alternative assumptions regarding functional forms and calibrations of parameters
  3. Solve more realistic models with constraints, uncertainty and non-convexities, where algebraic solutions are not available
  4. Work with online data and do programming based statistics and descriptive economics

Intended learning goals

- **In a nutshell:** *Learn how to use numerical analysis to improve your understanding of economic problems*
  1. Visualize solutions and simulations of well-known models
  2. Explore alternative assumptions regarding functional forms and calibrations of parameters
  3. Solve more realistic models with constraints, uncertainty and non-convexities, where algebraic solutions are not available
  4. Work with online data and do programming based statistics and descriptive economics
- Focus will be on **methods** rather than **economics**
  $\Rightarrow$ very relevant for applying in bachelor and master theses

Intended learning goals

- **In a nutshell:** *Learn how to use numerical analysis to improve your understanding of economic problems*
  1. Visualize solutions and simulations of well-known models
  2. Explore alternative assumptions regarding functional forms and calibrations of parameters
  3. Solve more realistic models with constraints, uncertainty and non-convexities, where algebraic solutions are not available
  4. Work with online data and do programming based statistics and descriptive economics
- Focus will be on **methods** rather than **economics**
  $\Rightarrow$ very relevant for applying in bachelor and master theses
- You will learn a **set of important tools**, but it is equally important that you **learn how to acquire new tools** for problems you will face in the future (in your studies or work-life)

Scientific programming

- **Numerical analysis** is a ***complement*** not a substitute for **math**

Scientific programming

- **Numerical analysis** is a ***complement*** not a substitute for **math**
- **Three central steps:**
  1. mathematical problem $\rightarrow$ construct algorithm
  2. algorithm $\rightarrow$ write code
  3. write code $\rightarrow$ present results

Scientific programming

- **Numerical analysis** is a ***complement*** not a substitute for **math**
- **Three central steps:**
  1. mathematical problem $\rightarrow$ construct algorithm
  2. algorithm $\rightarrow$ write code
  3. write code $\rightarrow$ present results
- **The set of potential errors is infinite:**
  A good work-flow is very important
  1. Clear structure reduces the number of bugs
  2. Testing helps discovering bugs
  3. Documentation helps removing bugs

Scientific programming

- **Numerical analysis** is a ***complement*** not a substitute for **math**
- **Three central steps:**
  1. mathematical problem $\rightarrow$ construct algorithm
  2. algorithm $\rightarrow$ write code
  3. write code $\rightarrow$ present results
- **The set of potential errors is infinite:**
  A good work-flow is very important
  1. Clear structure reduces the number of bugs
  2. Testing helps discovering bugs
  3. Documentation helps removing bugs
- **Programming is more than writing code:** Structuring, testing, documenting and collaborating on code is a central aspect of this course

Active learning

- **Active learning:** To learn scientific programming you need to work on actual problems yourself
  - I can show you examples
  - I can guide you in terms of where to start
  - I can answer questions
  - But the lectures are probably the least important part of this course

Active learning

- **Active learning:** To learn scientific programming you need to work on actual problems yourself
  - I can show you examples
  - I can guide you in terms of where to start
  - I can answer questions
  - But the lectures are probably the least important part of this course
- **High level:** Very few (if any) econ bachelor programs provide education on numerical analysis on the level you will get

Active learning

- **Active learning:** To learn scientific programming you need to work on actual problems yourself
  - I can show you examples
  - I can guide you in terms of where to start
  - I can answer questions
  - But the lectures are probably the least important part of this course
- **High level:** Very few (if any) econ bachelor programs provide education on numerical analysis on the level you will get
- **First generation:** All of your feedback is very important for optimizing and improving the course!

Who I am

- **Name:** Jeppe Druedahl (ph.d. polit)
- **Web-page:** www.econ.ku.dk/druedahl
- **Position:** Assistant Professor at Department of Economics, Center for Economic Behavior and Inequality (CEBI)
- **Research interests:**
  1. Macro-questions
  2. Micro-data
  3. Numerical methods
- **Modern macro-models**
  1. Heterogenous agents (households and firms) take decisions under uncertainty and imperfect information
  2. Markets are not complete
  3. The equilibrium is found approximately on a (large) computer

Who you are

[results from questionnaire]

Numerical analysis in action

Numerical analysis in action

- We work with **Python 3.7**
- **Suggested environment:**
  1. **Distribution:** Anaconda
  2. **Documents:** JupyterLab
  3. **Editor/IDE**: VSCode
- **I will show** how to
  1. Run Python in JupyterLab
  2. Solve the consumer problem from microeconomics

Infrastructure

Getting started

- **Web-page:** The course is organized around www.numeconcopenhagen.netlify.com
- **Install and run Python:** Follow these guides
  1. Installing Python and VSCode
  2. Running Python in JupyterLab
  3. Running Python in VSCode
- **DataCamp:** Online courses on Python (requires no installation)
  $\Rightarrow$ you get 6 months free access (see e-mail with details)
- **Binder:** Online access to interactive version of the course materials (requires no installation)

Time, place and exam

- **Time and place:**
  1. **Lectures:** Monday 15-17
  2. **Classes:** Tuesday/Wednesday 15-17
- **Exam requirements** (deadlines):
  1. Basic programming test (on DataCamp.com, see e-mail)
  2. Data analysis project
  3. 2x useful peer feedback on data analysis projects
  4. Model analysis project
  5. 2x useful peer feedback on model analysis projects
- **Exam:** Data and model analysis projects + exam problem
- **Grading:** Pass or fail
- **Groups:** All projects can be done in *fixed* groups (maximum of 3)

Course plan - lectures

1. Introduction
2. Fundamentals: Primitives
3. Fundamentals: Optimize, print and plot
4. Fundamentals: Random numbers and simulation
5. Fundamentals: Workflow and debugging
6. Working with Data: Load/save and structure data
7. Working with Data: Basic data analysis
8. Supervision on data project
9. Algorithms: Searching and sorting
10. Algorithms: Solving equations (numerically and symbolically)
11. Algorithms: Numerical optimization
12. Further Perspectives: The need for speed
13. Further Perspectives: R and MATLAB
14. Further Perspectives: Julia

Course plan - classes

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. Problem Set 3: Loading and combining data from Denmark Statistics
7. Problem Set 4: Analyzing data from Denmark Statistics
8. Work on your data project
9. Problem Set 5: Writing your own searching and sorting algorithms
10. Problem Set 6: Solving the Solow model
    + feedback on data analysis project
11. Problem Set 7: Solving the consumer problem with income risk
12. Problem Set 8: Comperehension, vectorization and numba
13. Work on your model analysis project
14. Feedback on model project

GitHub.com (code hosting platform)

- **All course materials will be shared on GitHub**
- **Organization:** www.github.com/NumEconCopenhagen

  Repositories:

  1. **lectures-2019**: slides, course plan, guides etc.
  2. **exercises-2019:** problem sets, solutions etc.
  3. **projects-2019-YOURGROUPNAME**: your own repository
  4. **snippets-2019:** your own examples

- **Git:** A version-control system for tracking changes in computer files and coordinating work on those files among multiple people.
  $\Rightarrow$ integrated in VSCode
  $\Rightarrow$ we will talk more about it in week 5

- **Note:** You can always download the content of a GitHub repository without using git.

Work-flow

Your work-flow

- **Lectures:** *Listen to me, ask questions and solve small tasks*
  1. Overview of topic
  2. Introduction to new concepts
  3. Live coding
  4. Presentation of problem set

Your work-flow

- **Lectures:** *Listen to me, ask questions and solve small tasks*
  1. Overview of topic
  2. Introduction to new concepts
  3. Live coding
  4. Presentation of problem set
- **Classes:** Work on problem sets and ask questions to your fellow students and the teaching assistants
  1. Solve tasks and puzzles
  2. Fill the missing code
  3. Find the error
  4. Solve full problem

Your work-flow

- **Lectures:** *Listen to me, ask questions and solve small tasks*
  1. Overview of topic
  2. Introduction to new concepts
  3. Live coding
  4. Presentation of problem set
- **Classes:** Work on problem sets and ask questions to your fellow students and the teaching assistants
  1. Solve tasks and puzzles
  2. Fill the missing code
  3. Find the error
  4. Solve full problem
- **In between classes and lectures:**
  1. Go through lecture notebooks (curriculum)
  2. Solve the problem set
  3. Experiment with your own ideas

Socrative

- **Room-name:** NumEcon
  1. Web: www.socrative.com $\rightarrow$ student login
  2. App: Socrative Student
- **Two use-cases:**
  1. Questions with 5 possible answers (A-E)
  2. Are you done with a task? (true in true/false)

Questions

- **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
  1. Look in the documentation
  2. Talk about it in your group
  3. Search Google + Stackoverflow [see this guide]
  4. Ask question online using GitHub issues [see this guide, example]
- **Help each other!!** You will learn a lot, but remember to be constructive and polite.

Projects

Basic programming test

- **You must complete the following courses on DataCamp**
  1. Intro to Python for Data Science
  2. Intermediate Python for Data Science
  3. Python Data Science Toolbox (Part 1)
  4. Python Data Science Toolbox (Part 2)
- **First 3 classes:** Reserved for your work on DataCamp

Data analysis project

- **Objectives:**
  1. Apply data analysis methods
  2. Structure a code project
  3. Document code
  4. Present results
- **Content:**
  1. Import data from an online source
  2. Present the data visually (and perhaps interactively)
  3. Apply some method(s) from descriptive economics (»samfundsbeskrivelse«)
- **Structure:**
  1. A self-contained single notebook presenting the analysis
  2. Fully documented python files
- **Hand-in:** Create and commit folder called "data_analysis_project" in your GitHub repository

Model analysis project

- **Objectives:**
  1. Apply model analysis methods
  2. Structure a code project
  3. Document code
  4. Present results
- **Content:**
  1. Description of algorithm to solve simple economic model
  2. Solution (and perhaps simulation) of simple economic model
  3. Visualization of results across e.g. parametrizations
  4. Analysis of extensions of the baseline model
- **Structure:**
  1. A self-contained single notebook presenting the analysis
  2. Fully documented python files
- **Hand-in:** Create and commit folder called "model_analysis_project" in your GitHub repository

More examples

More examples

- **I will show** how to
  1. Simulate the AS-AD model
  2. Write modules in VSCode
  3. Run python code in VSCode

NumEcon

Vision

- **Hypothesis:** Teaching in all areas in economics can be improved by giving access to numerical versions of the models taught

Vision

- **Hypothesis:** Teaching in all areas in economics can be improved by giving access to numerical versions of the models taught
- **NumEcon:** A Python package collecting:
  1. Numerical versions of models taught at polit
  2. Other models interesting for polit students
  3. Useful code snippets and examples

Vision

- **Hypothesis:** Teaching in all areas in economics can be improved by giving access to numerical versions of the models taught
- **NumEcon:** A Python package collecting:
  1. Numerical versions of models taught at polit
  2. Other models interesting for polit students
  3. Useful code snippets and examples
- **You can contribute!** Good projects and snippets can become part of NumEcon - beneficial for other students (and your CV)
  1. Minor contributor
  2. Major contributor

Summing Up

Summing up

- **I hope your have:**
  1. An idea of why learning numerical analysis is important
  2. What you will learn in this course
  3. How you will learn it by working actively and interact with your fellow students
  4. How you will qualify for and pass the exam

Your to-do list

1. **First priority:** Login to DataCamp (see info in e-mail)

Your to-do list

1. **First priority:** Login to DataCamp (see info in e-mail)
2. **Second priority:** Try running JupyterLab in binder
   Go to course page click on a 

Your to-do list

1. **First priority:** Login to DataCamp (see info in e-mail)
2. **Second priority:** Try running JupyterLab in binder
   Go to course page click on a 

3. **Third priority:**
   - 3.1 Install **Anaconda with VSCode**
     See the guide: Installing Python and VSCode
   - 3.2 Open **JupyterLab** and experiment
     See the guide: Running Python in JupyterLab

Your to-do list

1. **First priority:** Login to DataCamp (see info in e-mail)
2. **Second priority:** Try running JupyterLab in binder
   Go to course page click on a 

3. **Third priority:**
   - 3.1 Install **Anaconda with VSCode**
     See the guide: Installing Python and VSCode
   - 3.2 Open **JupyterLab** and experiment
     See the guide: Running Python in JupyterLab
4. **Fourth priority:** Read the guides on:
   - Searching for answers using Google and Stackoverflow
   - Asking questions using GitHub issues

Your to-do list

1. **First priority:** Login to DataCamp (see info in e-mail)
2. **Second priority:** Try running JupyterLab in binder
   Go to course page click on a 

3. **Third priority:**
   - 3.1 Install **Anaconda with VSCode**
     See the guide: Installing Python and VSCode
   - 3.2 Open **JupyterLab** and experiment
     See the guide: Running Python in JupyterLab
4. **Fourth priority:** Read the guides on:
   - Searching for answers using Google and Stackoverflow
   - Asking questions using GitHub issues

- **Next time:** Introduction to the fundamental of Python