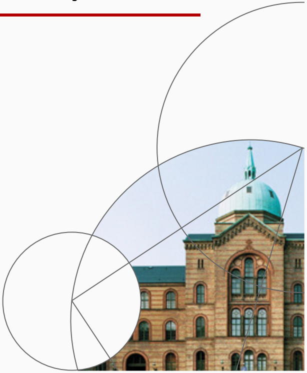
CENTER FOR
ECONOMIC
BEHAVIOR &
INEQUALITY

1. Introduction

Introduction to Programming and Numerical Analysis

Jeppe Druedahl

Spring 2019



Plan

Introduction

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-convexity, where algebraic solutions are not available

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-convexity, where algebraic solutions are not available
- You will get **a better understanding of**
    1. economic theory itself
    2. how to empirically test economic theories
    3. scientific programming in itself
    4. collaboration on code projects

Vision

- **Vision:** All of economics should be taught by a combination of numerical and mathematical analysis

Vision

- **Vision:** All of economics should be taught by a combination of numerical and mathematical analysis

  $\Rightarrow$ **NumEcon:** online repository with Python code to solve economic problems at the bachelor level

Vision

- **Vision:** All of economics should be taught by a combination of numerical and mathematical analysis

  $\Rightarrow$ **NumEcon:** online repository with Python code to solve economic problems at the bachelor level

- **Active learning:** To learn scientific programming you need to work on actual problems yourself

Vision

- **Vision:** All of economics should be taught by a combination of numerical and mathematical analysis

  $\Rightarrow$ **NumEcon:** online repository with Python code to solve economic problems at the bachelor level

- **Active learning:** To learn scientific programming you need to work on actual problems yourself

  $\Rightarrow$ your exam projects could end up as part of NumEcon

Vision

- **Vision:** All of economics should be taught by a combination of numerical and mathematical analysis

  $\Rightarrow$ **NumEcon:** online repository with Python code to solve economic problems at the bachelor level

- **Active learning:** To learn scientific programming you need to work on actual problems yourself

  $\Rightarrow$ your exam projects could end up as part of NumEcon

- **Pioneer spirit:** you are the first generation $\Rightarrow$ you can greatly affect how this course will be taught in the future

Vision

- **Vision:** All of economics should be taught by a combination of numerical and mathematical analysis

  $\Rightarrow$ **NumEcon:** online repository with Python code to solve economic problems at the bachelor level

- **Active learning:** To learn scientific programming you need to work on actual problems yourself

  $\Rightarrow$ your exam projects could end up as part of NumEcon

- **Pioneer spirit:** you are the first generation $\Rightarrow$ you can greatly affect how this course will be taught in the future

- **High level:** Very few (if any) econ bachelor programs provide education on numerical analysis on the level you will get

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 and documenting your code is a central aspect of this course

First Example

First Example

- We work with **Python 3.7**
- **Suggested environment:**
  1. **Distribution:** Anaconda
  2. **Editor/IDE**: VSCode
- **I will show** how to
  1. Run Python in a Jupyter Notebook
  2. Solve the consumer problem

Course Plan

Course plan - lectures

1. Introduction
2. Fundamentals: Primitives
3. Fundamentals: Printing and Plotting
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
    + supervision on model project
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 form Denmark Statistic
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

Infrastructure

Time, place and exam

- **Time and place:**
  1. **Lectures:** Monday 15-17
  2. **Classes:** Tuesday/Wednesday 15-17

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

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

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

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)

GitHub.com (code hosting platform)

- **Our organization:** NumEconCopehagen
  github.com/NumEconCopenhagen

GitHub.com (code hosting platform)

- **Our organization:** NumEconCopehagen
  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. **NumEcon:** Python package
- **Note:** Everything will be public

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:** *No curriculum to read!*
  $\Rightarrow$ spend your time on solving the problem sets
  $+$ experimenting with your own ideas

Questions

- **Observation:** Programming is the slow and painful removal of tiny errors in your code – one at a time

Questions

- **Observation:** Programming is the slow and painful removal of tiny errors in your code – one at a time
- **Ask questions!!** In the following order
  1. Documentation
  2. Your group
  3. Your fellow students
  4. Google + Stackoverflow
  5. The teaching assistants
  6. The lecturer

Questions

- **Observation:** Programming is the slow and painful removal of tiny errors in your code – one at a time
- **Ask questions!!** In the following order
    1. Documentation
    2. Your group
    3. Your fellow students
    4. Google + Stackoverflow
    5. The teaching assistants
    6. The lecturer
- **Online forums:** use **issues** in **exercises-2019** and **lectures-2019** (learn how to following this guide)

Questions

- **Observation:** Programming is the slow and painful removal of tiny errors in your code – one at a time
- **Ask questions!!** In the following order
  1. Documentation
  2. Your group
  3. Your fellow students
  4. Google + Stackoverflow
  5. The teaching assistants
  6. The lecturer
- **Online forums:** use **issues** in **exercises-2019** and **lectures-2019** (learn how to following this guide)
- **Everybody often forgets the correct syntax** ⇒ trial-and-error and testing is central, never a single correct approach

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 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 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. parametrization
  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

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
    4. How you will qualify for and pass the exam
- **Examples:** Can be accessed online through **binder** without installing Python

Your to-do list

1. **First priority:** Login to DataCamp (see info in e-mail)
2. **Second priority:**
   2.1 Install **Anaconda with VSCode**
   2.2 Open a **Jupyter Notebook** and experiment
3. **Third priority:**
   3.1 Create **GitHub account**
   3.2 Clone **exercises-2019** and **lectures-2019**
   3.3 **Create/join group** here

- **Next time:** Fundamental introduction to Python