

INTRODUCTION

QUANTITATIVE ECONOMICS 2024

Piotr Żoch

October 4, 2024

ABOUT THIS COURSE

- Goal of the course:
 - Teach you tools and techniques useful in modern economics.
 - Give you understanding of scientific computing.
 - Prepare you for work on quantitative projects.
- We will:
 - Learn how to write code in Julia.
 - Study elementary numerical methods.
 - Apply recursive methods to economic problems.
 - Solve and simulate economic models.
- **This course:** an **introduction** to the above.

COMPUTATION IN ECONOMICS

- Computational methods are used in many fields of economics:
 - **Macro**: dynamic general equilibrium models, heterogeneous agents, ...
 - **Micro**: dynamic games, life-cycle models, industry dynamics, ...
 - **Econometrics**: machine learning, non-standard estimators, large datasets, ...
 - **International/spatial**: models with heterogeneous firms and countries, dynamic models of trade, spatial models, climate change, ...
 - **Finance**: asset pricing, risk, non-arbitrage conditions, ...
 - **Economic history**: large sets of non-standard information, library data, historical counterfactuals, ...
- Judd (1997): “Computation **helps**, **complements**, and **extends** economic and econometric theory.”

QUANTITATIVE ECONOMICS

Data-driven study that solves and estimates structural models using computational techniques.

- **Question:** measurement
- **Answer:** numbers
- **Key piece:** a structural model (theory of the (aggregate) economy)
- Use the model to get quantitative implications of the theory
- The model is calibrated along some dimensions and used to explain some other dimensions of the data.
- The computer is used to solve for the equilibrium of the model and run computational experiments that answer the research question (and explain mechanism behind the result).

ROADMAP

1. Tools

- Introduction to Julia
- Numerical methods: root finding, optimization, interpolation

2. Techniques

- Recursive methods with discrete and continuous states
- Projection methods

3. Economics

- Consumption-savings problems
- Search models
- Heterogeneous agent models
- Dynamic stochastic general equilibrium models

REQUIREMENTS

1. Problem sets (4) 50%

- Up to five students per group. Two weeks for each problem set. Submit code and write-up via GitHub.

2. Final project 40%

- Three weeks to solve it.

3. Class participation 10%

- Class attendance and participation also rewarded. Sometimes mandatory readings, you will be cold-called to give a short (5 minutes) summary of them at the beginning of class.

LOGISTICS

- We meet on Wednesdays and Fridays at 9:45, Room B107.
- Classes will be a mix of lectures and coding sessions.
- Some classes (mostly practical sessions) will be taught by Marcin Lewandowski and some (mostly lectures) by Piotr Żoch.
- Problem sets will be graded by Marcin Lewandowski.
- Office hours: by appointment, send us an email.

LOGISTICS

- All class materials will be available on GitHub.
- Create a GitHub repo for your group. Send us the link to it.
- Your group composition **must** remain the same throughout the semester.
- Submit **code** and **write-up** via GitHub.
- Your code **must** be in Julia. It can be in a Jupyter or Pluto notebook or a Julia script. Your write-up must be in a PDF.
- We will **not** accept submission via email or other means.
- We will **not** accept late submissions (unless you have a good reason and let us know in advance).
- Your code **must** be reproducible. We need to be able to run it without any modifications (except for installing packages through `install`).

SOFTWARE

- We will teach you some basics of Julia, but it is practice that makes perfect
- Recommended introduction I: Julia Academy
- Recommended introduction II: QuantEcon
- Amazing book: Julia for Data Analysis
- Why Julia? What are the other options?

SOFTWARE

- Low-level languages: good performance (C, C++, Fortran)
- High-level languages: good productivity (Mathematica, Matlab, R, Python)
- Julia: good performance and productivity
 - Modern language.
 - High performance and easy to parallelize.
 - Easy to use.
- In quant. economics you will mostly see Fortran, Matlab, Julia and Python.
- Good to know more than one (+ something like R/Stata).
- Once you know one, it is easy to learn another.