# INTRODUCTION

Quantitative Economics 2025

Piotr Żoch

October 1, 2025

# 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

Loosely: a study that solves and estimates structural models using computational techniques.

- Question: measurement.
- Answer: numbers.
- Key piece: a structural model (theory of behavior / 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 the model and run computational experiments to answer the research question (and explain mechanism behind the result).

## EXAMPLE - INVESTMENT SUBSIDY

Suppose we are interested in the effect of an investment subsidy on firm investment behavior:

- Build a model with profit-maximizing firms that differ in size, sales, employment, and productivity as in the data.
- The investment behavior of firms crucial for this research question. We will choose parameters of the model to match this behavior. Note: some parameters might not correspond directly to what we see in the data - by bringing the model close to the data we learn about their values,
- We now have a laboratory that can help us with the research question. We compare two versions: with and without the subsidy.
- We can examine which features of the model matter the most for the result. Or how the subsidy is introduced (anticipated, non-anticipated).

# EXAMPLE - INVESTMENT SUBSIDY

- Problem of a firm:

$$\max_{\{k_{t+1},l_t,i_t\}_{t=0}^{\infty}} \mathbb{E}_0\left[\sum_{t=0}^{\infty}\frac{1}{(1+r)^t}\Big(pf(z_t,k_t,l_t)-wl_t-(1-\tau)p^I i_t-c(k_{t+1},k_t,i_t)\Big)\right],$$

$$\text{s.t. } k_{t+1}=(1-\delta)k_t+i_t,\quad z_{t+1}\sim Q(z_t,\cdot),\quad \forall t\geq 0,\quad k_0,z_0 \text{ given.}$$

- We will learn how to use dynamic programming to solve such problems.

# EXAMPLE - INVESTMENT SUBSIDY

- Recast it as

$$V(k,z) = \max_{k',l,i} pf(z,k,l) - wl - (1-\tau)p^I i - c\big(k',k,i\big) + \frac{1}{1+r}\int V\big(k',z'\big)Q\big(z,dz'\big),$$

$$\text{s.t. } k' = (1-\delta)k + i.$$

- Solve it to get policy functions such as $i(k,z)$.
- We now know how much capital next period $k'$ chooses a firm with $(k,z)$. Use it together with the stochastic process for $z$ to track the distribution of firms over $(k,z)$, $\mu_t(k,z)$.
- We might be interested in a stationary distribution $\mu(\cdot) = \mu_t(\cdot) = \mu_{t-1}(\cdot)$
- We will learn how to find it.

## EXAMPLE - INVESTMENT SUBSIDY

- How to parametrize the model? For example, let

$$\log z' = \rho \log z + \epsilon', \ \epsilon' \sim N\left(0, \sigma^2\right).$$

  What values of $\rho$ and $\sigma^2$ make sense?

- Calculate various statistics using $\mu(k, z)$ and policy functions. Match them to the data by appropriately choosing parameters of the model.
- Vary $\tau$, compare firm behavior, the implied distribution of firms, and the statistics.
- We can extend our analysis to consider:
  - endogenous prices $p, w, p^I, r$ that clear some markets (need to model other parts of the economy),
  - transition between two stationary distributions (need to consider time explicitly),
  - ...

# 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) 40%
   - Up to five students per group. Two weeks for each problem set. Submit code and write-up via GitHub.
2. Final project 30%
   - Three weeks to solve it.
3. Tests (2) 20%
   - In class, closed book. Verify if you understood the material and not simply used AI.
4. 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.
- All class materials available on GitHub.
- Problem sets will be graded by Marcin Lewandowski.
- Office hours: by appointment, send us an email.

## PROBLEM SETS

- 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.
- We will not accept submission via email or other means.
- Your code must be in Julia. Your write-up must be in a PDF.
- You can use AI tools to help you write code, but you must understand what the code does and be able to explain it. If we have doubts, we will ask you to explain your solution in person.
- 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 `instantiate`).

# 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?
- My view: debates *"X is better than Y!"* are rather unproductive.

# 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.
- Mathematica is very useful for symbolic algebra.
- Good to know more than one (+ maybe something like R/Stata).
- Once you know one, it is easy to learn another. Especially with AI.