Quantitative Economics

Fall 2025

University of Warsaw

Instructors: Marcin Lewandowski & Piotr Żoch
Email: *mr.lewandowski2@uw.edu.pl* & *p.zoch@uw.edu.pl*
Office Hours: by appointment

Course Description This course is a comprehensive introduction to quantitative methods used in modern economics. There are two main parts of the course. In the first part we discuss elementary topics in numerical analysis with applications to statistics, econometrics and economics. In the second part we will focus on recursive methods for solving sequential decision problems. The emphasis will be mostly on tools, techniques and theories used by macroeconomists. We will analyze consumption-savings problems, income dynamics and wealth inequality in partial and general equilibrium. To confront our models with the data we will solve them on a computer and compare their predictions with empirical regularities. Time permitting, we will also cover a range of other tools and techniques used in modern macroeconomics, primarily in models of business cycles.

We will make extensive use of mathematics, numerical methods and computer programming. There is a heavy theoretical component: this course provides basic background in dynamic programming techniques (Bellman equation) frequently used in modern economics. We will also study economic theory related to consumption, savings, wealth and income distributions. There is also a heavy computational component: we will study numerical methods and learn to apply them in various settings.

We will use Julia, a modern, open source, high productivity language primarily used in technical and scientific computing. No prior knowledge of Julia is needed as a brief introduction to this language and example code will be provided.

Students who do not find such a quantitative, computational approach to economics appealing, are <u>strongly</u> advised against taking this course.

Outline:

1. Programming in Julia.

2. Efficient and reproducible workflow: version control, debugging, benchmarking, modules, and packages.

3. Elementary concepts in scientific computing.
4. Linear and nonlinear (systems of) equations.
5. Optimization.
6. Function approximation and interpolation.
7. Numerical integration and Markov processes.
8. Introduction to dynamic programming.
9. Discrete Markov decision processes.
10. Continuous state models.
11. Heterogeneous agent models in macroeconomics.
12. Perturbation methods.
13. (Stochastic) difference equations and linear rational expectations models.
14. Projection methods.

Requirements

- Problem sets (x4) 40%
- Tests (x2) 20%
- Final project 30%
- Class participation 10%

There will be four problem sets. They will usually ask you to derive some theoretical results and/or implement some extensions of material covered in class on a computer and study their properties. If you wish, you may work on the assignments in groups of up to five students. You can (and you are encouraged to do so) discuss your solutions with other students, even those outside of your group. You must credit the people with whom you have worked on the problem set. You will have two weeks for each problem set. You need to submit your code in Julia as well as a write-up.

There will be also a final project, similar in style to problem sets, but longer and more difficult. Its details will be announced in the last week of classes. You will have three weeks to solve it. You are allowed to work on it in a group of up to five students.

Your grade will depend also on two short in-class tests. Each one of them is expected to be 15-20 minutes long and will check your understanding of basics of programming in Julia. This is to ensure that you actually learn, instead of relying fully on AI generated code.

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

On using AI AI tools are permitted as learning aids for problem sets and the final project, but your submissions must demonstrate your own understanding and meaningful contribution. Acceptable uses include getting help with debugging, learning new syntax, and reviewing code you have written. However, you cannot submit code generated entirely by AI without modification or understanding, nor copy-paste AI solutions you cannot explain. We value learning over perfect execution. A working solution you understand is preferable to flawless code written by AI without your engagement. If we have questions about your approach or understanding, we may ask you to explain your solutions in a brief meeting. The goal is ensuring you gain the knowledge this course provides, which requires your active intellectual engagement with the material rather than delegation of all work to AI.

Materials There does not exist a single textbook for this course. We will mostly rely on materials posted here. We will also post journal articles. Textbooks which you might find useful to get better understanding of theoretical aspects of the course are:

- Stokey, Lucas and Prescott (1989) *Recursive Methods in Economic Dynamics*
- Ljungqvist and Sargent (2018) *Recursive Macroeconomic Theory*
- Sargent and Stachurski (2023) *Dynamic Programming Volume I: Foundations*
- Jappelli and Pistaferri (2017) *The Economics of Consumption*

This great book provides a gentle introduction to Julia:

- Kamiński (2023) *Julia for Data Analysis*

Extra resources related to computation in economics:

- QuantEcon: https://quantecon.org/. The most relevant lectures in the Julia module (https://julia.quantecon.org/intro.html) are Lectures 31-38 and 51.
- Heer and Maussner (2009) *Dynamic General Equilibrium Modeling*
- Miranda and Fackler (2002) *Applied Computational Economics and Finance*
- Judd (1998) *Numerical Methods in Economics*

You can find more details about numerical analysis here:

- Kincaid and Cheney (2009) *Numerical Mathematics and Computing*