Computational Economics Lecture 1:
How to Code Like a *Software Development Engineer*

Min Fang

University of Florida

Spring 2025

Outline

1. **Course Introduction: Basics**
2. **Course Introduction: Project**
3. **Beyond Economics: Coding is just Coding!**
4. **Software Development Engineer's Toolbox**
5. **Potential Projects in More Details**
6. **Tasks for Week 1-3**

Course Introduction: Basics 1

- Part I: Basics (Weeks 1-3)
  - Beyond Economics: How to code like a Software Development Engineer: GitHub, IDE, etc...
  - Language Choices: Fast, easy, scalable, or just as long as it works
  - Mathematical Tools: How to turn equations into codes
  - MATLAB/Python/Julia Programming for Economics and Finance
  - Linear Dynamics, Probability and Distributions, Nonlinear Dynamics
  - Stochastic Dynamics, Optimization, Estimation (Basics)
- I will mainly use Julia, but you can use Python or MATLAB for sure.
- In week 2, you will choose a project and present your idea of solving such a project

Course Introduction: Basics 2

- Part II: Intermediate (Main Part; Weeks 4-9)
  - Optimization Methods (Gradient or Gradient-Free)
  - Value Function Iteration, Policy Function Iteration, Endogenous Grid Method
  - Applications: Consumption/Saving/Investment, Job Search, Information, Asset Pricing, . . .
  - Local Approximation: When Dynare would work
- We will solve the same problems with various methods
- We will also show the limitations of different methods

Course Introduction: Basics 3

- Part III: Advanced (Main Part; Weeks 10-14)
  - Heterogeneous Agents/Firms Models
  - Structural Estimation: GMM, MLE
  - Parallel Computing and GPU Computing
  - Deep Learning for Model Solutions
- Time to solve your projects while learning the advanced methods!
- We will have less or no homework on too advanced materials :)

Course Introduction: Project

- We will do a course project for practice purposes; it would require
  - [1] Clear computational setups and optimization
  - [2] Some equilibrium prices to be solved (either partial or general)
  - [3] Taking the model to the data (Calibration, SMM, or MLE)
  - [4] Counterfactual exercises
  - [5] ONE extension of your own
- You are expected to code together but should submit your own project
  - [1] Topic 1: Macro-Firms
  - [2] Topic 2: IO-Firms
  - [3] Topic 3: Labor/Public
- I will include a file for the papers, and you decide and present next week (25mins)
- Presentation (3 groups) should follow the 5 steps as above
- Let me know by Wednesday your choice!

Beyond Economics: Coding is just Coding!

- Okay, enough economics; now we only talk about coding!
- I will first introduce the concept of OOP: Objective-Oriented Programming
- OOP is actually straightforward: Object + (Object) Operator
- Object could be anything: number, matrix, words, even functions (operator)
- Operator is usually just functions: input objects and output objects
- Historically procedural languages: MATLAB, FORTRAN, etc
- Naturally born OOP languages: Java, C++, Python, Julia, etc
- We will demo the OOP features comparing sample codes in Julia and MATLAB
- *Disclaimer: You can write OOP in MATLAB as well; just more effort to be paid

SDE's Toolbox: IDE

- Integrated Development Environment is widely used by developers
- The currently most popular one is VS Code by Microsoft
  - [1] You can use it for almost any language, even MATLAB!
  - [2] GitHub Copilot is your lifesaver! Auto-fill, Chat Q&A, Debug, etc
- Certainly, I don't mind if you do the whole course with just MATLAB Interface
- But I strongly recommend you try out VS Code

SDE's Toolbox: Git, GitHub, Version Control

- Before we talk about GitHub, we should understand what is Git
- The key ideas: Distributed Nonlinear Workflow + Version Control
- Distributed Nonlinear Workflow: forward, backward, circling, etc among many!
- Version Control: Keep tracking the history of the above workflow
- GitHub is such a tool to distribute and control versions
- Use it for your project so you do not easily lose codes or build "spaghetti codes"

Final Projects

- Finally, we are talking about exciting final projects!
- I want to do it slightly different from what people have published
- We could publish your final project as a package on GitHub (Julia/Python)!
- In the next pages are several projects that you could choose from
- Let me know which one you would like to do by Wednesday (repeating myself...)

No.1: Macro-Monetary Project

- Probably the hardest but most familiar from my side
- "Financial Heterogeneity and the Investment Channel of Monetary Policy"
- The steps for the computation of the quantitative model:
  - [1] Solving individual firm's problem given equilibrium prices
  - [2] Solving equilibrium prices (general equilibrium in both steady state and transition)
  - [3] Taking the model to the data (Calibration)
  - [4] Counterfactual exercises of monetary shocks
  - [5] ONE extension of your own

No.2: Firm-Dynamics Project

- Still quite hard, but you could handle it!
- "Entry, Exit, Firm Dynamics, and Aggregate Fluctuations"
- The steps for the computation of the quantitative model:
  - [1] Solving individual firm's problem under aggregate uncertainty
  - [2] Solving equilibrium prices (general equilibrium)
  - [3] Taking the model to the data (Calibration)
  - [4] Counterfactual exercises of industrial subsidies
  - [5] ONE extension of your own

Other Projects

- You can also propose some other projects
- As long as the steps for the computation of the quantitative model:
  - [1] Solving individual firm's problem under aggregate uncertainty
  - [2] Solving equilibrium prices (general equilibrium)
  - [3] Taking the model to the data (Calibration)
  - [4] Counterfactual exercises of industrial subsidies
  - [5] ONE extension of your own
- It could be IO/Labor/Public, and you could present later (not next week)

Tasks of Week 1-3

- Pick a project or find a project
- Short presentation of the paper structure next week (individual)
- Install Julia, Python, and MATLAB
- Install GitHub Desktop, Copilot (Free from students)
- Go over the first five sections of "Quantitative Economics with Julia": Chapter 1 to Chapter 27 (https://julia.quantecon.org/intro.html)
    - Getting Started with Julia
    - Package Ecosystem
    - Software Engineering
    - Tools and Techniques
    - Introduction to Dynamics

Appendix

Additional Links

- Cursor AI: https://www.cursor.com/
- Actually, my friend told me I am outdated; we should all use Cursor now!