47-809 **Computational Methods for Economics** Mini-4 (Spring) 2023

Professor: David Childers
dchilder@andrew.cmu.edu,
TEP 5140
Office hours: Thursday 3:30-4:30 or by appointment

Grader: Arnav Sood arnavs@andrew.cmu.edu

Lectures: Mon, Wed 4:00-5:50, TQ4219

Objectives. This is a course in the basic tools of numerical analysis that can be used both to assess the quantitative implications of economic theory and to derive theoretical results of economic models without analytical solutions. While most examples will come from macroeconomics, the generality with which the techniques will be presented in this course will make them applicable to a wide range of fields like econometrics, financial economics, marketing, and microeconomics. To enable efficient application of numerical tools, this course endeavors to explain not only when and how to use various numerical algorithms but also how and why they work; in other words, the intention of the course is to open up some "black boxes" and provide the students with a versatile tool set.

The course will cover basic topics in numerical methods such as systems of equations, optimization problems, functional approximation and numerical integration. The course will then focus on solving dynamic problems using approaches like differential equations and dynamic programming. Subsequent advanced topics will include heterogeneous agent models or other methods if there is student interest.

Prerequisites. There are no formal prerequisites, but the course assumes that students are familiar with real analysis and particularly (constrained) optimization, linear algebra as well as the basics of differential equations, and functional analysis. It is assumed that you have learned Dynamic Optimization theory (Bellman equations) from Macro I.

You also need to know Matlab, Julia, Python, or another programming language suitable for numerical computation.

Textbook. The main text is "*Numerical Methods in Economics*" by Ken Judd. Other readings might be assigned as necessary. Due to lack of time we will cover material in substantially lower level of detail than the textbook. The homework and exam will be based on material covered in class, unless you are explicitly asked to read a specific section of the book. Electronic copies of this text are available via the CMU library.

Optional Texts. Additional topical readings will be suggested with topics. Particularly useful resources include the following

- The QuantEcon Project https://quantecon.org/
  - The best resource for coding and computation for economics.

  - Julia and Python versions are a source for code implementation for most methods we will discuss in class, as well as basic programming help.
- *Economic Dynamics: Theory and Computation* by John Stachurski
  - Covers theoretical aspects particularly clearly
- Solution and Estimation Methods for DSGE Models by Jesús Fernández-Villaverde, Juan Rubio-Ramírez, and Frank Schorfheide in Handbook of Macroeconomics Volume 2A 2016
  - Up to date on tools specifically for macroeconomics
- MIT Class 18.337J/6.338J Parallel Computing and Scientific Machine Learning by Christopher Rackauckas.
  - High performance computing, machine learning, and advanced Julia for larger-scale projects
- Algorithms for Decision Making by Mykel J. Kochendorfer, Tim A. Wheeler and Kyle H. Wray
  - Modern dynamic programming and extensions with Julia code

Software. In-class lectures will reference Julia, MATLAB, and possibly Python, though we may specialize to one based on class interest. Provided homework solutions are in Julia with Matlab versions are available upon request, and some partial results in Python. You are free to use other programming languages (C++, Fortran, Mathematica), but you will still be responsible for producing the same numerical results as well as a readable code. We may also spend time learning a modeling language like AMPL, in which case you will be required to use it; it is available via Web interface, so you do not need to worry about installing it. You are not allowed to use Excel, and it will be very hard to do so anyway.

Homework. Programming is a practical skill, so doing problem sets is the best way to understand the material. The lowest-scoring PS will not count towards your course grade (with the exception of the last one).

You can work in teams of up to three people. If you do, submit a single problem set but write down the names of all the team members. You can switch teams or choose to work alone.

When solving homework, you are allowed to use class materials and built-in Matlab, Julia, or Python help and basic programming guides. You are not allowed to receive help from people outside of your team, or to use ready-made code from the internet, code generating tools like Chat-GPT or Copilot, or to use previous-year solutions. Violations of this rule will reduce your learning and will be punished by zero scores.

Problem sets are due at the beginning of the class on the due date. Your solution should include a text document with results in numeric or graphical form, along with proper verbal discussion. This document should be uploaded to Canvas as a single document. The code should be also uploaded, as a single .zip file. You may, if you prefer, combine code and text in a notebook format, in which case please provide both your .ipynb (or other notebook code format) file along with the compiled version containing output.

Communication. In addition to office hours, we may have occasional recitations or makeup classes on Friday; keep an eye on Canvas announcements. The professor will respond to the e-mail questions within 24 hours, or 36 hours if the e-mail was sent on a weekend.

Final exam will be a take-home exercise, with questions similar to homework ones. We will discuss the date in class. Unlike homework, it has to be done individually.

Grading. 60% of the grade will determined by the problem set scores (excluding the lowest one, as described above) and 40% - by the take-home final.

Copyright. As the class materials will be distributed online, there will be audio, video, and other materials available; however, you may not distribute these materials beyond class participants without instructor's permission.

Integrity. You are bound by the University's academic integrity rules. Which means you cannot use work of others (including your classmates) without naming them. General rule is: when in doubt, cite your source.

Acknowledgements. The course material is derived from the materials of the previous instructor of this class, Yaroslav Kryukov, itself based on the similar courses taught by Ken Judd (University of Chicago & Hoover Institution), and Ulrich Doraszelski (Wharton @ UPenn).

Topics
The list of topics and class dates is below, subject to change depending on pace of the class. One make up day, likely in the third week of April, is likely to be scheduled.

1. Numerical Analysis: General Considerations (Mar-13)
Chapters 1 and 2. General ideas of convergence rates, computational errors, error analysis.

2. Linear Equations (Mar-15)
Chapter 3. LU, QR, and Cholesky decomposition, condition numbers, Gauss-Jacobi and Gauss-Seidel methods.

3. Nonlinear Equations (Mar-20)
Chapter 5. Gauss-Jacobi, Gauss-Seidel, Newton, continuation and homotopy methods. Application to general equilibrium.

4. Optimization (Mar-22)
Chapter 4. Search methods, bisection, Newton method, BFGS and DFP updates. Applications to consumer demand problems, incentive problems, and structural estimation.
Possible extra topic: Modeling languages (possibly JuMP/AMPL): Coding up the model, selecting solvers, submitting the model and processing the output.

5. Functional Approximation (Mar-27,29)

Chapter 6. Local Interpolation, Global Orthogonal polynomials, Splines, Nonlinear methods.

6. Numerical Integration and Monte Carlo simulation (Apr-3)
Chapters 7, and 8. Integration methods for single- and multidimensional integrals. Monte Carlo simulation methods. Applications to portfolio choice and Bayesian problems.

7. Differential and Functional Equations (Apr-5)
Chapters 10 (DE) and 11 (Projection). Methods for solving differential equations.

8. Numerical Dynamic Programming (Apr-10,12)
Chapter 12. Solutions to deterministic and stochastic dynamic programming problems. Applications to saving and consumption problems.

9. Optimal Control (Apr-17):
Chapter 10.6-.7. Optimal Control and Pontryagin Principle

10. Perturbation Methods (Apr-19):
Chapter 13. Local solution methods at first and higher order, Dynare

11. Advanced Topics (Apr-26,28):
- Heterogeneous Agent Models
- Other topics of students' choice – let the professor know your preference
    - Neural networks and differentiable programming methods
    - Probabilistic programming and Bayesian computation
    - Algorithmic Game Theory