Course Syllabus for Econ 70310-01

Course Title: Numerical Methods for Structural Economic Models, Fall 2017
Course Meetings: M W 12:30-1:45 PM in 3005 Nanovic

Instructor: Professor Zach Stangebye
Email: zstangeb@nd.edu
Website: www.zachstangebye.com
Office: 3015 Nanovic Hall, (574) 631-1054
Office Hours: T W Th 10:30-11:30, or by appointment

General Information

This syllabus consists of the general information on the logistics of the course and various procedures. At the end of this document is a class-by-class outline of the material as I expect to cover it over the course of the semester.

This class assumes knowledge of content from the first-year PhD sequence i.e. Econ 60101, 60201, 60302, 60102, 60202, and 60303.

Contact Policy

The best way to reach me is either by e-mail or during office hours. I will respond to e-mails within a 24 hour period on weekdays or by Monday evening if the question was posed on a weekend. You are also more than welcome to call my office number during weekdays at (574) 631-1054. If I am around at the time, I will answer.

Honor Code

Any behavior not in compliance with Notre Dame's Academic Code of Honor will be dealt with according to the consequences for violation contained in the Honor Code: http://honorcode.nd.edu/

Title IX

Please note that all professors are considered mandatory reporters of harassment and discrimination by Title IX law. Mandatory reporters are responsible for reporting all Title IX violations of sex-based discrimination and sexual harassment, including sexual violence, stalking/dating violence, and child sexual abuse. This means that if students disclose cases that meet this definition in written or verbal communication, I am obligated to report that to our Title IX office so they can help provide support and track such incidents. I welcome your deepest sharing but also want you to understand that all professors are bound to this law.

Sakai

I will be making use of the web-based program, Sakai, to which all registered students should have access. It can be accessed via insideND. Lecture slides, take-home exams, readings, etc. will all be available on this site.

Learning Goals

Throughout this course, you will develop

1. A working knowledge of programming languages used for technical computing
2. A working knowledge of a wide range of solution methods applicable to a wide class of structural economic models

3. A working knowledge of both classical and frontier-level algorithms used to implement the aforementioned solution methods
4. A working knowledge of a slew of techniques to improve the efficiency of your solution techniques for the purposes of calibrating and/or estimating structural models
5. The ability to apply your knowledge to new research projects.

Textbooks, Readings, and Resources

All required material will be found in lecture slides/notes **and** codes for lecture notes. They are a roughly a convex combination of the following useful references

1. "Numerical Methods in Economics," Kenneth L. Judd (1998)
    a. Old standard, but a little heavy to read
2. "Economic Dynamics: Theory and Computation," John Stachurski (2009)
    a. Solely focused on coding and dynamic programming
    b. A little heady, but very thorough references for this
    c. Sort of like an updated Stokey, Lucas, Prescott (1989) with coding techniques
3. "Dynamic General Equilibrium Modeling: Computational Methods and Applications," Burkhard Heer and Alfred Maussner (2009)
    a. Good, readable book. Covers most techniques we will cover
    b. Organization is a little sloppy, but content good
4. "Applied Computational Economics and Finance," Mario J. Miranda and Paul L. Fackler (2002)
    a. Nice, readable resource, but nothing on perturbation methods
    b. Very little on projection methods
5. "Recursive Methods in Economic Dynamics," Nancy Stokey, Robert Lucas, Jr., and Edward C. Prescott (1989)
    a. Classic work. No direct computational references, but fantastic exposition of dynamic programming methods
6. "Solution and Estimation Methods for DSGE Models," Jesus Fernandez-Villaverde, Frank Schorfheide, and Juan Rubio Ramirez (2016)
    a. Chapter from the Handbook of Macroeconomics, Vol. II
    b. Solid, very readable reference. No dynamic programming, though
    c. Content will be taken solely from Part I of this chapter
7. Unpublished notes by Fabrice Collard
    a. http://fabcol.free.fr/notes.html
8. Useful website by John Stachurski and Tom Sargent
    a. https://lectures.quantecon.org
    b. **This should be your first go-to source for programming/syntax questions**
9. An NBER online course by Jesus Fernandez-Villaverde and Lawrence Christiano
    a. http://www.nber.org/econometrics_minicourse_2011/
    b. These guys can probably explain many things better than me! But this content is not representative of our whole course
10. JuliaCon annual conference lectures. All available on Youtube in the channel "Julia Language"
    a. https://www.youtube.com/channel/UC9IuUwwE2xdjQUT_LMLONoA
    b. Many more details on parallelization, debugging, IDE navigation, plotting, packages, etc.

In this course, we will primarily be working in the programming language, Julia. The advantages of this language are numerous and will be exposited in this class. It is quite similar to Matlab or Python, so developing fluency in one will help you with the others. You can download Julia and its IDE (Integrated Development Environment), Juno, for free at the link below:

- https://juliacomputing.com/products/juliapro.html

You can also execute Julia code in your browser using the innovative, cloud-based JuliaBox service at the link below:

- https://juliabox.com/

You can find pretty good, quick-start references for writing in Julia at the links below. **These should be your next go-to sources for programming/syntax questions**:

- https://learnxinyminutes.com/docs/julia/
- https://www.youtube.com/watch?v=x0lI4omOYTw&list=PLeo1K3hjS3uvHr7N7j3GcMj48SdiTa_19

Python is also a free, open-source environment. Matlab is not free, but a student license is relatively affordable and amounts to roughly the price of a textbook should you choose to purchase it.

Problem Sets

There are 4 problem sets that together constitute 60% of the final grade. They count for 15% each. They are spaced roughly three weeks apart from each other. They will be mostly (but not exclusively) code-based and as such you will be required to submit **both** your answers to the questions **and** any code that you used to generate those answers.

You will be required to submit your problem sets in Julia code. Julia is becoming an increasingly popular language in the profession at large, and so learning its ins and outs is a useful tool, even if you are already fluent in alternative programming languages.

Final Project

40% of your final grade will be determined by a final project. You will have one of two options for this:

1. Design and implement a solution method for a research project of your own (either a current project, or a project that you'd like to begin).
2. A final assignment given by me. In contrast to problem sets, this will be one, relatively larger and more sophisticated model that you will solve from the ground up.

The final project will be due during the final exam slot. You must consult **with me before Thanksgiving Break** to let me know which option you will choose. You will be required to hand in the following for the final project.

1. A complete description of the model you are solving.

2. A complete description of your solution method.
    a. Includes a justification for your method
3. A complete description of the algorithm you use to implement your solution method.
    a. Includes a justification for your choice of algorithm
    b. Includes all steps taken to improve efficiency of your algorithm
4. The code you designed to solved your problem.
    a. Make sure your code is well-commented and readable
5. The Euler Equation errors for your benchmark case.
6. The time-to-run for your benchmark code.
7. All requests made by me at our meeting prior to Thanksgiving break. This will obiously be project specific and will likely (but not necessarily) consist of
    a. A calibration (or at least a calibration strategy and ability to soon implement it)
    b. Model simulations
        i. Simulated model moments
        ii. Simulated impulse-response functions (standard or stochastic)
    c. Plots of policy and/or value functions
    d. Illustrative exercises
    e. Counterfactual exercises

Your final project need not be a new project, but you should implement a *new* solution method. If you choose to solve an existing project, I will be in touch with your advisors to ensure that you are designing a new solution. You will also be required to present comparisons in Euler Equation errors, times-to-run, and key model moments between your old solution and your new one.

Also, for the final project you *need not use Julia* if you find it to be a hindrance. You need only provide a reasonable justification for your language of choice.

Grading

Your grade will be decomposed of the following:

Problem Sets..……………..…60%

Final Project………………....40%

There will be no curve for this class. The grade decomposition from the final grade will be as follows:

- A: 93-100
- A-: 90-92.9
- B+: 87-89.9
- B: 83-86.9
- B-: 80-82.9
- C+: 77-79.9
- C: 73-76.9
- C-: 70-72.9
- D: 60-60.9

- F: Less than 59.9

Final grades will be graded on a letter scale, with the implied number grade for purpose of aggregation determined by the midpoint of each interval e.g. a B on a paper will translate to 85. The only exception would be an A+, which will imply a 100. This could be a paper grade but it cannot be a course grade because it cannot be entered into the registrar.

Re-grade Policy

If you believe a grading error has occurred and would like to request a re-grade, you are welcome to do so. However, a request for a re-grade must contain both the assignment itself and a brief paragraph *in writing* explaining why you believe you deserve a re-grade. Also, re-grades will only be allowed for *one week* after the problem set or paper has been handed back to you. Afterward, all grades are final, even if an error occurred. Also, be aware that your grade could go down as well as up if additional mistakes are found, since re-grading is a comprehensive process.

Assignment Timeline

- Sept 6th: Problem Set 1 Due
- Sept 27th: Problem Set 2 Due
- Oct 23rd: Problem Set 3 Due
- Nov 20th: Problem Set 4 Due
- Final Exam Slot (TBD) : Final Project Due

Content Covered (Subject to Change)

The course will be broken into roughly 7 topics:

1. Introduction to numerical methods in economics
    a. Uses and applicability in economics
    b. Basics of coding
    c. Basic useful algorithms
2. Functional analysis
    a. General recursive forms of dynamic models
    b. Analysis of errors and assessment of accuracy
3. Dynamic programming
    a. Forward induction/time-difference solution methods
    b. Backward induction
    c. Numerical derivatives and expectations
    d. Functional operators: Contractions and monotonicity
    e. Value function/policy function iterative methods
4. Perturbation methods
    a. Linear and log-linear (certainty equivalence)
    b. Higher-order and pruning strategies
    c. Dynare implementation
5. Projection methods
    a. General approach
    b. Choice of basis functions
    c. Residual functions and collocation
    d. Finite elements and discontinuous methods
6. Efficiency gains

    a. Parallelization techniques
    b. Endogenous grid methods
    c. Stylized/tailored algorithms
    d. Efficient memory management
7. Heterogeneous agent models
    a. Solving the Aiyagari model
    b. Introducing aggregate uncertainty: Krusell-Smith Approach

The majority of content will be introduced in the context of canonical structural economic models, though the properties of these models are not the focus of the course per se.

<u>Content will be updated as the semester progresses a few weeks in advance to include potentially new material.</u>

1. August 23rd
    a. Introduction and course overview
    b. Static vs dynamic languages: Traditional trade-offs and current advances
    c. Starting simple: Command-line terminal, Juno IDE, and Jupyter
    d. Data types
    e. Loops and conditions
    f. Basic examples of numerical methods in economics
        i. Interval bisection and market clearing
2. August 28th
    a. Scope: Local and global variables in Julia
    b. Basic examples of numerical methods in economics continued
        i. Gradient-free maximization techniques (grid-based and golden-search)
        ii. Computation of demand and supply functions
        iii. Use of numerical solution for economic analysis
    c. Plotting in Julia
3. August 30th
    a. Basic examples of numerical methods in economics continued
        i. Maximization techniques (gradient-based)
            1. Newton-Raphson and Bisection Methods
            2. Numerical derivatives
            3. Autodifferentiation in Julia
        ii. Optimization packages
            1. The Optim package
            2. Constrained and multivariate optimization with JuMP and NLopt
4. September 4th
    a. Distributions: Densities and draws
    b. Basic examples of numerical methods in economics continued
        i. Markov Chains
    c. Numerical integration and expectations
        i. Quadrature methods
        ii. LLN and Monte Carlo methods for expectations
        iii. Tauchen's method for Markov Chains
5. September 6th

    a. **Problem Set 1 Due**
    b. Functional Equations: Deterministic Systems
        i. Recursive Representations
        ii. Neoclassical Growth Model I: Finite Horizon
            1. Grid method
            2. Euler Equation method
            3. Interpolation methods: Linear piecewise, spline
6. September 11th
    a. Functional Equations: Deterministic Systems
        i. Neoclassical Growth Model II: Infinite Horizon
        ii. Dynamical Systems and the Shooting Algorithm/Finite-Difference Solution Technique
7. September 13th
    a. Functional Equations: Stochastic Systems
        i. RBC Model
        ii. Functional Analysis: Theory and application
            1. Fixed point analysis
            2. Contractions and Monotonicity
8. September 18th
    a. Functional Equations: Stochastic Systems
        i. RBC Model
        ii. Dynamic Programming I: Value Function Iteration
9. September 20th
    a. Functional Equations: Stochastic Systems
        i. RBC Model
        ii. Dynamic Programming II: Euler Equation/Policy Function Iteration
        iii. Method of the endogenous grid
10. September 25th
    a. Functional Equations: Stochastic Systems
        i. The Eaton-Gersovitz limited commitment model
        ii. Pathological cases: Exploiting what you know
            1. Monotonicity
            2. Smoothing
11. September 27th (guest lecture)
    a. **Problem Set 2 Due**
    b. Perturbation Methods: Approach
        i. Theory
        ii. Usefulness and limits
12. October 2nd
    a. Perturbation Methods: Linear Implementation
        i. Constructing steady states
        ii. Obtaining derivatives
        iii. Solving
        iv. Log-linearization
        v. RBC Model
13. October 4th

    a. Perturbation Methods: Higher-Order Implementation
        i. RBC Model with Epstein-Zin preferences
        ii. Certainty equivalence (or failure there-of)
14. October 9th
    a. Perturbation Methods: Higher-Order Implementation
        i. Explosive paths and pruning
        ii. Perturbing the value function: Welfare analysis
15. October 11th
    a. Perturbation Methods: More sophisticated models
        i. Dynare implementation
        ii. Analysis of errors
        iii. New Keynesian Model
16. FALL BREAK
17. October 23rd
    a. **Problem Set 3 Due**
    b. Tools: Nonlinear optimizers and solvers
        i. JuMP and related packages revisited
        ii. Efficiency in higher dimensions
18. October 25th
    a. Projection Methods: Basics
        i. Deterministic models
        ii. Spectral Basis functions
        iii. Neoclassical Growth Model Revisited
19. October 30th
    a. Projection Methods: Measuring the Residual
        i. Collocation
        ii. Subdomain, Galerkin, and Least-squares methods
20. November 1st
    a. Projection Methods: Adding shocks
        i. Discretization approach
        ii. RBC Model
21. November 6th
    a. Projection Methods: Adding states
        i. Multidimensional bases and tensor products
        ii. Complete polynomials
22. November 8th
    a. Projection Methods: Adding states
        i. Smolyak's Algorithm
23. November 13th
    a. Projection Methods: Finite Elements as basis function
        i. Approach and refinement classes
    b. Hybrid methods
        i. Perturbation + Projection
        ii. Projection + VFI (least-squares updates)
24. November 15th
    a. Auxiliary algorithms: Simulation methods

        i. Computing the ergodic distribution
        ii. Stochastic impulse response functions

25. November 20th
    a. **Problem Set 4 Due**
    b. Speeding Things Up: Parallelization
        i. Theory and Implementation
26. November 22nd (NO CLASS: Thanksgiving)
27. November 27th
    a. Heterogeneous Agent Models
    b. Distributions as State Variables: The Aiyagari model
        i. Statement of the problem/Description of solution
28. November 29th
    a. Heterogeneous Agent Models
    b. Implementation of solution: Solving for the distribution
        i. Iterative procedure to compute equilibrium interest rate
        ii. Counterfactual analysis of income/consumption distribution
29. December 4th
    a. Heterogeneous Agent Models: Adding Aggregate Shocks
    b. Krusell-Smith Solution Approach
        i. `Approximate Equilibrium' and solution strategy
30. December 6th
    a. Heterogeneous Agent Models: Adding Aggregate Shocks
    b. Krusell-Smith Implementation
        i. Analysis of errors
        ii. Cyclical properties of income/consumption distribution
31. Final Exam Slot (TBD)

    a. Final Project Due