106: TOPICS IN ECONOMIC THEORY

Instructor: A. Banerji

Course Outline

This course will provide an introduction to optimization and dynamics in discrete time. A principal focus will be dynamic programming. The treatment will be rigorous, so the theory will be developed slowly. The required mathematics will be developed within the course. Time permitting, we will have a module in which we numerically solve dynamic programming problems on the computer, for which the choice of language will almost surely be Python.

The optimization and dynamics will take a long time. We will have little time left to discuss equilibrium models beyond baby ones. This course design therefore will give a lower current payoff in order to have a higher continuation value, in dynamic programming parlance. Thus it makes sense for you to take this course only if you wish to continue your study of economics and can postpone the natural desire to use all of the tools immediately in substantive applications.

The applications of these tools (that we won't really see) are widespread in macroeconomics. They are also used in dynamic games; recursive methods in particular have been developed in the class of repeated games with imperfect public monitoring. Markov perfect equilibria are also a natural extension of these tools, in some sense. We won't have time for all of this.

Prerequisites: You would have learned (i) basic math econ, (ii) probability and depending on your year, have learned or will be learning (iii) game theory in the compulsory courses. The knowledge from (i) and (ii), mainly, is presumed. But in particular, you will find things doable (if hard) only if you were comfortable with the level of real analysis used in Course 002. That was not much, but the level was comparable to small parts of a Real Analysis I course. In the present course, the level will sometimes be comparable to some parts of a Real Analysis II course. For the numerical methods for dynamic programming, no prior knowledge of numerical methods or programming languages is assumed; if we use Python, we'll start from scratch.

Of course, if you've done more math or more programming than the prerequisites, and game theory as well, you may find it easier going. But in any case, the main things needed are a good deal of mathematical sophistication, the ability to work hard, and the desire to learn what may appear to be a dry subject.

Books:

[1] Nancy Stokey, Robert Lucas with Ed Prescott (1989): Recursive Methods in Economic Dynamics.

The first half of the above book will be the main required reading. We may sometimes refer to this book as the bible (for now), in the secular sense of being a classic. The lectures may often resemble a close reading of the classics, with active participation from you.

The bible will take up much of the time you allocate for this course. Some by and large optional reading is listed below. [2] is a very basic, user-friendly backup for Real Analysis at an introductory level. The typical application in Rangarajan Sundaram's optimization textbook that we used for Course 002 is more advanced than [2]. For Math and probability at the level we will use and higher, see [3] to [6] below.

[2] Stephen Abbott (1997): Understanding Real Analysis, 2nd edition.

[3] Efe Ok: Real Analysis with Economic Applications (the chapters on Metric Spaces).

[4] Richard Bass (2013): Real Analysis for Graduate Students (the chapters on Measure and Integration).

[5] Achim Klenke (2014): Probability Theory – A Comprehensive Course (the first 4 to 8 chapters).

[6] Marek Capinski and Peter Kopp: Measure, Integral and Probability. (a nice, gentle introduction).

For some of the dynamics, and for exercises in dynamics using the Python language, we have readings [7] – [10].

[7] John Stachurski (2009): Economic Dynamics – Theory and Computation.

[8] Thomas Sargent and John Stachurski (ongoing): Quantitative Economics. (Access the 'book' at its website, quant-econ.net. You can always download a pdf version for your computer/tablet/phone. You can access the codes at github: https://github.com/QuantEcon/QuantEcon.py. Github is a social network for collaborative projects that uses the version control software git.) Several of the applications on this site resemble extracts from

Lars Ljungqvist and Thomas Sargent (2012): Recursive Macroeconomic Theory (3rd edition).

When we venture into numerically solving dynamic programming problems, we may use discrete dynamic programming code developed by contributors to this website.

General sources for starting out with Python could include an early chapter each from [7], [8], and early chapters from [9], [10]. Chapter 7 of [10], on object-oriented features of Python, is particularly recommended.

[9] Allen Downey (2013): Think Python.

[10] Hans Peter Langtangen (2012): A Primer on Scientific Programming with Python.

For a detailed introduction to computational methods in macro, which we will not do at all, see [11]. [12] is a go to reference for numerical methods in economics more generally.

[11] B. Heer and A. Maussner (2009): Dynamic General Equilibrium Modeling: Computational Methods and Applications.

[12] Kenneth Judd (1998): Numerical Methods in Economics.

Topics: (Starred readings are required).

1. Metric Spaces; Contraction Mapping Theorem; Correspondences; Theorem of the Maximum; Blackwell's conditions for a Contraction.
   Readings: Lecture Notes*, [1]* (Chapter 3), [7] (Chapter 2).

2. Deterministic Dynamic Programming and Applications.
Readings: [1]* (Chapters 4,5).

3. Deterministic Dynamics:

Deterministic dynamical systems.

Readings: Lecture Slides*, [7]* (Chapter 4), [2], [1]*(Chapter 6). The listing for topic 4 is tentative. A good treatment of these dynamical systems (basically, a system of first-order difference equations) should include a discussion of so-called 'forward-looking' solutions; and there are a couple of nice sources for this that are not listed here. But I will probably skip all this at this stage; so it's likely I'll just do a little bit here.

4. Measure, Integration and Probability:

Classes of Sets, Measures, Probability Measures, Measurable Functions, Integration, Monotone Convergence Theorem, Fatou's Lemma, Dominated Convergence Theorem, Product Measures, Fubini's Theorem, Conditional Expectation.

Readings: Lecture Notes*, [1]*(Chapter 7). For those looking for more comprehensive treatments, you could refer to [4](Chapters1-15, 21), or [5] (Chapters 1-4).

5. Markov Processes:
Transition functions, probability measures on spaces of sequences, iterated integrals, stochastic difference equations.
Readings: [1]* (Chapter 8, especially Chapter 8.1*).

6. Stochastic Dynamic Programming and Applications:
Principle of Optimality, bounded returns, constant returns to scale, unbounded returns, stochastic Euler equations, policy functions and transition functions. Applications: Optimal growth, Industry investment under uncertainty, Search models.
Readings: [1]*(Chapters 9, 10; especially 9.2* and a few applications from Chapter 10).

7. Additional Topics and Numerical Applications:
(i) Strong Convergence of Markov Processes – Markov Chains.
(ii) Discrete Dynamic Programming.
Readings: Lecture Notes*, [1]* (Chapter 11.1), selected sections from [7]*, [8]*. If you want to look ahead at numerical applications, you are strongly urged to install Anaconda, a Python distribution that comes rolled in with Python Libraries NumPy, SciPy and Matplotlib, used for numerical applications. Just do Chapter 1 of [8], line by line, starting with instructions to install Anaconda.

Internal Assessment will be based on a mix of a midterm, coding exercises, and an exercise on typing up lecture notes for a section of the course; the final form will be determined soon, after consulting you. We can discuss the structure of the final exam later.