Computational Methods in Macroeconomics

ECON 216, Winter 2010, Weds 5-7:50 pm Economics 304

Instructor: Irina A. Telyukova
E-mail: itelyukova@ucsd.edu
Office: Economics 224
Phone: 822-2097
Office hours: by appointment (please e-mail in advance)

I. Description of the Course

Models in modern macroeconomics often are analytically intractable and have no closed-form solutions. In order to answer many interesting questions in the field, macroeconomists have been leaning more and more on numerical solutions of their models using powerful computers. In this course, we will learn many of the standard computational techniques required to solve such models. We will cover some macroeconomic theory by studying some standard classes of macro models that require the computational approach. An equally important focus of the course is on tools. We will begin by a brief overview of dynamic programming, followed by a survey of numerical methods required when analytical methods are not available for solving equations and systems of equations, for finding derivatives and integrals, and for solving optimization problems. After this, in the context of some standard models, we will look at methods that can be used to solve and parameterize them. Toward the end of the course, as we learn some methods, you can expect to see a few papers that use them presented in class.

II. Requirements for the Course

1. Homework. By its nature, the learning of methods requires using them. So there will be a number of hands-on homeworks assigned, which you should do carefully. I encourage you to talk to each other as you do these to speed up the learning. Learning computation for the first time is quite time-consuming; be ready to spend the time! At each homework due date, submit a summary of results (which include economic analysis), and your code, *by e-mail.* I will grade these informally to give you an idea of where you stand. Your “cumulative” homework performance will account for 60% your final course grade.

2. Project. By the first Monday of the finals week (March 15), you are to submit a referee report on a computational paper from the course reading list (or one that’s not on the list that you like, after prior discussion with me). Apart from a typical referee report, you are to write a research proposal section at the end where you either propose an interesting extension of the paper, or a related question on which you yourself might consider writing a paper, employing methods learned in class. This part is to be taken very seriously. To do a good proposal, it may be necessary for you to review other literature related to the paper at hand, and I expect not only a write-up of an idea, but also a fairly clear plan on how you would pursue the execution of the idea, computation included. *On the last day of class, we will have each of you present your idea to the class, in a short talk.* This project and its presentation will account for 40% of your grade.

III. Programming Languages

This is not a computer science course, so I will not teach you programming - this part is up to you. So it helps if you have had previous programming experience in a language such as C, but it is not a prerequisite - you can learn as you go, and I do not expect prior programming experience. If you intend to work with computation in your research, now is the time to invest in learning a low-level language. Scientists and Economists everywhere work with Fortran 90 (or 95); if you prefer, you can work in C (or C++, though the more low-level the language, the better). Fortran and C are much (I have heard numbers in the hundreds of times) faster than Matlab and other comparable languages. The speed will really matter if you ever plan to attempt any serious computation-based research. Matlab can be a fine substitute if you are not planning on taking computation far beyond this course. I will make available notes from a previous student on how to access a Fortran compiler on the iacs5 server. Matlab is widely available directly on the lab computers. I do not know much about C compilers, but it is my belief that they are not hard to come by and we may have them in the department's lab as well. *Before you begin, think hard about which language you will work in.* I cannot tell you how many people have told me in the profession how much they wish they had learned one of these languages when they were in grad school. At any point afterwards, the time and other costs of learning will be much higher than they are now.

For Fortran and Matlab alike, there are lots of codes available online - which is a good way to learn. There is also a good Matlab primer available if you have never used it before. *Numerical Recipes* cited below are great for help when you are learning Fortran 90. Finally, there is a book by Ellis, Phillips and Lahey, *Fortran 90 Programming*, that is a good reference (though not a necessity) - though it may be out of print.

IV. Relevant Books

The following forms the beginnings of an excellent reference library for a macroeconomic theorist who is interested in computation. I will refer you to parts of these books when relevant.

1. Stokey, N., R. Lucas with E. Prescott. *Recursive Methods in Economic Dynamics.* 1989.
2. Sargent, T. and L. Ljungqvist. *Recursive Macroeconomic Theory.* MIT Press, 1998.
3. Heer, B. and A. Maußner. *Dynamic General Equilibrium Modelling.* Springer, 2005.
4. Judd, K. *Numerical Methods in Economics.* MIT Press, 1998.
5. Marimon, R. and A. Scott, eds. *Computational Methods for the Study of Economic Dynamics.* Oxford University Press, 1999.
6. Cooley, T.F., ed. *Frontiers of Business Cycle Research.* Princeton University Press, 1995.
7. Press, W., S. Teukolsky, W. Vetterling and B. Flannery. *Numerical Recipes in Fortran77: The Art of Scientific Computing.* Cambridge University Press, 2001. *Available full-text online along with a Fortran 90 edition; the Fortran 95 edition, on the other hand, is not free.*

V. Tentative Course Outline - Not Necessarily in Order of Presentation

1. **The Computational Experiment**

   The big picture of what we will be doing in this class.

2. **Dynamic Programming**

   A brief review, because you already know it.

   Reference: Stokey, Lucas, Prescott (1989), chapters 4,5.

3. **Representative Agent Models**

   Motivation: neoclassical growth model (NGM); real business cycle (RBC) model; non-optimal RBC models - e.g. with distortionary taxation; extensions - multiple choice variables and inequality constraints.

   (a) Value Function Methods
   - Discretization
   - Linear quadratic methods
   - Finite elements methods
   - Weighted residuals methods

   (b) Euler Equation Methods
   - Linearization
   - Perturbation
   - Policy function approximation
   - Parameterized expectation approach

   References: Heer and Maußner (2005), chapters 1-4. Marimon and Scott (1998), ch. 2-7. Kydland and Prescott (*Econometrica* 1982); Danthine and Donaldson (in Cooley, 1995).

4. **Bewley-Aiyagari Heterogeneous Agent Model**

   Motivation: models with infinitely-lived agents subject to idiosyncratic uninsurable risk; equilibria with nondegenerate distributions of agents.

   (a) Stationary equilibrium (Aiyagari, *QJE* 1994)

   (b) Comparison of steady states: transition paths

   (c) Aggregate uncertainty (Krusell-Smith, *Macro. Dyn.* 1998; Algan, Allais, Den Haan, *JEDC* 2008)

   Reference: Heer and Maußner (2005), chapters 5-6; Sargent and Ljungqvist (2004), chapters 16, 17; Rios-Rull (1999) in Marimon and Scott (chapter 11).

5. **Overlapping Generation Models**

   Motivation: models with many generations of finitely-lived agents; equilibria with nondegenerate distributions of agents; evaluation of redistributional policy.

   (a) Solution method (Huggett, *JME* 1996)
   (b) Comparison of steady states: transition paths (Conesa and Krueger, *RED* 1999)
   (c) Aggregate uncertainty (Rios-Rull, *REStud* 1996)

6. **Models where Standard Recursive Methods Do Not Apply**

   Motivation: optimal policy questions and contractual problems.

   (a) Recursive Contracts (Marcet and Marimon, 1994; Aiyagari, Marcet, Sargent and Seppala, 2002 – flexible prices; Schmitt-Grohe and Uribe, 2004 – sticky prices).
   (b) Abreu, Pierce and Stachetti approach (1990).

7. **Underlying It All: Numerical Methods**

   (a) Function Approximation Methods
       - Finite-elements methods: linear interpolation, splines, etc.
       - Weighted-residuals methods
       - Chebyshev polynomials
   (b) Numerical Differentiation
   (c) Numerical Integration
       - Newton-Coates quadrature
       - Gaussian quadrature
       - Monte-Carlo methods
   (d) Root Finding (Solving Systems of Equations)
   (e) Numerical Optimization

   Reference: Heer and Maußner (2005), chapter 8; Judd (1998).

8. **Calibration, If There Is Time**

   (a) Classic approach: Cooley and Prescott (1995), in Cooley (chapter 1).
   (b) A more involved approach: Chatterjee, Corbae, Nakajima and Rios-Rull (2002).
   (c) Estimation: Gourinchas and Parker (*Econometrica* 2002).