

Econ 561b  
Yale University  
Spring 2004  
Prof. Tony Smith

**Syllabus for**  
**COMPUTATIONAL METHODS FOR ECONOMIC DYNAMICS**  
**ECON 561b**

**Course Objectives:** The goals of this course are to teach students the basic tools of numerical analysis and to illustrate how these tools can be used to address analytically intractable problems in economics and econometrics. The underlying theme of this course is that computational methods belong in every economist's toolkit. These methods can be used not only to explore the theoretical implications of economic models in the absence of analytical solutions, but also to assess the *quantitative* importance of the various forces at play in an economic model. Thus computation can help to advance both theoretical and empirical work in economics. In order to teach students how to wield computational tools in an informed and intelligent way, 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 course opens up the "black boxes".

Another theme of this course is that computational methods are vital to all types of research in economics, from the most theoretical to the most applied. To this end, the substantive applications in the course are drawn from a wide range of fields, including macroeconomics, finance, game theory, industrial organization, public finance, contract theory, and econometrics. The course will pay special attention to dynamic economic problems, including methods for solving stochastic dynamic programming problems, for computing equilibria with heterogeneous firms and consumers, and for estimating the parameters of dynamic economic models.

**Contact Information**

Office: 28 Hillhouse, Room 306  
Office phone: (203) 432-3583  
Email address: [tony.smith@yale.edu](mailto:tony.smith@yale.edu)  
Course web site: [fasttone.gsia.cmu.edu/econ561b](http://fasttone.gsia.cmu.edu/econ561b)  
Office hours: Thursdays from 10AM–noon, or by appointment

**Class Meetings:** The course meets on Mondays from 4PM to 5:20PM in Room B8 (28 Hillhouse) and Wednesdays from 4PM to 5:20PM in Room 106 (28 Hillhouse).

**Texts:** The required textbooks for this course are: *Numerical Methods in Economics* by Kenneth L. Judd (MIT Press, 1998) and *Numerical Recipes in Fortran 77 (Second Edition)* by William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery (Cambridge University Press, 1992). Both of these books are available in the Yale bookstore. Students who are familiar with the C or C++ programming languages may want to use versions of *Numerical Recipes* geared towards these languages. *Numerical Recipes in Fortran 90: The Art of Parallel Scientific Computing* is a useful companion volume to *Numerical Recipes in Fortran 77*; in addition to using a more modern version (or standard) of the Fortran programming language, this book shows how to write Fortran programs that take advantage of parallel computing. The Numerical Recipes books are also available online at: [www.nr.com](http://www.nr.com).

Other (optional) books that students might find useful are: the *Handbook of Computational Economics (Volume 1)*, edited by Hans M. Amman, David A. Kendrick, and John Rust (North-Holland, 1996); *Computational Methods for the Study of Dynamic Economies*, edited by Ramon Marimon and Andrew Scott (Oxford University Press, 1999); *Frontiers of Business Cycle Research*, edited by Thomas F. Cooley (Princeton University Press, 1995); and *Dynamic Economics: Quantitative Methods and Applications* by Jérôme Adda and Russell Cooper (MIT Press, 2003).

**Grading:** Occasional problem sets will constitute 30% of the course grade and a project (described in more detail below) will constitute 70% of the course grade.

Students may use any programming language to complete the problem sets, including Fortran, C, Matlab, and Gauss. If you do not already know a programming language, Matlab is probably the easiest to learn. If you want to do state-of-the-art research using computational methods, however, you will need ultimately to learn a fast high-level programming language such as Fortran or C. This course will not teach programming *per se*, but it will teach and emphasize general principles of programming, such as simplicity, clarity, structure, replicability, and testing. Since one of the goals of this course is to teach students what is going on *inside* the “black boxes” of numerical algorithms, students are asked to avoid the use of such black boxes except for routine tasks.

The project will consist of the application of computational methods to a problem in economics or econometrics. This problem could be original research, perhaps as a first step towards a Ph.D. dissertation. However, the problem need not consist of original research: one acceptable option for the project is, in fact, to attempt to replicate the computational results in an existing paper.

The written report for the project should consist of three parts: a brief description of the problem, a detailed description of the computational methods used to solve the problem (including a copy of the code), and a thorough description of the numerical results.

# COURSE OUTLINE

## 1 Basic numerical methods

- Introduction (built around the Huggett-Aiyagari model; see references in Section 3 below); general considerations in numerical analysis (convergence, roundoff error, truncation error), numerical differentiation.  
[Judd: Chapters 1, 2, and 7.7; *Numerical Recipes*: Chapters 1 and 5.7]
- Root-finding in one or more dimensions (bisection, secant method, Newton's method, fixed-point iteration, Gauss-Jacobi, Gauss-Seidel, Brent's method).  
[Judd: Chapter 5; *Numerical Recipes*: Chapter 9]
- Minimization in one or more dimensions (golden section search, Brent's method with or without derivatives, simplex method, Newton-Raphson, conjugate gradient methods, variable metric methods, simulated annealing).  
[Judd: Chapter 4; *Numerical Recipes*: Chapter 10]
- Interpolation and approximation of functions (linear interpolation in several dimensions, cubic splines, polynomial interpolation, searching a table, orthogonal polynomials).  
[Judd: Chapter 6; *Numerical Recipes*: Chapters 3 and 6]
- Random numbers, simulation, and introduction to asymptotic theory.  
[Judd: Chapter 8; *Numerical Recipes*: Chapter 7]
- Integration (cubic spline integration, Gaussian quadrature, Monte Carlo integration, integration of multivariate normal densities).  
[Judd: Chapter 7; *Numerical Recipes*: Chapter 4]

## 2 Numerical dynamic programming

- Linear-quadratic methods
  - Uhlig, H. (1999), "A Toolkit for Analysing Nonlinear Dynamic Stochastic Models Easily," in: *Computational Methods for the Study of Dynamic Economies*.
  - Anderson, E., L.P. Hansen, E.R. McGrattan, and T.J. Sargent (1996), "Mechanics of Forming and Estimating Dynamic Linear Economies," in: *Handbook of Computational Economics*.
- Second-order methods