Objective

This course introduces computational approaches for solving economic models. It focuses on a broad range of numerical methods and then applies them to economic problems. We formulate economic problems in computationally tractable forms and use numerical analysis techniques to solve them. We will study examples of computational techniques in the current economic literature as well as discuss areas of economic analysis where numerical analysis may be useful in future research of dynamic economic problems. The substantive applications will cover a wide range of problems including public finance, macroeconomics, game theory, mechanism design, econometrics, finance, and industrial organization.

Lectures

The lectures will be given 18:00 CET (9:00am PST). They will be streamed using Zoom. They will also be recorded and put on the web. Exact details will be posted later. Ken Judd will deliver the lectures except where noted otherwise below.

Overview

We begin with an overview of the necessary numerical analysis and approximation theory. First, we review standard numerical analysis - interpolation and approximation techniques, numerical optimization, numerical solutions to systems of linear and nonlinear equations, numerical integration, and basic solution techniques for ordinary and partial differential equations. We also discuss perturbation methods useful in economics.

Second, there will be an emphasis on applications related to current research areas. We will discuss methods for solving dynamic programming problems, as well as dynamic stochastic equilibrium models. We will solve for optimal incentive mechanisms using numerical optimization.

The course aims to acquaint students with the range of techniques that have been useful in economic analysis as well as expose students to techniques that have potential use in economic applications. I want to develop a basic understanding of numerical methods, demonstrate their
use in economic examples, show how existing techniques fit into the broader numerical literature, and point the students to potentially useful numerical techniques.

**Prerequisites**

It is best if students have a good undergraduate background in mathematics (such as calculus, linear algebra, eigenvalues and eigenvectors) and are familiar with static optimization theory (such as Lagrangians and Kuhn-Tucker conditions) as well as basic dynamic optimization theory. I will also presume some experience with graduate level economics, however any student who is interested in the course and has taken either Masters or PhD courses in econometrics, microeconomics, and macroeconomics will be able to follow the course.

**Course Grade**

Doing computation is the only way to learn computation. The course grade will be based on computational exercises assigned every week along with term projects. I encourage you to work in groups of two or three.

**Computing Languages**

Students will need to know use some computational language. My recommendation is that you use Python, Matlab, or R. Students are welcome to use FORTRAN, C, or C++. Statistical software such as TSP, Eviews, RATS, and SAS are not acceptable. The US Federal Reserve uses Eviews but I expect more of you. Don’t even ask about Excel. If you do not know any suitable language, I recommend that you learn Python or Matlab. Both are adequate for this course and useful in research. Python is much better in terms of long-run value and flexibility.

In operations research, the popular languages are GAMS and AMPL. They are is easy to learn and allows you to access the best numerical software through the NEOS website (https://neos-server.org/neos/). NEOS time is FREE, reflecting the “Wisconsin idea”! If there is interest, I will have a couple of optional tutorial sessions on AMPL (sorry, I don’t know GAMS).

I will use Mathematica in my lectures and some classnotes. It is particularly useful for symbolic methods and more flexible than Matlab for doing serious programming. Mathematica also has excellent graphics commands. It is free for UZH students. All students will be asked to download the free CDF player (https://www.wolfram.com/cdf/); it will allow you to run my examples on your laptops.

**Textbook**

The text is *Numerical Methods in Economics* by Judd, published by MIT Press. I will be particularly appreciative of any comments you may have on the text (typos, lack of clarity, etc.) since I am beginning to prepare the second edition.

I am working on a second edition. As we progress, I will (hopefully) have drafts on the course website of chapters for the second edition.
Course Outline and Schedule

Feb. 18: Introduction
Chapter 1: Computational power. Computational math as an economics problem: managing scarce resources. Economics VERSUS computational mathematics

Feb. 20: Computer arithmetic
Chapter 2. General ideas of computational errors, and rates of convergence. Finite precision arithmetic and finite - difference derivatives

Feb. 25: Linear algebra and equations
Chapter 3. LU, QR, and Cholesky decomposition, condition numbers, Gauss-Jacobi and Gauss-Seidel methods.

Feb. 27: Unconstrained optimization
Chapters 4 and 5. Search methods, bisection, gradient descent, Newton’s method, derivative free optimization (DFO). Applications to consumer demand and life-cycle problems, and maximum likelihood estimation.

March 3: Nonlinear equations
Chapters 4 and 5. Bisection, Newton’s method, BFGS and DFP updates, and Powell hybrid. Applications to general equilibrium and Nash equilibrium.

March 5: Constrained optimization: theory and methods
Chapters 4 and 5. Linear and nonlinear optimization. KKT conditions, augmented lagrangian, SQP and interior point methods

March 10: Constrained optimization: applications
Introduction to multiobjective optimization. Applications to consumer demand and incentive problems

March 12: Structural estimation I
Basic ideas. MPEC versus NFZP

March 17 Finite-difference ODEs

March 19 Git and software tools (Robert Erbe and Gregor Reich)

March 24 Automatic Differentiation (Philipp Mueller)

March 26 Homotopy (Philipp Mueller and Karl Schmedders)
Chapter 5. Applications will include general equilibrium, Nash equilibrium of static gamea.
March 30: Numerical quadrature, MC, qMC

April 2: Approximation I
Chapter 6. Interpolation, regression, orthogonal polynomials, splines, least squares, LAD and Lasso fits.

April 7: Dynamic optimization, equilibrium, NLCEQ

April 9, 14, and 16 (No lectures. Easter break)

April 21: Dynamic programming - discrete state
Chapter 12. Value function iteration, policy iteration, acceleration methods.

April 23: Structural estimation II
Su-Judd and Skrainka-Judd papers.

April 28: Dynamic programming - continuous state
Chapter 12. Solutions to deterministic and stochastic dynamic programming problems using approximation, integration, and optimization methods. Applications to savings-consumption problems, climate change policy, and portfolio problems.

April 30: Projection methods I
Chapter 10, 11, and 17. Methods for solving ordinary differential equations as well as the more complex equations arising in dynamic economic models.

May 5: Projection methods II
Chapter 10, 11, and 17. Methods for solving ordinary differential equations as well as the more complex equations arising in dynamic economic models.

May 7: Approximation II
Neural nets, radial basis functions, machine learning

May 11 - May 30: “Office hours”
Ken Judd will be in Zurich. He will be happy to chat with students during this time.

May 12: Perturbation methods
Chapter 13, 14, and 15. Taylor series approximations to find numerical solutions of equations, linearizing around a steady state, simple bifurcation methods.

May 14: DSGE, NK models

May 19: Structural estimation III (Philipp Mueller and Gregor Reich)
May 20: Dynamic games
(Note that this is a Wednesday. Ascension Day on May 21 is a holiday in Zurich.)
Markov perfect equilibria of dynamic games.

May 26: Polynomials
Solving systems of polynomials. Homotopy methods and Groebner bases.

May 28: Concluding remarks: future of computational economics