Advanced Macroeconomics

Introduction

Tomás R. Martinez

INSPER

Introduction

What is this course about?

- **Macroeconomics with Heterogeneity**: we will go beyond the representative agent.
  - Household heterogeneity: earnings and wealth.
  - Firm heterogeneity: productivity, employment, and capital.
- **Computational Methods**: mainly the methods to solve these models, but a lot can be generalized/used for other topics.
- We will also discuss calibration and the use of data in macro.

Macro and Heterogeneity?

Table 7: Unconventional DSGE Features

Year	Heterogeneous agents	Finite horizon	Unconventional expectations / preferences	Indeterminacy	Continuous Time
1980	0		18	0	
1990	10	38	30	10	13
2000	10	10	27	0	5
2006–10	15	6	19	6	3
2016–18	29	4	25	9	9

Note: The figures are percentages of all articles in the *JME* and *JMCB*, plus the E-designated articles in the five general interest journals. The 2006–10 figures use data from 2006, 2008 and 2010; and the 2016–18 figures use data from 2016, 2017 and 2018.

Source: Glandon et al (2021, NBER WP).

Numerical Methods?

Table 8: Solution and Fitting Techniques in Theory-Centric Research

Year	All articles: numerical methods	——— Of DSGE: ——— Calibration	Optimization	Of optimized DSGE: Bayesian methods
1980	9	0	0	0
1990	33	0	0	0
2000	71	89	11	0
2006–10	78	73	27	15
2016–18	81	63	37	23

Note: The second column reports the share of theory-based articles that use numerical methods to solve or fit the model. The third and fourth columns report the shares of articles with quantitative DSGE models using calibration versus optimization to fit the model. The fifth column reports the share of DSGEs with optimized fit that use Bayesian methods. See section 3.1.5 for details. The calculations are based on articles in the *JME* and *JMCB*, plus the E-designated articles in the five general interest journals. The 2006–10 figures use data from 2006, 2008 and 2010; and the 2016–18 figures use data from 2016, 2017 and 2018.

Source: Glandon et al (2021, NBER WP).

Data?

Table 10: Econometric Methods and Data Types Over Time

	Methods		Data				
Year Year	Time series	Applied micro	Micro data	Time series	Cross section	Panel	Proprietary
1980	75	25	22	89	8	3	13
1990	62	38	28	70	14	16	32
2000	58	42	28	54	8	38	30
2006–10	46	54	41	42	13	45	41
2016–18	35	65	56	34	10	56	52

Note: The figures are the shares, expressed as percentages, of econometrics-based articles articles in the *JME* and *JMCB*, plus the E-designated articles in the five general-interest journals. The method and data attributes are defined in section 3.2. The 2006–10 figures use data from 2006, 2008 and 2010; and the 2016–18 figures use data from 2016, 2017 and 2018.

Source: Glandon et al (2021, NBER WP).

Evaluation

Problem Sets (30%): Individual, but you should discuss the solutions among yourselves.

Presentation (30%): Individual.

Research Proposal (40%): Individual or in pairs. If you do it in pairs, I am allowed to be stricter in the evaluation

Presence will be required.

Presentation

- 30 minutes presentation followed by a 10-15 minutes discussion (if time permits).
  - Leave time for questions during the presentation. The audience **SHOULD** ask questions.
- A paper on one of the topics of the course. You are free to choose which paper to present (upon my approval).
- There are some suggestions in the syllabus. You can also tell me your interests and I will be happy to give you extra suggestions.
- When? Either by the end of the course or after I finish the "topic" related to the paper chosen (household heterogeneity, firm, etc).
  - Let me know your paper as soon as possible to decide the exact date.

Presentation: Guide for the Presenter

Goal: Explain the paper and why it was successful.

- What is the question this paper aims to answer?
- What is the contribution to the literature?
- You should convey **one main idea** to the audience
- Do not spend too much time on technicalities or robustness/extensions.

Some extra tips about the presentation **here** and **here**.

Presentation: Guide for the Presenter

At the end of the presentation, have a slide to kickstart the discussion

- What are the crucial assumptions (i.e, modeling assumptions or empirical identification strategy)? Can we relax?
- Is there any "dead bodies" hidden (i.e, some crucial details that the author might have "forgotten" to discuss)?
- Any possible extensions/applications?

You don't need to have it all. Feel free to add anything that you believe is interesting to discuss.

Research Proposal

- It does **NOT** to be in the topics covered in the course. In fact, it doesn't even need to be in macro.
- The idea is to stop wasting time and start working in your thesis/paper.
- Feel free to meet with me to discuss your proposal in the middle of the semester or before you submit the final version.
- **Deadline:** To be handled in the end of the course.

Research Proposal

- **Evaluation criteria:** half of the points are given to **feasibility**, the other half to **originality/potential**.

- There is no pre-given format. It can have sections or not, it can have tables or equations or not.

- There is no page max or min, and it does not need to be long (about 2-5 pages is enough).

- If you can, try to go a bit beyond than just a research proposal
  - It should be well motivated, briefly discuss the literature, and highlight possible contributions.
  - If it is empirical, do some data work. If it is theoretical/quantitative, solve a simple or analytical version of the model.

Research Proposal

It is never easy to have a good research project. Some potential entry points:

- Uncover a new stylized fact and discuss how this affects existing theories.
- Evaluate a policy or a (macro-)economic phenomena.
- Start from an existing economic model and discuss why it cannot explain some phenomena; provide an extension to address it.

Many other ways to start a project; when in doubt, look at the **real world**.

If you are in doubt, come talk to me. I may have projects that you might find interesting.

Problem Sets

- Three computational problem sets. You are free to use the programming language of your choice (PSET #2 has to be done in Python since we will use a specific package).
- **Delivery**: The solution should be a short PDF with the answers and a code that replicates your results.
- Your code should be **REPLICABLE**.
  - Try to set the path only once at the beginning of the code;
  - Specify any extra package that you are using apart from the main code (especially user-written code);
  - Specify the version of the software used.

Problem Sets: Which programming language should I use?

Figure: Comparison across languages: Neoclassical Growth Model with Value Function Iteration

Table 1
Average and relative run time (Seconds).

	Mac			Windows		
Language	Version/Compiler	Time	Rel. Time	Version/Compiler	Time	Rel. Time
C++	GCC-4.9.0	0.73	1.00	Visual C++ 2010	0.76	1.00
	Intel C++ 14.0.3	1.00	1.38	Intel C++ 14.0.2	0.90	1.19
	Clang 5.1	1.00	1.38	GCC-4.8.2	1.73	2.29
Fortran	GCC-4.9.0	0.76	1.05	GCC-4.8.1	1.73	2.29
	Intel Fortran 14.0.3	0.95	1.30	Intel Fortran 14.0.2	0.81	1.07
Java	JDK8u5	1.95	2.69	JDK8u5	1.59	2.10
Julia	0.3.7	1.91	2.62	0.3.7	1.80	2.37
Matlab	2014a	7.91	10.88	2014a	6.74	8.92
Python	Pypy 2.2.1	31.90	43.86	Pypy 2.2.1	34.14	45.16
	CPython 2.7.6	195.87	269.31	CPython 2.7.4	117.40	155.31
R	3.1.1, compiled	204.34	280.90	3.1.1, compiled	184.16	243.63
	3.1.1, script	345.55	475.10	3.1.1, script	371.40	491.33
Mathematica	9.0, base	588.57	809.22	9.0, base	473.34	626.19
Matlab, Mex	2014a	1.19	1.64	2014a	0.98	1.29
Rcpp	3.1.1	2.66	3.66	3.1.1	4.09	5.41
Python	Numba 0.13	1.18	1.62	Numba 0.13	1.19	1.57
	Cython	1.03	1.41	Cython	1.88	2.49
Mathematica	9.0, idiomatic	1.67	2.29	9.0, idiomatic	2.22	2.93

Source: Aruoba and Fernández-Villaverde (2015, JEDC).

Problem Sets: Which programming language should I use?

- Most quantitative macroeconomics is done using: Matlab, Fortran, Python, Julia (also R + Rcpp).

- **Speed is fundamental**, but very often the choice of the language is a bit overrated.

- Once you pick a language, try to understand the details that make the language faster.
  - e.g., memory allocation, vectorization in Matlab, Numba in Python, type declaration in Julia, etc.
  - A well-written code in a language that you master is often faster than a badly written code in another "supposedly faster" language.

- I will not teach you the basics of programming, but I can provide some references for Python, Julia, Matlab and Fortran.
  - For Python and Julia see QuantEcon: https://quantecon.org/.

Programming

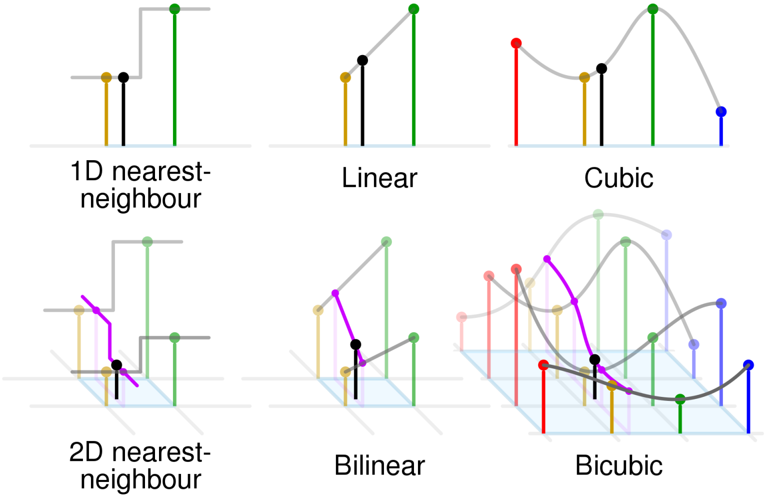
Some stuff that you should learn as soon as possible

- Open your code editor and write simple array operations, loops, etc.
- Some basic plotting.
- Some basic numerical methods: optimization, root finding, numerical integration and differentiation, function interpolation, etc
  - Most languages already have the algorithms implemented, do not try to re-invent the wheel.

Interpolation

- Very often we represent a function with a set of points, but need to evaluate the function "off-grid".
- How to connect the dots? Linear, splines, Chebyshev...
- **Issues**:
  - Kinks;
  - Shape preservation (concavity, convexity, etc);
  - Derivatives;
  - Bounds and extrapolation;

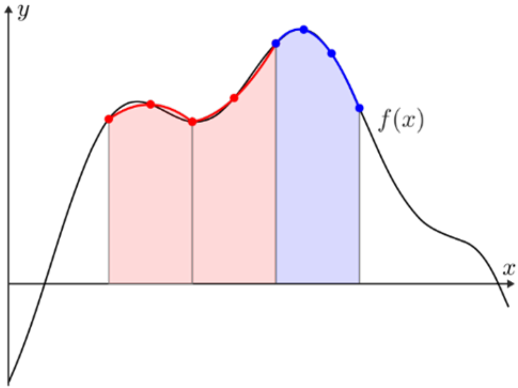
Interpolation



Numerical Integration

$$\int_a^b f(x)dx \approx \sum_{j=1}^{N} v_j f(x_j)$$

- Gaussian quadratures;
- Midpoint quadratures;
- Monte-Carlo methods (simulation);



Numerical Derivative

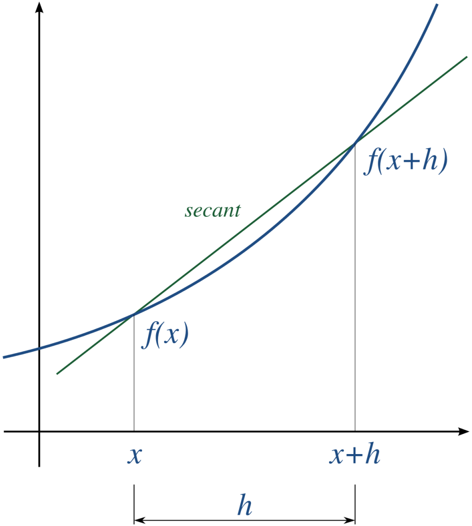
- Finite differences;

$$\frac{f(x+h) - f(x)}{h}$$

  we can do it forward, backward, or symmetric.

- Symbolic differentiation;

- Automatic differentiation;



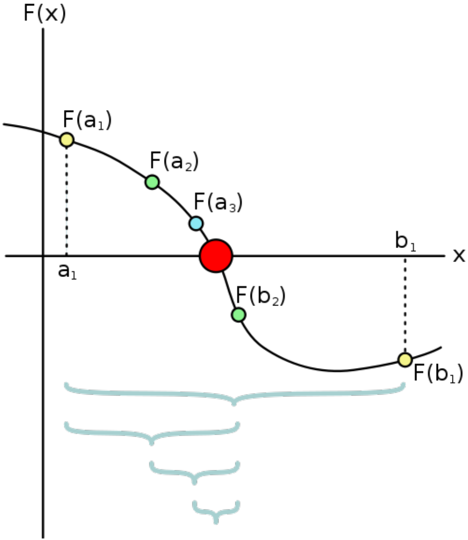
Root Finding

- Every time you solve for equilibrium, you solve a root-finding problem:

$$ED(x) = Demand(x) - Supply(x)$$

- Derivative-free / Bracket methods:
  - Bisection;
  - Brent's method (recommended);
- Newton's / Quasi-Newton methods;
  - Fast, but requires derivatives.
- Easy problem if one-dimensional; Multi-dimensions are much harder and problem dependent.
- **Tip:** one can always transform the root-finding in a minimization problem: $f(x) = (ED(x))^2$.

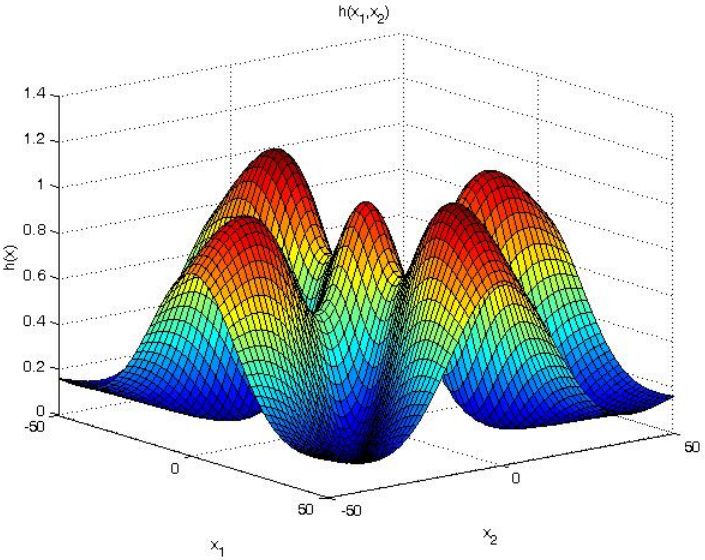
Root Finding



Optimization

- **Derivative-free methods**:
  - Nelder-Mead Simplex: slow, but very robust;
  - Golden-section;
  - Genetic algorithms (e.g., differential evolution): very slow, but can get closer to global solution;
- **Derivative methods** (Newton, Gradient descent, etc):
  - Fast, but requires derivative.
  - Often get stuck in local solution;
- **Issues**:
  - Global or local solution?
  - Constraints?
  - Function domain?
  - Initial guess?

Optimization



Problem Set 0

- To warm-up your programming knowledge, we will solve a simple static economic model: **the Lucas Span-of-control (1978)**.
- It will require some numerical integration and root-finding, but under some assumptions we can solve the model analytically.
- **Exercise**: solve the model analytically and numerically to practice. You should find the same result.

Lucas Span-of-control

- Every period the agent decides whether to be an entrepreneur or a worker.
  - **Worker**: she earns the market wage, w.
  - **Entrepreneur**: she earns profits, $\pi(z)$.

- Agents are heterogeneous in their managerial ability $z \sim F(z)$.

- An Entrepreneur produces the final good with technology $y = zn^{\alpha}$. Profits:

$$\pi(z) = \max_{n} \{zn^{\alpha} - wn\}$$

Lucas Span-of-control

- Solution of the profit maximization of the entrepreneur implies:
  - Individual labor demand:
$$n^d = \left(\frac{\alpha z}{w}\right)^{\frac{1}{1-\alpha}}$$
  - Profits:
$$\pi(z) = \frac{z^{\frac{1}{1-\alpha}}(1-\alpha)\alpha^{\frac{\alpha}{1-\alpha}}}{w^{\frac{\alpha}{1-\alpha}}}$$
- If $\pi(z) \geq w$ agent decides to be an entrepreneur. Let z^* be the cutoff rule s.t. $\pi(z^*) = w$. Using the profit equation:
$$z^*(w) = \frac{w}{(1-\alpha)^{1-\alpha}\alpha^{\alpha}}$$

Lucas Span-of-control

- Equilibrium: aggregate labor demand is equal to aggregate labor supply.
  - Aggregate labor demand, sum all the individual demands:

$$N^d(w) = \int_{z^*(w)}^{\infty} n^d(z; w) f(z) dz,$$

    where $f(z)$ is the probability density function of $F(z)$.
  - Aggregate labor supply, all the agents that decide to be workers

$$N^s(w) = F(z^*(w))$$

- Equilibrium involves finding the wage that equalizes:

$$N^s(w) = N^d(w)$$

- Must integrate numerically to compute the demand and use a root-finding to solve the equilibrium condition.

Lucas Span-of-control

- **Closed-form solution**: suppose $F(z)$ is Pareto with support $[z_m, \infty)$:

$$F(z) = 1 - \left(\frac{z_m}{z}\right)^{\gamma} \qquad \text{and} \qquad f(z) = \frac{\gamma z_m^{\gamma}}{z^{\gamma+1}}$$

- Equilibrium condition becomes:

$$1 - \left(\frac{z_m}{z^*}\right)^{\gamma} = \int_{z^*}^{\infty} \left(\frac{\alpha z}{w}\right)^{\frac{1}{1-\alpha}} \frac{\gamma z_m^{\gamma}}{z^{\gamma+1}} dz,$$

- Suppose $1/(1-\alpha) - \gamma < 0$ (so the integral is well defined). Substitute $w(z^*)$ and we get:

$$z^* = z_m \left(\frac{\gamma - 1}{\gamma(1-\alpha) - 1}\right)^{\frac{1}{\gamma}}.$$

  Using z^* we can get w, the income distribution, the firm size distribution, etc.