On Computational Economics

Giorgio Ricchiuti
www.grarchive.net

UNIVERSITÀ DEGLI STUDI FIRENZE

UNIVERSITÀ CATTOLICA del Sacro Cuore
CLE
Complexity Lab in Economics

Firenze,2023

- This is just a talk about CE
- The tentative program is:
  1. What is CE;
  2. Some notes on softwares;
  3. CE and Teaching;
  4. CE and Research;

Computational Economics is a research field at the **crossroads** between computer science and economics.

The main objective is the analysis of economic modeling through the application of **computational and/or graphical tools**.

We can learn CE in three different ways:

- through computational methods, focusing on softwares (Excel, Mathematica, Maple, Matlab, Python)
- via mathematical methods, focusing on algorithms to solve various classes of mathematical models (difference or differential equations, nonlinear programming, and so on)
- or via economic areas (micro? macro? finance? game theory?)

I do prefer the last of these three approaches: the study of economic models using computational methods.

My approach in learning and teaching CE is as follows:

1. Detect the economic problem;
2. Transform it in a mathematical model;
3. Analyze the economic model through computational methods.

I approach CE as an economist, I am neither a computer scientist nor mathematician! These are just languages...

How to choose a software:

- ...what is your economic problem?
- availability of the software
- maximum accessibility for students and colleagues
- larger web community
- ...the problem, of course, is the possible path dependence and lock-in (now I am in a lock-in with matlab..but...)

Some programs for beginners like me..:

- **Excel** has a nonlinear optimization solver (good for mainstream simple models), it is available to most PCs and 'other Excel' can do a good job too. Unfortunately, it is not well designed for vector matrix operations

- **Matlab** is a high productivity scripting language with fast vectorized operations and a large user base. While MATLAB has many useful routines and libraries, it's starting to show its age..and is also proprietary. And I am going to abandon it, maybe...

Other programs for not-beginners..:

- **Julia vs.Python**: both can be interfaced with low level languages either directly or using existing tools. And both offer high quality just-in-time compilation;
- The focus of **Julia** is bound to scientific applications. Third party libraries are often written entirely in Julia. But It?s still early days for Julia,and Knowledge of Julia is still a niche skill;
- **Julia** is supported by a vast collection of standard and external software libraries. Python has become one of the core languages of scientific computing. For scientific operations, the standard implementation of Python is typically slower than Julia, so additional steps are required to obtain fast execution speeds;
- **R** is a very useful open source statistical environment and programming language, but you can recall in both Julia and Python;
- **C** / **C++** / **Fortran**? Maybe too much for us...

- to solve economic problems (too easy);
- analytical vs. numerical solutions?
- to analyze complex systems (the economic system?);
- to teach economics in a different way (why not);
- to change your economic field (definitely);
- to change Economics

- **A complex adaptive system** is a system in which a perfect understanding of the individual parts does not automatically convey a perfect understanding of the whole system's behavior.
- The approach of CAS is highly interdisciplinary and blends insights from the natural and social sciences to develop system-level models and insights that allow for heterogeneous agents, phase transition, and emergent behavior
- They are **complex** in that they are dynamic networks of interactions, and their relationships are not aggregations of the individual static entities, i.e., the behavior of the ensemble is not predicted by the behavior of the components.
- They are **adaptive** in that the individual and collective behavior mutate and self-organize corresponding to the change-initiating micro-event or collection of events.
- They are a "complex macroscopic collection" of relatively "similar and partially connected micro-structures" formed in order to adapt to the changing environment and increase their survivability as a macro-structure.

- Some of these areas are unique to computational economics: ACE, HAMS, ABM
- while others extend traditional areas of economics: DSGE

- ACE, agent-based computational economics, is a research program that uses computer-based economic modeling;
- it studies dynamic systems of interacting agents;
- there it is an economic adaptation of the complex adaptive systems paradigm.
- the agent (firms, consumers, governments, and so on) refers to ‘computational objects modeled as interacting according to rules’;
- Behavioral equations may be derived from specific optimization problems (optimal rules). AB modellers, however, generally prefer to assume that agents are characterized by bounded rationality: in this case they are ?not global optimizers, they use simple rules (rules of thumb) based on local information? (Epstein).
- No equilibrium condition required (out-of-equilibrium dynamics).
- The aim of the research project is to explain empirical findings, calibrate and validate the models

- A Macroeconomic Agent Based Model(MABM) is a model of the macro-economy as a complex adaptive system in which a multitude of heterogeneous agents (firms, households, banks) interact with each other and the environment.

- Heterogeneity comes from a variety of sources: skill, productivity, size,leverage, wealth...

- Interaction may or may not entail the presence of a network where agents are connected by a partnership, e.g. customer relationships between a firm and a bank

- Aggregate (macroeconomic) variables such as GDP, consumption etc. are computed ?from the bottom up? i.e. summing individual quantities across agents.

- Starting point: consider a population of heterogeneous agents (households, firms, banks,...)

- Theory: write behavioral rules (e.g. demand and supply of goods, labour, credit).

- Codification: translate the rules into code lines

- Validation: calibrate the parameters, run simulations, analyze the emerging properties of the simulated data, both at the cross-sectional level (e.g. firms? size distribution) and at the macroeconomic level (GDP growth and fluctuations, inflation/unemployment trade off), compare these properties with real world ?stylized facts? (e.g. volatility and correlation of macro-variables).

Pros:

- Flexibility;
- No more ad hoc assumptions to get the analytical solution;
- Comparison with true data.

Cons:

- sometimes it could be a black box.

There is of course a compromise, a middle way...

- Zeeman 1974;
- Goldman (1980);
- Frankle and Froot (1986, 1990);
- Day and Huang (1990);
- Chiarella (1992);
- Brock and Hommes (1997, 1998);
- Farmer (2002), Joshi (2002);
- and so on...

Main Elements:

- Group of Agents with different strategies (Fundamentalists, Chartists, Noise traders, Bias traders, Naive)
- Fixed quotas vs. Evolutionary switching mechanisms
- Dynamical systems
- Walrasian Auctioner vs. Market Maker

- first of all, thanks!
- CE is already a growing field with a big community at both national and international level;
- CE can be used for all fields, but you should translate static models in dynamical models; and learn and teach dynamical systems..
- Between a clear analytical solution with poor contacts with true data and a large model with a google of equations ...practically a black hole...there is a different method (methodology?)...