An introduction to ABM

Dr. Claudius Gräbner

July 6, 2017

Johannes Kepler University of Linz, Austria

Motivation

- The story so far:
  - Economies are ...

"... a complex object, every part or component of which is connected with other parts of the same object in such a manner that the whole possesses some features that its components lack that is, emergent properties" (Bunge, 1990).

- Network theory as a language to describe relation
- Dynamics as a language to describe dynamics
- Game theory as a language to describe interactions
- Bringing all together is difficult: requires exact, but flexible language...
  - ...an algorithmic language as 'the right mathematics for the social sciences'

What is ABM?

Agent based modelling

- Formally: a system of difference equations
    - Computers must always discretize
- Definition according to Tesfatsion (2017): *locally-constructive sequential games*
    - Local: interaction between parts is directly modelled
    - Constructive: the model *generates* its output
    - Sequential: agents do things one after another
    - Games: interaction of interested elements takes place
- Helps to avoid assumptions made only for the sake of analytical tractability
- Builds upon 7 modelling principle

Modeling principles in ABM I

1. **Agent definition** Agents are software entities operating within an artificial system
2. **Agent constructivity** Agents make their decision based on time, their internal state, the system state, etc.
3. **Scope of agents** Agents represent not necessarily persons. They may also represent groups, countries, animals, physical entities, etc.
4. **Local constructivity**: The action depends own state at that time.

Modeling principles in ABM II

5. **Agent autonomy**: Agents decide on the basis of their states, i.e. by an external force
    - Although the dependency of an agents from a third force can become a property of the agent
6. **System Constructivity**: The state of the modeled system at any given time is determined by the ensemble of agent states at that time.
    - Note that agents can refer to different components of the system
7. **System Historicity**: Given initial agent states, all subsequent events in the modeled system are determined solely by the interactions among the agents
    - Again, agents might represent very different thing

ABM and the Lucas critique

- Reference point for DSGE and CGE:
  - Stagflation in the 70s
  - Misleading policy advice based on the Cowles Commission approach
- Lucas critique (1976): necessity for sound micro foundations of economic analysis and consideration of expectations
  - An epistemological change w.r.t. to what counts as an ‘explanation’

ABM and the Lucas critique

- Methodological consequence: Use of rational expectations (RE)
  - Focus on RE reduces the degrees of freedom of the complexity reduction function
  - No real micro calibration of assessment of intermediate results, e.g. Euler equation
- Features such as true uncertainty and the corresponding heuristics (Gigerenzer et al.) are excluded from analysis
- ABM as the better answer to the Lucas critique?!

Basics of ABM

ABM, OOP, and Systemism

- To specify agents in an ABM we often exploit the OOP features of Python
- Agents should be defined as a class
- Classes as ‘blueprint’ to create a large population of agent without difficulties
- Thus, you may also have a class `model`
- There is a deeper meta-theoretical connection between ‘systemism’ and OOP
    - $\langle \text{Systems}, \text{Mechanisms} \rangle \leftrightarrow \langle \text{Objects}, \text{Methods} \rangle$

ABM: workflow

1. Define your system under investigation (SUI)
2. Build a conceptual model of the system
3. Translate this model into code
4. Think about initial conditions and run the model
5. Verify and validate the model
6. Analyse your results
7. Interpret the results with regard to the SUI

ABM: Randomness

- Most ABM involve randomness
- Usually, two runs of the ABM do not yield the exactly same results
- To study the model, you need to run it many times and analyse it statistically
    - There is literature and techniques inspired by experimental research to answer questions such as "How many runs are needed"
- Python offers many random number generators, particularly through the numpy

The epistemological benefits

- ABM are open to all kind of model validation...
    1. Input validation
    2. Process validation
    3. Descriptive and predictive validation
- ABM are flexible and may include many mechanisms
- ABM can be related to analytical models
    - Helps to highlight and defend pluralism...
    - ...but also to make it work
        - Relate different research programs to each other
- *Policy readiness*

Main challenges

- ABM are comparatively hard to verify
- Different to communicate, no unique theoretical core
- More freedom on the assumption level necessarily introduces more arbitrary parameters

How to build a model

Some suggestions for model building

- Start with writing comments of what you want to do first
- Do not start with writing classes immediately
  - Test their elements outside a class
- Write a simple interaction, once it works make a look
- Once the '`for i in t` '-loop works, use functions
- As soon as the loop works, build a simulation class '`model`'
  - In the init function, this class should create everything that is needed for the simulation
  - It should feature the following functions:
    - a function '`go`' that goes through one particular time step
    - a function '`run`' that runs the mode for `t` timesteps
    - a function '`save`' that saves/analyzes the results
  - Sometimes it is also useful to write a meta-class '`main`' that calls '`model`' often, stores all results and later calculates summary statistics

Now build an ABM!