# Agent Based Modelling with Applications in Economics

M Czupryna

Cracow University of Economics

07-03-2022

# Overview

# Agent based model

A typical agent-based model has three elements:

- A set of agents, their attributes and behaviours.
- A set of agent relationships and methods of interaction: An underlying topology of connectedness defines how and with whom agents interact.
- The agents' environment: Agents interact with their environment in addition to other agents.

Macal, C. M., & North, M. J. (2005, December). Tutorial on agent-based modeling and simulation. In Proceedings of the Winter Simulation Conference, 2005. (pp. 14-pp). IEEE.
Macal, C. M., & North, M. J. (2006, December). Tutorial on agent-based modeling and simulation part 2: how to model with agents. In Proceedings of the 38th conference on Winter simulation (pp. 73-83). Winter Simulation Conference.

# Agent based model

What is an agent?

- An agent is a self-contained, modular, and uniquely identifiable individual.
- An agent is autonomous and self-directed
- An agent has a state that varies over time.
- An agent is social having dynamic interactions with other agents that influence its behaviour.
- An agent may be adaptive, for example, by having rules or more abstract mechanisms that modify its behaviours.
- An agent may be goal-directed, having goals to achieve (not necessarily objectives to maximize) with respect to its behaviours.
- Agents may be heterogeneous.

# Agent based model

When building an agent based model...

- What specific problem should be solved by the model? What specific questions should the model answer? What value-added would agent-based modelling bring to the problem that other modelling approaches cannot bring?
- What should the agents be in the model? Who are the decision makers in the system? What are the entities that have behaviours? What data on agents are simply descriptive (static attributes)? What agent attributes would be calculated endogenously by the model and updated in the agents (dynamic attributes)?
- What is the agents' environment? How do the agents interact with the environment? Is an agent's mobility through space an important consideration?

# Agent based model

When building an agent based model...

- What agent behaviours are of interest? What decisions do the agents make? What behaviours are being acted upon? What actions are being taken by the agents?
- How do the agents interact with each other? With the environment? How expansive or focused are agent interactions?
- Where might the data come from, especially on agent behaviours, for such a model?
- How might you validate the model, especially the agent behaviours?

# Agent based model

Bottom-up approach (emergence)

- agent based modelling

Top-down approaches

- econometrics
- general equilibrium models (CGE, DSGE)

# Agent based model

Purposes for a model:

- prediction - "By 'prediction',we mean the ability to reliably anticipate well-defined aspects of data that is not currently known to a useful degree of accuracy via computations using the model"
- explanation - "By 'explanation'we mean establishing a possible causal chain from a set-up to its consequences in terms of the mechanisms in a simulation."
- description - "A description (using a simulation) is an attempt to partially represent what is important of a specific observed case (or small set of closely related cases)."

Edmonds Bruce et. al. (2019) Different Modelling Purposes, Journal of Artificial Societies and Social Simulation 22(3) 6, 2019

# Agent based model

Purposes for a model:

- theoretical exploration - "Theoretical exposition' means establishing then characterising (or assessing) hypotheses about the general behaviour of a set of mechanisms (using a simulation)."
- illustration - "An illustration (using a simulation) is to communicate or make clear an idea, theory or explanation."
- analogy - "An analogy (using a simulation) is when processes illustrated by a simulation are used as a way of thinking about something in an informal manner."
- social learning - "A simulation is a tool for social learning when it encapsulates a shared understanding (or set of understandings) of a group of people."

Edmonds Bruce et. al. (2019) Different Modelling Purposes, Journal of Artificial Societies and Social Simulation 22(3) 6, 2019

# Agent based model

Complete Agent-Based Modeling (c-ABM):

1. **Agent Definition** An agent is a software entity within a computationally constructed world, characterized at each instant by its current state (data, attributes,and/or methods), that is capable of affecting world outcomes through expressed actions
2. **Agent Scope** Agents can represent a broad range of entities, e.g., individual life-forms, social groupings, institutions, and/or physical phenomena.
3. **Agent Local Constructivity** An intended action of an agent at any given instant is determined by the agent's state at this instant.

Leigh Tesfatsion, (2022) Agent-Based Computational Economics: Overview and Brief History, Working paper

# Agent based model

Complete Agent-Based Modeling (c-ABM):

1. **Agent Autonomy** All agent interactions (expressed agent actions) at any given instant are determined by the ensemble of agent states at this instant.
2. **System Constructivity** The state of the world at any given instant is determined by the ensemble of agent states at this instant.
3. **System Historicity** Given an initial ensemble of agent states, any subsequent world event (change in agent states) is induced by prior and/or concurrent agent interactions.
4. **Modeler as Culture-Dish Experimenter** The role of the modeler is limited to the configuration and setting of initial agent states, and to the nonperturbational observation, analysis, and reporting of world outcomes.

Leigh Tesfatsion, (2022) Agent-Based Computational Economics: Overview and Brief History, Working paper

# Agent based model

Complete Agent-Based Modeling (c-ABM): Learning

- Reactive reinforcement learning. Roth-Erev reactive reinforcement learning
- Belief-based learning. Fictitious play, Camerer/Ho EWA algorithm
- Anticipatory learning. Q-learning, adaptive dynamic programmming
- Evolutionary learning. Genetic algorithms, genetic programming
- Connectionist learning. Associative memory learning, artificial neural network (ANN) learning, deep learning using ANNs with multiple hidden layers

Leigh Tesfatsion, (2022) Agent-Based Computational Economics: Overview and Brief History, Working paper

# Let us start step by step

NetLogo → Help → NetLogo User Manual

Tutorial 1: Models

File → Models Library → Biology → Wolf Sheep Predation

# Models: Wolf sheep predation model

- Setup
- Go
- Tools $\rightarrow$ Halt
- Speed slider
- Sliders
- Switches
- Plots and monitors
- Settings

# Commands: Traffic basic

File $\rightarrow$ Models Library $\rightarrow$ Social Science $\rightarrow$ Traffic Basic
Please read the model "Info"

# Commands: Traffic basic

File → Models Library → Social Science → Traffic Basic
Please read the model "Info"

Let us go the command window now.
Please try. What is happening?

- observer> ask patches [set pcolor yellow]
- observer> ask turtles [set color brown]
- observer> ask turtles [set pcolor pink]
- observer> ask patches [set color green]

Agent monitor

# Procedures: a new model

- Select “New” from the File menu
- Click the “Add” icon in the toolbar at the top of the Interface tab.
- On the menu next to Add, select Button (if it isn’t already selected).
- Click wherever you want the button to appear in the empty white area of the Interface tab.
- A dialog box for editing the button opens. Type setup in the box labeled “Commands”.
- Press the OK button when you’re done; the dialog box closes.

# Procedures: a new model

- Switch to the Code tab.

**Type the following:**

```
to setup
clear-all
create-turtles 100 [ setxy random-xcor random-ycor ]
reset-ticks
end
```

Hint: random-xcor, random-ycor from NetLogo Dictionary
Reports a random floating point number from the allowable range of turtle coordinates along the given axis, x or y. Turtle coordinates range from min-pxcor - 0.5 (inclusive) to max-pxcor + 0.5 (exclusive) horizontally; vertically, substitute -y for -x.

# Procedures: a new model

- View updates continuous → ticks.
- For Commands enter go instead of setup.
- Check the "Forever" checkbox in the edit dialogue.
- Check the "Disable until ticks start" checkbox too.

**Type the following:**

```
to go
move-turtles
tick
end
```

Hint: tick from NetLogo Dictionary
Advances the tick counter by one and updates all plots.
If the tick counter has not been started yet with reset-ticks, an error results.
Normally tick goes at the end of a go procedure.

# Procedures: a new model

## Type the following:

```
to move-turtles
ask turtles [
right random 360
forward 1
]
end
```

Hint: right (rt) from NetLogo Dictionary
The turtle turns right by number degrees. (If number is negative, it turns left.)
Hint: forward (fd) number from NetLogo Dictionary
The turtle moves forward by number steps, one step at a time. (If number is negative, the turtle moves backward.)
If the turtle cannot move forward number steps because it is not permitted by the current topology the turtle will complete as many steps of 1 as it can, then stop.

# Procedures: a new model

- Try in a Command window: turtles> pen-down

## Adjust set up procedure

```
to setup
clear-all
setup-patches
setup-turtles
reset-ticks
end
```

# Procedures: a new model

## Add

```
to setup-patches
ask patches [ set pcolor green ]
end
```

## Add

```
to setup-turtles
create-turtles 100
ask turtles [ setxy random-xcor random-ycor ] end
```

# Procedures: a new model

## Add

turtles-own [energy]

## Add

to go
move-turtles
**eat-grass**
tick
end

# Procedures: a new model

## Add

```
to eat-grass
ask turtles [
if pcolor = green [
set pcolor black
set energy energy + 10
]
]
end
```

# Procedures: a new model

Create the first monitor:

- Create a monitor by clicking the Add icon on the toolbar, selecting Monitor next to it, and clicking on an open spot in the Interface. A dialogue box will appear.
- In the dialogue type: **count turtles**. Press the OK button to close the dialogue.

And another one...

- Create a monitor by clicking the Add icon on the toolbar, selecting Monitor next to it, and clicking on an open spot in the Interface. A dialogue box will appear.
- In the Reporter section of the dialogue box type: **count patches with [pcolor = green]**.
- In the Display name section of the dialogue box type: green patches
- Press the OK button to close the dialogue box.

# Procedures: a new model

Create the first switch:

- Click on the Add icon on the toolbar (in the Interface tab).
- Select Switch from the menu next to Add.
- Click on an open spot in the interface.
- A dialogue will appear.
- Into the Global variable field, type show-energy? Don't forget to include the question mark in the name.

# Procedures: a new model

## Add

```
to eat-grass
ask turtles [
if pcolor = green [
set pcolor black
set energy energy + 10
]
ifelse show-energy? [ set label energy ] [ set label "" ] ]
end
```

# Procedures: a new model

## Adjust

to go
move-turtles
eat-grass
reproduce
check-death
regrow-grass
tick
end

# Procedures: a new model

## Add

```
to reproduce
ask turtles [
if energy > 50 [ set energy energy - 50 hatch 1 [ set energy 50 ] ] ]
end
```

Hint: hatch number [ commands ] from NetLogo Dictionary
This turtle creates number new turtles. Each new turtle inherits of all its variables, including its location, from its parent.
The new turtles then run commands. You can use the commands to give the new turtles different colors, headings, locations, or whatever. (The new turtles are created all at once, then run one at a time, in random order.)

# Procedures: a new model

## Adjust

```
to move-turtles
ask turtles [
right random 360
forward 1
set energy energy - 1
]
end
```

# Procedures: a new model

## Add

to check-death
ask turtles [
if energy ¡= 0 [ die ] ] end

## Add

to regrow-grass
ask patches [
if random 100 ¡ 3 [ set pcolor green ] ] end