

Notes and links

January 11th

What means economy/economics and what these words mean for you.

Home-work (please send your homework to pietro.terna@unito.it)

—

A close look to the program:

http://fisica-sc.campusnet.unito.it/do/corsi.pl/Show?id=gwpc;sort=DEFAULT;search=;hits=59#a_programma_gwpc

My book.

Epstein, J.M. (2014), *Agent_Zero: Toward Neurocognitive Foundations for Generative Social Science*:

see the Cmap/Readings

—

The importance of our Cmap

—

Leibniz's dream

Leibniz (xi. De scientia universali seu calculo philosophico): ... quando orientur controversiae, non magis disputatione opus erit inter duos philosophos, quam inter duos computistas. Sufficiet enim calamos in manus sumere sedereque ad abbacos et sibi mutuo (...) dicere, calculemus.¹

Calculemus = Simulemus

Artifacts in social science

—

A short paper on models, Simon and complexity, by Pietro Terna.

From the Cmap look at

Terna p. 4, three types of models

Axtell and Epstein p. 6, the [Axtell and Epstein's definition of ABMs](#)

January 16th, 17th, and 18th

homework

—

Agent-based models and programming with the boxes “Agent based models (ABMs) for simulation experiments” and “programming capabilities”

—

—

Cmap novelties

—

A first look to NetLogo

Using NetLogo Starter

The content of the **NetLogo** box in the Cmap

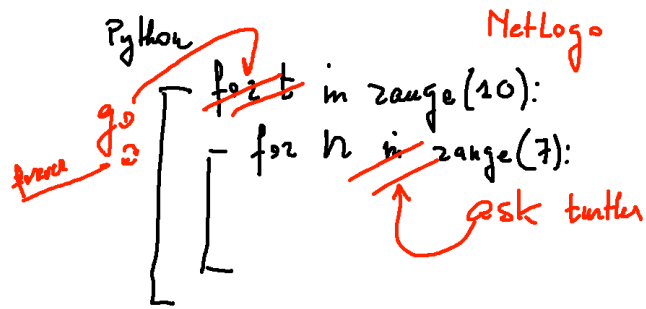
Look at the files in *Programming examples/NetLogo examples* with 20180117a.nlogo, 20180117b.nlogo, 20180118.nlogo

Mainly in 20180118.nlogo have a look to the similarities with natural language

for ? in []:

--..

ask turtles [.....]



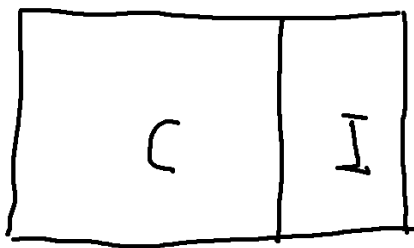
20180117a.nlogo, 20180117b.nlogo

The Go button 'forever' (using also reset-ticks, risk and ticks)

—

Using the box *Programming examples* of the Cmap

National accounting
A rectangular pie.



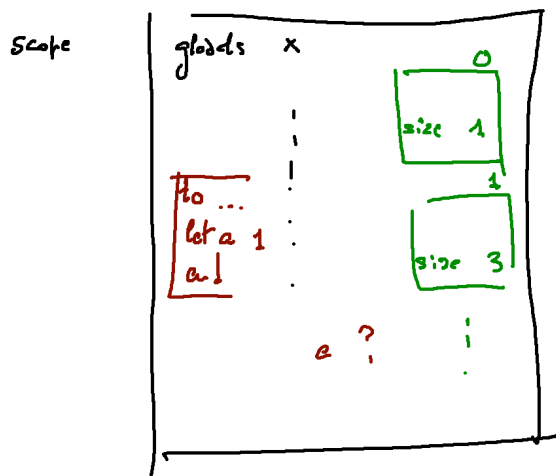
2917

January 23rd, 24th, and 25th

Jan. 30th, Tuesday, **no lesson**

homeworks (from last week and new one)

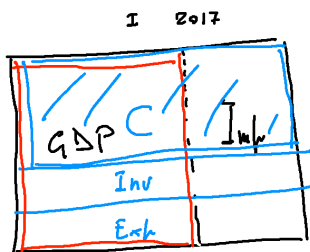
correcting the hw: the scope of the variables (globals and -own)



—

Back to the *pie*

$$GNP = \sum \text{added values}$$



GDP? = Rekonstruierbar
 GDP - C → Savings GDP > C

$$GDP + Imp = C + Inv + Exp$$

$$\underbrace{GDP - C}_{\text{Savings}} = Inv + (Exp - Imp)$$

$$\begin{array}{l} ITA > 0 [?] > 0 \geq 0 \\ Cina \gg 0 \gg 0 \gg 0 \end{array}$$

National accounting, look at the online primer (from the program, at https://www.bea.gov/national/pdf/nipa_primer.pdf, figure at p.3 and table at p. 4)

<http://hdr.undp.org/en/content/human-development-index-hdi>

—

NetLogo: turtles going around and changing their color if they found closely another turtle of a different color (both change to a third color)

Adding a plot

A quick look to <http://terna.to.it/chameleons/chameleons.html>

A quick look to NetLogo Starter in the Camp

—

A first look to Python (3)

The content of the **Python box in the Cmap**

—

familiarize with Python live in the IDLE shell or creating a .py file or via IPython and Jupyter notebook

—

SLAPP <https://terna.github.io/SLAPP/>

with <https://blog.jupyter.org/binder-2-0-a-tech-guide-2017-fd40515a3a84> founded by <https://www.moore.org>

have a look also to <https://bids.berkeley.edu/news/binder-20-has-arrived>

—

on Jupyter with Python, or Julia, or ...
<https://tmp67.tmpnb.org/user/MUKUevWP6RRW/tree>

—

FOLLOWING THE SLAPP TUTORIAL

The folder “1 plainProgrammingBug” in SLAPP

The program “start 1 plainProgrammingBug.py” in Cmap/Programming Examples”

The program “start 1 plainProgrammingBug.nlogo” in Cmap/Programming Examples

~~January 30th~~, January 31st, February 1st

homeworks (from last week and new one)

novelties in the Cmap, Python box

—

again Python 3.6.4 or more

from python.org download python-3.6.4 installer
for **Windows 10** my suggestion is

Download “Windows x86-64 web-based installer”

run it

Install now / Includes IDLE, pip and documentation / Creates shortcuts and file / Install launcher for all users (recommended) / check Add Python 3.6 to PATH

At the end, Disable path length limit

Use IDLE
or launch python from
Windows PowerShell (x86) // Do not use ISE version

python3 here is python

below use **pip** for pip3

—

libraries

[sudo -H] pip3 install numpy

[sudo -H] pip3 install scipy
(contains numpy and F2PY for Fortran)

[sudo -H] pip3 install matplotlib

[sudo -H] pip3 install networkx

for SLAPP

[sudo -H] pip3 install xlrd

grey part for Linux/Mac; Windows users: open PowerShell
]

It is highly useful to familiarize with the Unix-like commands of the Linux/Mac OS Terminal and Windows PowerShell, e.g., at https://en.m.wikipedia.org/w/index.php?title=Command-line_interface&redirect=no and with the DOS-like commands of Command Prompt of Windows, e.g., at <http://pcsupport.about.com/od/termsc/p/command-prompt.htm>

OR, use <https://www.anaconda.com>

====

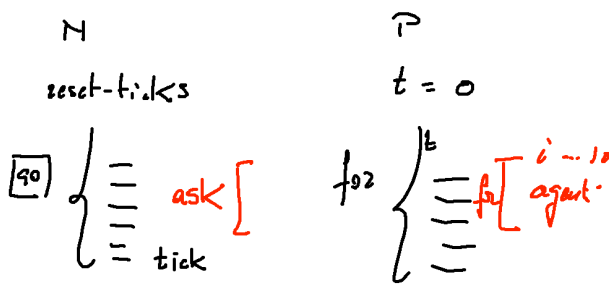
FOLLOWING THE SLAPP TUTORIAL

Again:

The folder “1 plainProgrammingBug” in SLAPP

The program “start 1 plainProgrammingBug.py” in Cmap/Programming Examples”

The program “start 1 plainProgrammingBug.nlogo” in Cmap/Programming Examples



The folder “2 basicObjectProgrammingBug” in SLAPP

Adding classes

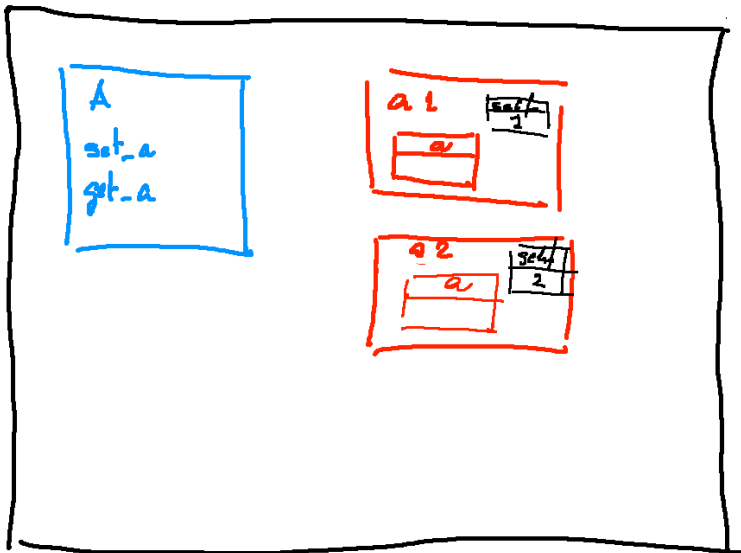
with

start 2 basicObjectProgrammingBug.py

start 2 basicObjectProgrammingBug.nlogo

Class, instances, the *self* concept as a memory address;

as a starting point have a look to the file classes_instances_methods.py, (in the Python repository of the Cmap)



—

The folder “3 basicObjectProgrammingManyBugs” in SLAPP

Adding collections

start 3 basicObjectProgrammingManyBugs.py

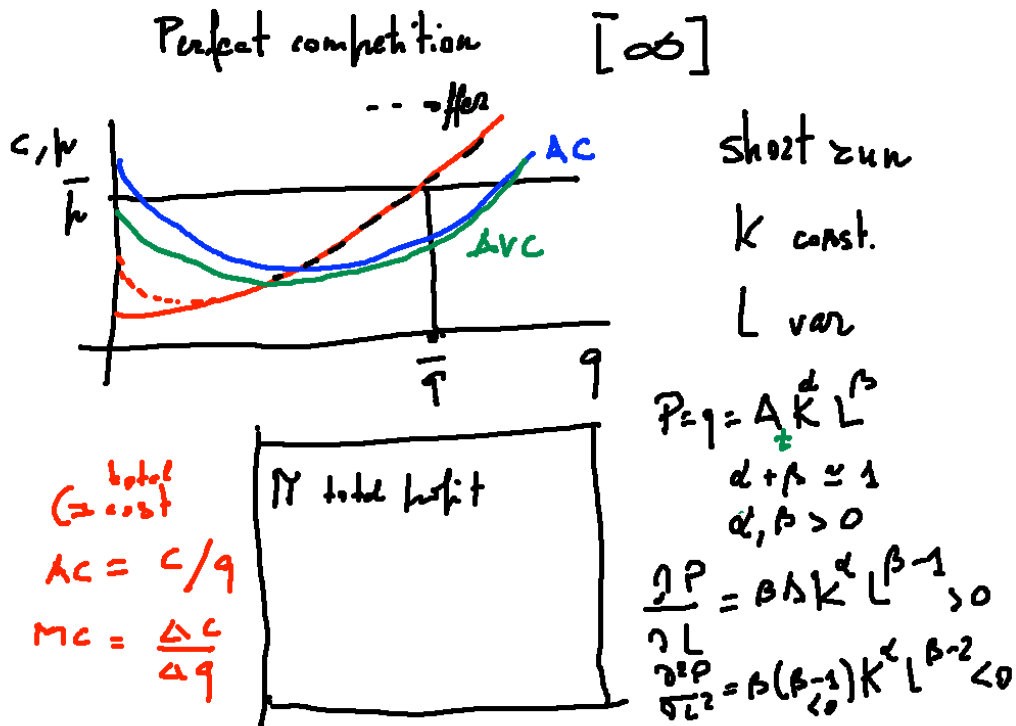
start 3 basicObjectProgrammingManyBugsAppend.py

start 3 basicObjectProgrammingManyBugs.nlogo

—

Economics

Quantity in a perfect competition market



February 6th, 7th, and 8th

Old and current Home-works

<https://inventwithpython.com/blog/2012/07/18/using-trigonometry-to-animate-bounces-draw-clocks-and-point-cannons-at-a-target/>

Novelties in the Cmap

—

copying a matrix and a quick tour of numpy

```
>>> a=[1,2]
>>> aa=a
>>> a[0]=11
>>> aa[0]
11
```

```
>>> a=[1,2]
```

```
>>> aa=a[:]
>>> a[0]=11
>>> aa[0]
1
```

import copy

```
>>> A=[[1,2],[3,4]]
>>> AA=copy.copy(A)
>>> A[0][0]=11
>>> AA[0][0]
11
```

```
>>> A=[[1,2],[3,4]]
>>> AA=copy.deepcopy(A)
>>> A[0][0]=11
>>> AA[0][0]
1
```

—

anyway, use numpy / a look to a few examples

—

(Economy)

again the picture above, to be completed

<http://www.telegraph.co.uk/news/uknews/theroyalfamily/3386353/The-Queen-asks-why-no-one-saw-the-credit-crunch-coming.html>

Mario Draghi at Prometeia 40th year celebration (<http://www.prometeia.it/home>), Dec. 14th, 2015:
<https://www.youtube.com/watch?v=L0FFCRliskyQ>

Daron Acemoglu and Pascual Restrepo. Secular stagnation? the effect of aging on economic growth in the age of automation. 2017.
https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2899142

—

Python/SLAPP folder 4:
basicObjectProgrammingManyBugs_bugExternal+_shuffle.py
Bug.py

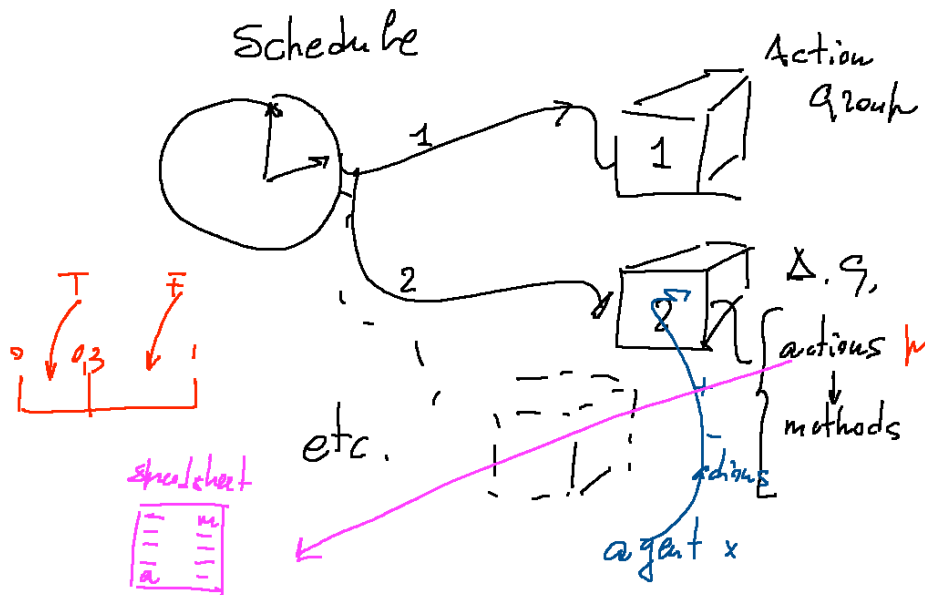
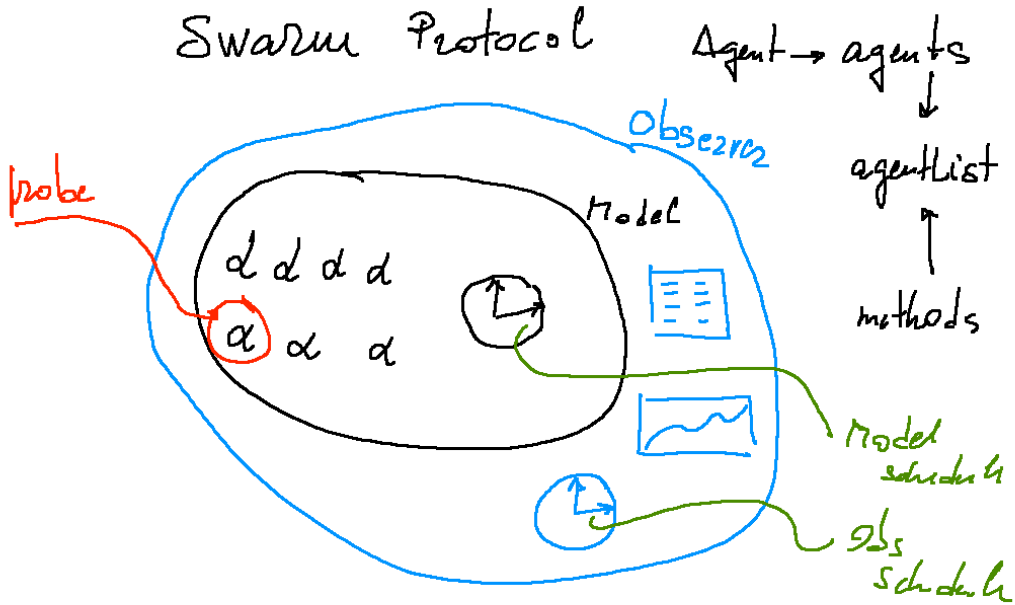
start 4 basicObjectProgrammingManyBugs+_shuffle.nlogo
start 4bis basicObjectProgrammingManyBugs+_shuffle+_ModAttributes.nlogo

breeds.nlogo

—

—

The Swarm protocol and the schedule representation



—

class Turtle and
 turtlesMovingTalking.py
 turtlesMovingTalking_DISPLAYING.py

Economics again

Basics of consumer choice

$$U(x_1, x_2, \dots, x_n) \left| \begin{array}{l} \frac{\partial U}{\partial x_i} > 0 \\ \frac{\partial^2 U}{\partial x_i^2} < 0 \end{array} \right.$$

$$-\bar{R} + p_1 x_1 + \dots + p_n x_n = 0$$

$$L = U(x_1, x_2) - \lambda [p_1 x_1 + p_2 x_2 - \bar{R}]$$

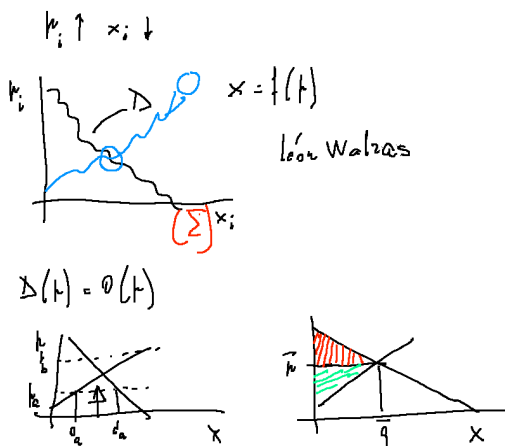
$$\frac{\partial L}{\partial x_1} = \frac{\partial U}{\partial x_1} - \lambda p_1 = 0 \quad \frac{\partial U}{\partial x_1} = \lambda$$

$$\frac{\partial L}{\partial x_2} = \frac{\partial U}{\partial x_2} - \lambda p_2 = 0 \quad \frac{\partial U}{\partial x_2} = \lambda$$

$$\frac{\partial L}{\partial \lambda} \rightarrow \text{constraint}$$

Economics

Price, quantity, global demand



(with Quantity in a perfect competition market, pp. 23–66 of Fenoaltea's book)

February 13th, 14th [with Jacopo Pellegrino], and 15th

novelties in the Cmap

Old and current Home-works

—



Exploring the stock market with

CDA_basic_model.nlogo

Trend agents in CDA: CDA_trend_model.nlogo (also as hint for further works)

<http://www.ilsole24ore.com/art/finanza-e-mercati/2018-02-13/nel-lato-oscuro-mercati-quando-grande-finanza-puo-andare-fuori-controllo-115739.shtml>

<https://www.project-syndicate.org/commentary/stock-market-lessons-of-black-monday-by-barry-eichengreen-2018-02>

—

Python/SLAPP folder: 5 objectSwarmModelBugs

—

<https://terna.github.io/SLAPP/>

projects in SLAPP

SLAPP in terminal, in Jupyter and online (via SLAPP home)

installing the stuff for SLAPP (see above, installing Python)

Jupyter install <http://jupyter.readthedocs.io/en/latest/install.html>

—

February 20th, 21st and 22nd

—

Old and current Home-works

for the current HW, to use online SLAPP, access the running file system at <https://beta.mybinder.org/v2/gh/terna/SLAPP3/master>

then folder 6, basic, try to modify schedule.xls

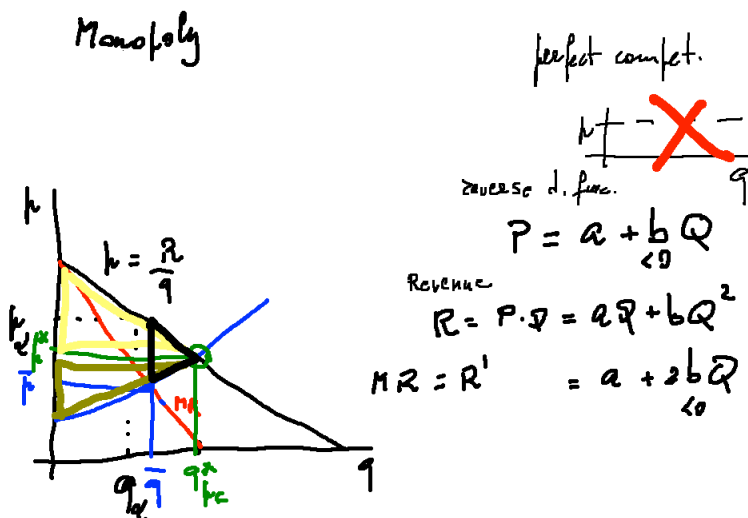
—

Exam

—

Economics

Monopoly



—

—

<http://www.ilsole24ore.com/art/mondo/2018-02-13/amazon-senza-freni-impero-bezos-sfida-limiti-dell-antitrust-162014.shtml>

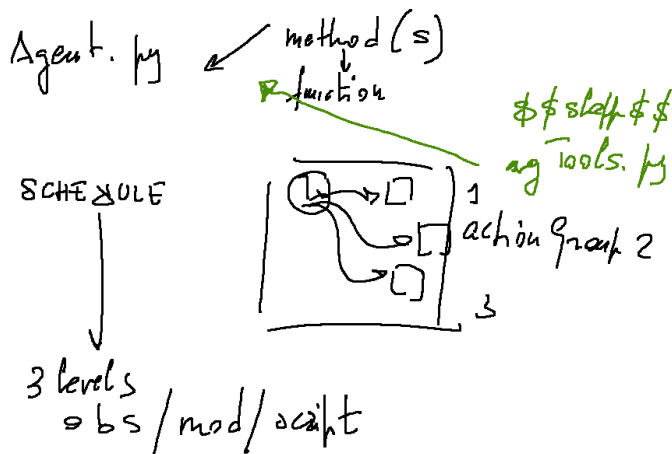
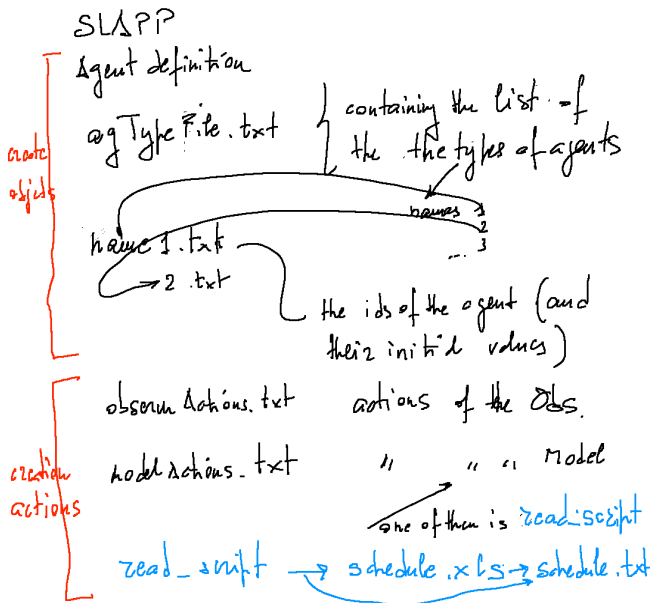
have a look to monopoly.nlogo into the usual repository

—

getting closer to

the running shell of SLAPP, folder

6 objectSwarmObserverAgents_AESOP_turtleLib_NetworkX



—

SLAPP again

The AESOP (Agents and Emergencies for Simulating Organizations in Python) implementation, has the capability of reading the schedule from a text file.

The test file is schedule.txt and its contents are read if we have "read_script" steps activated in the schedule of the model.

If the file is finished, further attempt to "read_schedule" have no effect.

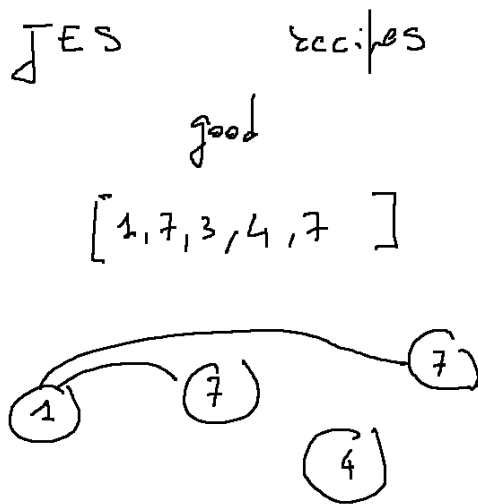
We can read more than one command per step.

If we create a xls file, this is used instead of the txt one.

explore projects:

basic
production

a short explanation about production (which is deeply explained in my 2015 book):



—

related to the production project, a first step in Python with NetworkX <http://networkx.github.io>

aFirstStepWithNetworkX.py

aFirstStepWithNetworkX_&_agents.py

—

a neural network to classify the people of the class

—

basics on artificial neural networks

classification / time series / reinforcement learning

February 27th, 28th, March 1st

a tail from the last argument

<http://terna.to.it/ct-era/ct-era.html>

—

Cmap novelties

Homeworks

—

NetworkX again (modified file: `aFirstStepWithNetworkX_&_agents.py`)

betweenness examples:

`betweennessWithNetworkX_&_agents.py`

`betweennessWithNetworkX_&_agents_large.py`

(later, with NetLogo and NW extension)

a very recent paper on networks <https://arxiv.org/abs/1802.05337>

—

ANN introduction, via the Cmap

—

R (with a few quick introductory steps)

special learning tool for R: `swirl`

```
install.packages("swirl")
```

```
library("swirl")
```

```
swirl()
```

a book about
R Programming for Data Science

examples via the Cmap

`a_R_demand_offer.R`

a first look to ANNa (box Artificial Neural Networks, `annBasics`).

b_nnet_xor.R
c_nnet_1_o.R
d_nnet_2_o.R

—

the ANN of the class
e_people_20180224.zip

(and the old RNA code).

—

markets again:

the **a_R_demand_offer.R model** (in R just to exercise) vs. the **interactingBuyersAndSellers.nlogo** model in NetLogo examples, with the **interactingB&S_pictures.zip** collection

—

R with NetLogo

within NetLogo, from extensions/r/models

try

example1.nlogo

plot-example1.nlogo (JavaGD already installed; missing spatstat)

maybe you have to install rJava too

look at

<https://ccl.northwestern.edu/netlogo/docs/r.html>

—

R with Python

following the Cmap

March 6th and 7th

Novelties in the Cmap

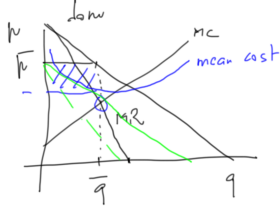
Homeworks: old and new ones

—

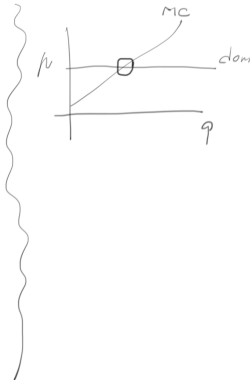
Economics

Monopolistic competition

Monopolistic competition



$$\pi = R - C$$
$$\pi' = R' - C' = 0$$



x

(with pp. 67–88 and 121-128 of Fenoaltea's book)

—

NetLogo and Python

NetLogo extension for Python

<https://github.com/qiemem/PythonExtension>

You need *sbt* (Simple Build Tool) being the package written in Scala you can obtain it from <https://www.scala-sbt.org> (in my Mac installing with Macport:

```
sudo port install ffmpeg
sudo port install sbt
cd in the downloaded folder (/PythonExtension-master) and run
sbt package
```

a new folder will be created with name *py*; copy it into the extensions folder of the NetLogo package

the folder contains a demos with example and a text.txt file with steps to experiment with the extension

added after the conclusion of the course

library to connect to run NetLogo from Python

<http://jasss.soc.surrey.ac.uk/21/2/4.html>

pip or pip3 install pyNetLogo

currently (April 2018) requires Jupyter matplotlib pandas numpy seaborn jpyype

to install jpype
pip or pip3 install JPype1 (vedere <http://konlpy.org/en/v0.4.4/install/>)

currently (April 2018) pyNetLogo works nicely only in Linux

—

NetLogo and the NW extension.

Preliminary steps with NetLogo and Links-verySimpleExample.nlogo.

Preliminary steps with the NW library:

a) ourFirst_NW_Example.nlogo

b) following the Usage section in
<https://github.com/NetLogo/NW-Extension>
with:

- context (our file contexts.nlogo)

- Special agentsets vs. normal agentsets

c) the Network Extension General Demo. You can find it at
<https://github.com/NetLogo/NW-Extension/tree/5.x> (*), downloading the
whole zipped content
or in our repository as
Network Extension General Demo.nlogo

Look at the Info sheet and to its links.

an emerging network: productionWorld.nlogo

finally, have a look to
http://terna.to.it/econophysics18/NetLogo_examples/nw-ext-cheat-sheet.pdf

—

About networks also look at

<http://networkx.github.io/documentation/latest/index.html>

<https://networkx.github.io/documentation/networkx-1.10/reference/algorithms.html>

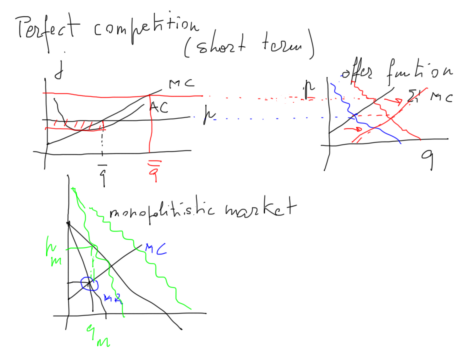
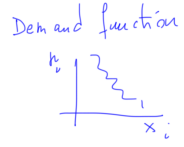
—

Economics

Summarizing

Consume z
 $\max U = f(x_1, x_2, \dots, x_n)$ $\bar{R} - p_1 x_1 - \dots - p_n x_n = 0$

$\frac{\partial U}{\partial x_i} > 0$ $\frac{\partial^2 U}{\partial x_i^2} < 0$ $\frac{\partial U}{\partial u_i} = \lambda$
 p_i



Oligopoly

Duopoly Cournot $P = A - Q$ $Q = Q_1 + Q_2$

$C_1 = C_2 = 0$

$R_1 = P Q_1 = [A - (Q_1 + Q_2)] \cdot Q_1$

$\max R_1 = A - 2Q_1 - Q_2 = 0$

$Q_1 = \frac{A}{2} - \frac{Q_2}{2}$ (guess)

reaction function $Q_2 = \frac{A}{2} - \frac{Q_1}{2}$

Duopoly Stackelberg $P = A - Q$

$R_1 = A Q_1 - Q_1^2 - Q_1 Q_2$

$\text{z.f. } Q_2 = \frac{A}{2} - \frac{1}{2} Q_1$

$R_1 = \frac{A}{2} Q_1 - \frac{1}{2} Q_1^2$ $R_1' = \frac{A}{2} - Q_1 = 0$

$C_1 = C_2 = 0$ $Q_1 = \frac{A}{2}$ $Q_2 = \frac{A}{2} - \frac{1}{2} \frac{A}{2} = \frac{1}{4} A$

Oligopoly model with Marco Mazzoli and Matteo Morini

<https://terna.github.io/oligopoly/>

—

ANN again

a complete example of use of Python/R is timeSeriesNNs.zip
in our repository of Python examples/

—

March 13th

Homework

Novelties in the Cmap (in: Python, Readings and Links)

Novelties in Squibb and Squirks

—

Production function and long run production

Decreasing returns?

Cobb-Douglas $P_t = A L_t^\alpha K_t^\beta$ $\alpha > 0$
 $\beta > 0$
 $\alpha + \beta \stackrel{?}{=} 1$

$$\frac{P_t}{L_t} = \frac{P_t}{K_t}$$
$$\frac{\partial P}{\partial L} = \alpha A L^{\alpha-1} K^\beta \quad \left. \begin{array}{l} \text{idem} \\ K \end{array} \right\}$$
$$\frac{\partial^2 P}{\partial L^2} = (\alpha-1) \alpha A L^{\alpha-2} K^\beta$$
$$\bar{P} = A L^\alpha K^\beta \quad \min C = wL + rK$$
$$\Delta = C + \lambda (\bar{P} - P(L, K)) \quad \frac{\partial \Delta}{\partial L} = w - \lambda \frac{\partial P}{\partial L} = 0$$
$$\frac{\partial \Delta}{\partial K} = r - \lambda \frac{\partial P}{\partial K} = 0$$
$$\frac{\partial P / \partial L}{w} = \frac{\partial P / \partial K}{r} = \lambda$$

$$P = A L^\alpha K^\beta \quad \gamma$$

$$P_2 = A (\gamma L)^\alpha (\gamma K)^\beta = A \gamma^{\alpha+\beta} L^\alpha K^\beta$$

$$P_2 = \gamma P$$

between } of scale \rightarrow constant
 = 1
 < 1 decreasing
 > 1 increasing

(with pp. 90-106 of Fenoaltea's book)

Let us recapitulate from the

beginning, about economics: the role of prices

—

Genetic Algorithms and NetLogo

within the NetLogo 6.2 folder, we have the "Behaviorsearch 6.2.app" for Mac (or proper executables for the other operating systems) and a folder of examples
 a folder of documentation (with a tutorial)

let's see the Example_Flocking_Convergence.bsearch example (via File / Open Example)
 we are using models/Sample Models/Biology/Flocking.nlogo

what if we want to evolve a specific agent, not a model?

—

In Readings and Links,

Jan C. Thiele, Winfried Kurth, and Volker Grimm. Facilitating parameter estimation and sensitivity analysis of agent-based models: A cookbook using netlogo and 'r'. Journal of Artificial Societies and Social Simulation, 17(3):11, 2014. ISSN 1460-7425.

A. Ghorbani, F. Dechesne, V. Dignum, and C. Jonker (2014), Enhancing ABM into an Inevitable Tool for Policy Analysis, in "Policy and Complex Systems", 1(1):61–76, 2014, http://www.ipsonet.org/images/Westphalia_Press/Policy_and_Complex_Systems/Spring_2014/3.%20Ghorbani_et_al_Article.pdf

P. L. Borrill and L. Tesfatsion. Agent-based modeling: The right mathematics for the social sciences? Staff general research papers, Iowa State University, Department of Economics, 2010. <https://econpapers.repec.org/paper/isugenres/31674.htm>

Links in Reading and Links to the

Centre for Research in Social Simulation (Nigel Gilbert)

and to the

Agent-Based Computational Economics web site (Leigh Tesfatsion)

—

DSGE

https://en.wikipedia.org/wiki/Dynamic_stochastic_general_equilibrium
browsing it ...

The virtues and vices of equilibrium and the future of financial economics
COMPLEXITY Special Issue: Econophysics
Volume 14, Issue 3, January/February 2009, Pages: 11–38, J. Doyne Farmer and John
Geanakoplos
<http://onlinelibrary.wiley.com/doi/10.1002/cplx.20261/abstract>

Also interesting the whole number of Complexity, on Econophysics
<http://onlinelibrary.wiley.com/doi/10.1002/cplx.v14:3/issuetoc>
also in readings and links

Giorgio Fagiolo and Andrea Roventini. Macroeconomic Policy in DSGE and Agent-Based Models. *Revue de l'OFCE*, (5):67–116, 2012
http://www.cairn.info/load_pdf.php?ID_ARTICLE=REOF_124_0067

—

As a (controversial) conclusion
Mauro Gallegati, Steve Keen, Thomas Lux, Paul Ormerod
Physica A 370 (2006) 1–6
Worrying trends in econophysics
<http://www.sciencedirect.com/science/article/pii/S0378437106004420> (from Unito network)

browsing it ...

—

The future about work: a jobless society?

—

a special conclusion
<http://www.artspecialday.com/9art/2017/09/27/rivelazione-montale-non-chieder-ci-la-parola/>

squib & quirks

Alibaba and Microsoft AI beat humans in Stanford reading test
to read it, google the title above

—

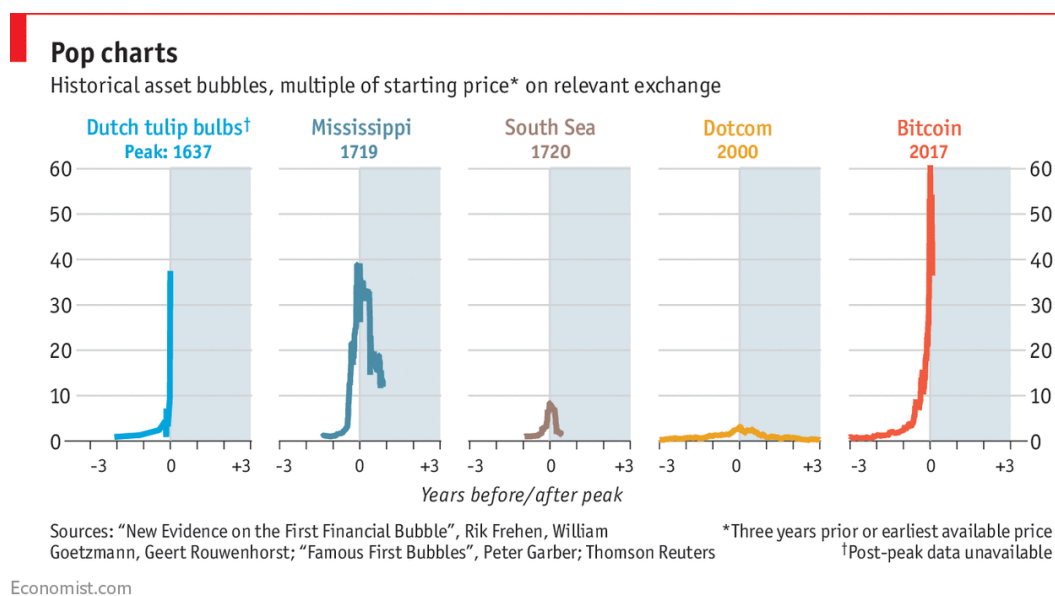
<https://github.com/Zeta36/chess-alpha-zero>

—

https://welcome.linode.com/writethedocs/?utm_source=writethedocs&utm_medium=banner&utm_campaign=writethedocs

Economist January 22nd, 2018 (Blog Graphic Detail):

<https://www.economist.com/blogs/graphicdetail/2018/01/daily-chart-14>



Krugman on Bitcoin <https://www.nytimes.com/2018/01/29/opinion/bitcoin-bubble-fraud.html>

<https://medium.com/applied-data-science/how-to-build-your-own-alphazero-ai-using-python-and-keras-7f664945c188>

<https://github.com/eriklindernoren/NapkinML>

random paper generator <https://pdos.csail.mit.edu/archive/scigen/>

AI and the future of work

Slides at <http://terna.to.it/accademia/>

<https://medium.com/intuitionmachine/deep-learning-the-unreasonable-effectiveness-of-randomness-14d5aef13f87>

<https://www.quantamagazine.org/scant-evidence-of-power-laws-found-in-real-world-networks-20180215/>

<https://arxiv.org/abs/1801.03400>

Spyder as IDE <https://github.com/spyder-ide/spyder>

<http://www.linuxjournal.com/content/introducing-spyder-scientific-python-development-environment>

Matplotlib <https://realpython.com/blog/python/python-matplotlib-guide/>

<https://jupyter.org/try>

https://en.wikipedia.org/wiki/Agent-based_model

other deep learning tools

<https://mxnet.apache.org>

<http://pytorch.org/about/>

connecting different environments and codes with <https://redis.io>