Computational Economics: Homework

October 25, 2013

9 HW9

Exercise 1
For Prof. Mathy's lecture, read `http://www.ericswanson.us/papers/pert.pdf`. (I placed a copy in the `paper_presentations` folder.) Download PerturbationAIM from `http://www.ericswanson.us/perturbation.html`, and the first few examples as well. Place these into a folder where other files for Tuesday's lecture can also reside. Make sure you can import PerturbationAIM, following the instructions on the download site.

Computational Exercise 1
Read Blinder (1973). Provide an agent-based version of the Blinder (1973) model. Represent an agent as a location in a list (of wealths), one list for males and one list for females.

I have provided a template for you in the hw/blinder folder. Note that your own work should use such templates. Each student should write the body of one of the function templates. Each student should write a unit test for one of the implemented functions, ideally a function written by another student.

Computational Exercise 2 (Simple Simulation)
This is finally starting to look better. Remember, every student must contribute to this. I see good contributions by most students. (Thanks!) Now I would like each student to replicate one of the charts in this spreadsheet.

Computational Exercise 3 (Change Savings Rate)
Note that there is still work to do on the Solow-Swan simulation for change in the saving rate. See HW 5.6. Look in the file for hints, searching on `ai:`.

Exercise 2
You need to learn LaTeX if you will be doing research in computational or agent-based economics. Start with `http://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf`. Use LaTeX for your presentations, your class paper, and your dissertation: `https://subversion.american.edu/aisaac/authesis/authesis.htm`. (Not required.)

8 HW8

Exercise 3
Required readings for Tuesday:
For Boris Gershman's presentation read Ashraf et al. (2012) (available at `http://www.cairn.info/landing_pdf.php?ID_ARTICLE=REOF_124_0043`). For Jeff Levy's presentation read Salerno et al. (2011) (in the repository).

Recommended readings for Tuesday:
For Boris Gershman's presentation read Ashraf et al. (2011) at `http://www.nber.org/papers/w17102.pdf`.

Exercise 4
Here are some readings that I think meet Jeff's request last night: `https://subversion.american.edu/aisaac/syllabi/796syl.xhtml#introductory-macro-readings`. Please feel free to request more specific readings.

Computational Exercise 4
This exercise is intended to be a lesson in not being timid. You have learned some minimal C++. Most of you will be interested in being able to read C++ rather than in writing it, and the basic tutorial you have done allows you to do that. But if you want to replicate code written in C++, you should also be able to compile it the code you get your hands on. And if it is older code, you expect to run into some small problems. Here is an example of how simple it can be to overcome those. (You should do this exercise, not just read it!) By the way, Zidong will cover the paper that goes with this code on October 29. (Howitt, Peter Macroeconomics with Intelligent Autonomous Agents, In Macroeconomics in the Small and the Large: Essays on Microfoundations, Macroeconomic Applications and Economic History in Honor of Axel Leijonhufvud, edited by Roger Farmer. Cheltenham: Edward Elgar, 2008.)

Open CodeBlocks. Pick File/NewProject/EmptyProject and click Go. Answer the "wizard" questions by entering the title: `ABM_UCLA`. You can just pick the debug configuration.

Look in the new `ABM_UCLA` folder you just created, you will see a single file: `ABM_UCLA.cbp`. This is a CodeBlocks project file, containing some simple information about your (currently empty) project. You can look at it with a text editor if you wish, but that is not necessary.

Using the filename `ABM_UCLA.cpp`, copy `http://www.econ.brown.edu/fac/Peter_Howitt/publication/ABM_UCLA.txt` into the new `ABM_UCLA` folder you just created. Back in CodeBlocks, use Project/AddFile to add this new file. Back in CodeBlocks, use File/Open to open this new file. (If you are not on Windows, you probably need to change `#include <conio.h>` to '`#include <ncurses.h>`, which should be available on Linux and Mac (BSD) platforms.) Pick Build/Build to try to compile it.

Bummer! It will not compile. We see the problem is the `__max` and `__min` functions, which were not provided. However, equivalents are available in the C++ standard library. Let us make the following changes.

1. Add an inclusion: `#include <algorithm>`.
2. change `__max` to `max` and change `__min` to `min`
3. Just below all the includes, add the line: `using namespace std;` (and don't forget the semicolon!).

Report to the class list what happens. (Be as specific as possible. Note I am asking you just to report the problem, not to fix it. I'll walk you through how to fix it once we all see the problem.) Bummer! It still will not compile. But this is just because `max` does not know how to compare a float to the int `0`. (This isn't Python, remember!) Just change that 0 to 0.0, and it compiles!

If you look in your project folder you will find new `obj` and `bin` folders (see the tutorial again if this surprises you), and in the `bin/Debug` folder your will find the binary executable file. (E.g., on Windows this is `ABM_UCLA.exe`.)

You may have noticed one warning: unknown escape sequence '\A'. This is a (luckily) harmless typo that you can fix as '\nA' if you wish.

OK, go ahead and run your new executable! (Warning: it may run for a couple hours.)

Computational Exercise 5
A little more work on NFR chapter 3. (You should really be mastering this chapter by now.) Translate the code in `nfr_ck_c_transition\transition.py` from Matlab to Python. Please feel free to discuss this model on the class list.

References

Ashraf, Q., B. Gershman, and P. Howitt (2011). Banks, market organization, and macroeconomic performance: An agent-based computational analysis. NBER Working Paper 17102, National Bureau of Economic Research.

Ashraf, Q., B. Gershman, and P. Howitt (2012). Macroeconomics in a self-organizing economy. *Revue de l'OFCE 5*(124), 43–65.

Salerno, J., S. J. Yang, D. Nau, and S.-K. Chai (Eds.) (2011). *Rebellion on Sugarscape: Case Studies for Greed and Grievance Theory of Civil Conflicts Using Agent-Based Models*, Volume 6589 of *Lecture Notes in Computer Science*. ISBN: 978-3-642-19655-3.

Past Assignments

7 HW7

Exercise 7.1 I should see a well developed project outline at this point in the repository. (See last week's assignment.) Recall that you have been asked to use LaTeX for your outline now. (See last week's assignment.) This can raise the question of how to best manage your bibliography. Try Jabref: `http://jabref.sourceforge.net/`.

Sketch of Exercise Answer:

Exercise 7.2 Some recommended readings:
Ray Fair reflects on macro modeling: `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2323549`
`http://bibliotecadigital.fgv.br/ojs/index.php/rbe/article/view/2619`
Streamlined dynamic programming notes: `http://www.eui.eu/Personal/Researchers/Gimber/notes/dynamic_programming.pdf`

Sketch of Exercise Answer:

Exercise 7.3 Please review class creation and subclassing:
`http://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial/Classes`
`https://subversion.american.edu/aisaac/notes/python4class.xhtml#defining-classes`
Explain the meaning and use of `self` as encountered in these readings.

Sketch of Exercise Answer:

Computational Exercise 7.1 Remember to install Code Blocks before next class: `http://www.codeblocks.org/`. You need some basic familiarity with compiling C and C++ for this class. Please read this tutorial: `http://www.cplusplus.com/files/tutorial.pdf`. (Be sure to work the examples as you go along, and post questions as you run into difficulties.)

Computational Exercise 7.2 (Simple Simulation) There is still much that has not been done in `simple_simulation.py`. Recall that I asked you to use Python to read Excel data and confirm that you are correctly replicating `simple_simulation.xls`. Since nobody really addressed this, I have placed illustrative code for the first NFR random walk in `simple_simulation.py`. Please produce simular checks for the rest of their series. (*Each* student should replicate and verify one series. Keep in mind our previous discussions of replication problems!)

Discussion:

Exercise 7.4 (Solow Growth) You should have mastered NFR chapter 2. Show how to formulate the discrete-time Solow-Swan model as a simple recurrence in k. (Use our `Recurrence01` class.)

Sketch of Exercise Answer:

Exercise 7.5 (Optimal Growth) Finish reading NFR chapter 3.

Read my posted dynamic programming notes. Work the value function iteration by hand (using Mma is fine). (The notes go through this, but after reading it once, try it without looking at the notes.)

Sketch of Exercise Answer:

6 HW6

Exercise 6.1 (Python Classes) Read `http://www2.lib.uchicago.edu/keith/courses/python/class/5/`, `http://en.wikibooks.org/wiki/A_Beginner%27s_Python_Tutorial/Classes` and `https://subversion.american.edu/aisaac/notes/python4class.xhtml#defining-classes`.

Exercise 6.2 (Optimal Growth) You should have mastered NFR chapter 2.

Start reading NFR chapter 3.

Read Sargent and Stachurski on optimal growth: `http://quant-econ.net/dp_intro.html`.

Students desiring more theoretical detail can try the Stokey and Lucas text, on reserve. (Our course focuses on computation rather than theory.)

Sketch of Exercise Answer:

Exercise 6.3 Move the project outline you have developed into a `.tex` document in your personal folder, and the this `.tex` document to the repository. Your bullets should now become the section, subsection, etc. headers, and you should begin to fill in text for your paper. Include a `\tableofcontents` command at the beginning of the document, which will then reproduce your outline for you.

In `notes.txt`, leave the outline header, but add a note that it's been moved to `<filename.tex>` in the same folder.

Comment: Your paper should be starting to develop now. Your literature review should show the directions you are pursuing. (You can always revise it.) I should start seeing code related to your paper in our repository.

Sketch of Exercise Answer:

Computational Exercise 6.1 The replications we have been working on still have many comments in them that you have not responded to. Let's make them beautiful. (E.g., steady state computations should always use a function. Utiltity computations should always use a function. Iteration should use an iterator when possible. Graphs should be beautiful. (Some of these have been much improved!) Graph code should use functions when possible.

Computational Exercise 6.2 Recall our discussion of iterators (`https://subversion.american.edu/aisaac/notes/iterators.pdf`). Here is an abstract representation of a first-order recurrence, as an iterator. You can find it in `nfr_utils/utils.py`.

```python
class Recurrence01(object):
    """Provides an abstract first-order recurrence, as an iterator.
    (Subclass and implement `knext` using only `self.k`.)
    """
    def __init__(self, k0=0):
        self.k = k0
    def __iter__(self):
        return self
    def next(self):  # Python 2
        self.k = k = self.knext()
        return k
    def knext(self):
        raise NotImplementedError
```

Note that you cannot use this class directly: it is *abstract*, in the sense that it has a method that is not implemented. You use this class by subclassing it and immplementing the `knext` method. (Reread `http://www2.lib.uchicago.edu/keith/courses/python/class/5/#inheritance` on subclassing.) Here are some exercises to try with this class. (Try one of these exercises, and let someone else try another.)

In simple_simulation.py, show how to produce a random-walk class. (You will create the shocks in your knext implementation.)

In simple_simulation.py, show how to produce a AR1 class. (You will create the shocks in your knext implementation.)

In solow_deterministic.py produce the discrete-time path of the capital stock.

Computational Exercise 6.3 Change Recurrence01 to accept an exogenous iterable. Call the new class Recurrence01x. (Note that this class should also be abstract. That is, knext should not be implemented.)

Computational Exercise 6.4 Look in hw/resources for ai_CK_c_change_structural_parameters.m. Copy this to nfr_CK_c_change_structural_parameters/structural_change.py. Translate the Matlab code to Python. (Use the techniques you have been learning in this course.)

Once again, it will be important to *communicate* on the email list, to avoid simultaneous work on overlapping parts of the code. (E.g., only one person should add this file!! Otherwise, you will have to learn how to resolve "tree conflicts" in Subversion.

5 HW5

URGENT: Some students have little in notebook.txt. Be sure to respond to my notes in your notes.txt file. At this point you should have a substantial outline in your notes.txt. It is time to see the papers you are reading discussed in your notes.txt file.
URGENT: Some students still do not have an adequate level of participation in the HW. Participation is a *large* graded component of the course. (See the syllabus for details.)

Computational Exercise 5.1 (Multiprocessing) If you have not done last week's multiprocessing homework, do it now. *Every* student is expected to do this.

Computational Exercise 5.2 (Dynamic Inefficiency) In the folder hw\nfr_dynamic_inefficiency is a file dynamic_inefficiency.py. This is just NFR's change_savings.m with a new name. Convert this to Python code. (I especially want contributions from last week's laggards. Talk to me if you are finding this difficult, so that I can suggest resources.)

Computational Exercise 5.3 (Simple Simulation) Now that you know how to use Python to read Excel data, use that facility to confirm that you are correctly replicating simple_simulation.xls.

Discussion:

Computational Exercise 5.4 (Simple Simulation) Create a generator that can produce any AR2 process. Use it to generate all the series in simple_simulation.xls from the provided white-noise shocks.

Discussion:

Computational Exercise 5.5 Recall the Solow-Swan fundamental dynamic equation $\dot{k} = Ask^{\alpha} - (d + g_{\Gamma} + g_N)k$. Using the NFR parameter's, solve this numerically. (If you are programming in Mathematica, use NDSolve. If you are programming in Python, use odeint.) Let your initial value of k be the steady-state when $s = 0.2$, and solve for the path with $s = 0.3$.

Computational Exercise 5.6 (Change Savings Rate) Steady state computation: NFR compute SS values identically twice (and then again in other files) except for changes in parameters. This calls for a function! Your function sskyc should accept a dictionary of parameter values and return a dictionary of steady-state values (kss,yss,css).

NFR compute utility values identically twice (and then again in other files). This calls for a function! Your function `u` should accept two arugments (c and sigma) and return the resulting utility.

4 HW4

Note: I am repeating here some HWs that nobody tackled or that remain incomplete.
URGENT: Some students have nothing in notebook.txt (not even quality control partners)! Be sure to respond to my notes in your notes.txt file.
URGENT: Some students do not have an adequate level of participation in the HW. Participation is a *large* graded component of the course. (See the syllabus for details.) If you are finding it difficult to contribute, discuss this on the class email list. If you would like to see the extent of your own contributions, you can to an `svn log` on the `hw` folder.
URGENT: It is time to see the papers you are reading discussed in your `notes.txt` file. Your outline should be well-along at this point. Be sure to respond to comments in `notes.txt` (including mine) in a timely manner.

Computational Exercise 4.1 (Simple Simulation) Justin proposed using an object for the stochastic processes. What are the advantages and disadvantages of doing this?

Read `http://www.ibm.com/developerworks/library/l-pycon/index.html`. In Python, what is an iterator? How do you create a Python iterator object? Can an iterator usefully represent a random walk?

In Python, what is a generator function? Could generator functions be used to produce our shocks? Can a generator function usefully represent a random walk?

Discussion:
See my posted notes on Python generators.

Computational Exercise 4.2 (Solow Stochastic) Implement Jelena's suggestion for parameter handling.

Discussion:
See `params.py`.

Computational Exercise 4.3 (Change Savings Rate) In the folder `hw\nfr_change_savings` is a file `change_savings.py`. This is just NFR's `change_savings.m` with a new name. Convert this to Python code. (I especially want contributions from last week's laggards. Talk to me if you are finding this difficult, so that I can suggest resources.)

Computational Exercise 4.4 (Multiprocessing) Answer the HW3 multiprocessing question.

Also, each student should make an `hpc` folder in their personal folder. Commit it to our Subversion repository. Copy one of Jelena's example scripts into this hpc folder and commit it. Add your own comments explaining how this script works. (If you are experienced, choose a script that teaches you something about multiprocessing.) Add and commit an `.lsf` script to run this script on the HPC.

On the HPC, checkout your hpc folder as svn796. Submit your `.lsf` file (using `bsub`).

Report any problems or confusions.

Computational Exercise 4.5 (Simple Simulation) Continuing with the spreadsheet ...

Spreadsheet Precision Get the white noise series from the spreadsheet in two ways. First, do the best you can to get it out of Excel, using only Excel facilities (including copy and paste), saving as CSV, etc. Second, use `xlrd` to read the data from your worksheet. Compare the two series. Are they the same?
Hints: i. Python indexing is zero-based. ii. The following may be useful: `http://www.youlikeprogramming.com/2012/03/examples-reading-excel-xls-documents-using-pythons-xlrd/`

Computational Exercise 4.6 (Object Oriented Programming) Read `http://quant-econ.net/python_oop.html`. Propose an object-oriented approach to descriptive statistics for a time series. Use it to desribe the white-noise series from the previous exercise.

3 HW3

Exercise 3.1 (Project) I should be seeing entries in your notes.txt file by now! Start working on an outline and taking notes on your readings. Feel free to discuss you project on the class email list. Also, list the papers you are reading, and comment on what you find in them relevant to your project.

Exercise 3.2 (Quality Control) Each student needs two (2) quality control partners for their project. Each student should be a quality control partner on two (2) projects. Record the names of your quality control partners in your notebook.txt.

Computational Exercise 3.1 (Solow-Swan (Continuous)) Get NFR's `solow_stochastic.m` file. Translate this Matlab code into Python+NumPy. (Actually, I put comments in `solow_stochastic.py`, in the HW folder. So you can start there.)

Replicate the exact solution for the continuous time model. Discuss any replication issues issues on the class list. (I'd like to see some code added to the dummy files I put in the hw folder.)

Web Resources:
Matlab intro: `http://rt.uits.iu.edu/visualization/analytics/math/matlab-getting-started.php`
Matlab to Python translation: `http://wiki.scipy.org/NumPy_for_Matlab_Users`.

What are some of the translation (Matlab to Python) issues that you have encountered? Where would you like to change the approach to the code?

Discussion:
Translation issues:

- Different notation for exponentiation.
- Slightly different `for` loop syntax.
- Different notation for appending values (and list concatenation)

Let us consider the last one a bit more. One you notice how Matlab appends values, you should ask how you would like to accumulate values. (And whether you actually need to accumulate them.) Here are some considerations.

- List convenience (e.g., I don't have to know the list size ahead of time)
- Array convenience (e.g., I may want to do array operations on my data)
- Memory consumption (except for very small arrays, arrays will use less memory)
- Memory consumption (whether I really need to create the whole list or whole array depends on the application)
- speed (pre-allocating and filling arrays should be faster than growing lists, but ... test it!)

Computational Exercise 3.2 (Solow-Swan (Continuous)) NFR provide a parameter set in their Matlab file. It would be nice to be able to handle parameter changes in a clear and transparent manner. (Going into a file and changing the value, and then later hoping to change it back, is neither clear nor transparent.) Make suggestions about how to approach this problem.

Computational Exercise 3.3 (Shock Process) In `solow_stochastic.m` we find the following:

```
for t=1:T,
    thetat = exp(rho*log(theta0)+innov(t));
    theta0 = thetat;
    theta = [theta;thetat];
end
```

What is this code doing? What is a Python equivalent? Can you suggest alternative approaches?

Discussion:

This is an implementation of their shock process on p.96.

Computational Exercise 3.4 (Multiprocessing) Run a couple of Jelena's example scripts. (I have copied them to `project/econ796/mp`.) Discuss what you learn on the class list.

Comment: Jelena notes that you should either i. run the code from your home directory (e.g., copy it and then run from 'home') or at least direct the stdout/sterr files to your home directory.

The documentation says: "Note: Functionality within this package requires that the `__main__` module be importable by the children." What are implications of this?

Computational Exercise 3.5 (Simple Simulation) Continuing with the spreadsheet ...

Spreadsheet Precision Get the white noise series from the spreadsheet in two ways. First, do the best you can to get it out of Excel, using only Excel facilities (including copy and paste), saving as CSV, etc. Second, use `xlrd` to read the data from your worksheet. Compare the two series. Are they the same?

Hints: i. Python indexing is zero-based. ii. The following may be useful: `http://www.youlikeprogramming.com/2012/03/examples-reading-excel-xls-documents-using-pythons-xlrd/`

Computational Exercise 3.6 (Object Oriented Programming) Read `http://quant-econ.net/python_oop.html`. Propose an object-oriented approach to descriptive statistics for a time series. Use it to desribe the white-noise series from the previous exercise.

2 HW2

Reminder: everyone should have `notebook.txt` and `notes.txt` files under version control at this point. (A couple of you do not. Let me know if this is a Subversion problem.)

Computational Exercise 2.1 (Simple Simulation) Continuing with the spreadsheet ... (I'd like to see some code added to the dummy files I put in the hw folder.)

Impulse Response What is an impulse response? How do NFR produce their impulse responses? Replicate their impulse response graph.

Computational Exercise 2.2 (Solow-Swan (Continuous)) Get NFR's `solow_deterministic.xls` workbook and replicate the exact solution for the continuous time model. Discuss replication issues issues on the class list. (I'd like to see some code added to the dummy files I put in the hw folder.)

1 HW1

Computational Exercise 1.1 (Simple Simulation) Go to the NFR website, which is currently at `http://pendientedemigracion.ucm.es/info/ecocuan/anc/Growth/growthbook.htm` and decompress the files for ch.1. Attempt to replicate the "Simple Simulation.xls" workbook. You may use Python or Mma. Feel free to discuss your efforts on this list, including a discussion of strategies you might take to this replication. (Also, feel free to criticize their work.)

Items for discussion (on list or in classroom):

Best Practice What kind of changes should you make to this spreadsheet as you try to understand it?

Discussion:
None. In fact you should make it a "read only" file. If you want to make changes, make a copy of the file, and keep the original.

Purpose What is the purpose of this spreadsheet?

Discussion:
To present simulations of some simple stochastic processes.

Time Index How do you create the time index $[0, 200]$ in Excel? In Python? In Mathematica?

Discussion:
There are always multiple approaches. In Excel, you can enter a 0 in a cell, go to the `Home` tab, click Fill, and enter a linear series in the column. In Python you can let `time_index = np.arange(201)`. In Mathematica you let `timeIndex=Range[0,200]`.

Replication As soon as you start thinking about column B, what barriers do you see to replication? How might you deal with them?

Discussion:
On the one hand, by not providing the seed NFR hinder replication. On the other hand, we can just copy the column from the spreadsheet. (However, copying at the true precision rather than the displayed precision is trickier.)

White Noise Column B is labeled "White Noise". Can you tell how NFR produced this column? Does it look to be "white noise"? What else can you tell about it?

How would you similarly draw 200 times from $N(0, 1)$ in Mathematica and in Python?

How would you produce the two columns (M and N) in Panel 4 in Mathematica and in Python? The data from these columns was saved as CSV to wn.txt (in the repository). The data in columns M and N are in the last two columns of wn.txt. Produce your replicated values and compare. Do you get a match? (If not, what do you think is going wrong?)

Roughly reproduce the "White Noise with Constants" chart (in Mathematica and in Python). (Include a legend and label the horizontal axis.)

Discussion:
Right below the cells saying "White Noise" is a cell saying $N(0, 1)$, so the spreadsheet has some information. It would be better to add a comment to this cell (Review ¿ New Comment) that provides details of how the values were generated. The NFR book does provide some details (on page 9), however they do not provide a random seed (which we need for replication).

To produce Panel 4, just add the appropriate constants (3 and 5) to your white noise series.

Random Walk What is a random walk? In Panel 2, how are the random-walk time series constructed? What is the difference between the first colum on Panel 3 and the second column of Panel 2?

What is a random walk with drift? Try to reproduce Panel 5. Do you run into any problems? If so, suggest ways to investigate the source of the problem.

Roughly reproduce the "Random Walk with Drift" chart (in Mathematica and in Python). (Include a legend and label the horizontal axis.)

Discussion:
If you were very careful, you noticed these two errors: their cell P12 uses 0.1 instead of 1.0, and their cell S12 uses 0.5 instead of 5.0.)

Summary Statistics The workbook creates 19 difference series and reports statistics for most of them. Try to replicate the series and the statistics. (Create a function to report the statistics for any given series.)

Mathematica Discussion: To load `wn.txt`, use Mathematica's `Import` command. Before creating your charts read `http://reference.wolfram.com/mathematica/guide/DataVisualization.html` and `http://reference.wolfram.com/mathematica/ref/ListPlot.html`.

Python Discussion: To load `wn.txt`, use NumPy's `loadtxt` function. To compare roughly equal series, use NumPy's `allclose` function. Before creating your charts read `http://matplotlib.org/faq/usage_faq.html` and `http://scipy-lectures.github.io/intro/matplotlib/matplotlib.html`. I find it is often useful to be fairly explicit about the plotting process: `http://econpy.googlecode.com/svn/trunk/software4econ.xhtml#mpl-hints`.