Economic Growth and Sustainable Development

Assignment 2025

The assignment (worth 20% of the final grade) consists of replicating the key graphs from the lecture notes and tutorial exercises using Python (or R). This requires learning the material beforehand to be able to replicate the graphs. So long as you read the material and work through the codes I provided, the assignment should be straightfowards. You will replicate the graphs while working with the same colleagues you are currently working with for the projet d'éco (marks will be equally weighted). The assignment is to be submitted no later than the 28th of March.

To calibrate the models, use the following parameterization:

Cobb-Douglas production function $F(K_t, A_t L_t) = K_t^{\alpha}(A_t L_t)^{1-\alpha}$, where α denoting the capital share - ($\alpha = 0.33$), if needed, the form of the utility function is $u(c_t) = \frac{c_t^{1-\theta}}{1-\theta}$ with θ denoting the inverse of inter-temporal elasticity of substitution - $\theta = 2$. The time preference parameter is $\rho = 0.02$. Technology and population grow exogenously, $A_t = A_0 e^{gt}$ and $L_t = L_0 e^{nt}$, with $A_0 > 0$ and $L_0 > 0$. Should it be required, calibrate the savings rate (s) to 0.2. The depreciation rate, the rate of population growth, and the rate of technological progress take the following values: $\delta = 0.05$, $n = 0.01$, $g = 0.02$. Assume that the initial steady state level of capital per effective unit of labor is 1.5 (for convergence graphs you may need to use different values). For those graphs that require a value of government per effective units of labor (χ) - corresponding to questions around Tutorial 2 -, you can choose a parametrization (20% output) so that the figure makes sense. The same for those exercises that require a shift in one parameter.

You are expected to submit one deliverable: (1) a Jupyter Notebook (.ipynb) with comments or a Markdown file (.md) with comments, a Python file (.py) file with comments, indicating clearly your group and names in the file. You can also send me an R code but Python is preferred. Ensure that you have Python, or R, in order to run the code. I use VScode as an interface to run both Python and R codes. If you install VSCode, you will need to also install R and Python into your computers. You should already know how to do this.

Your code will be evaluated based on the following criteria:

- whether the code runs correctly and replicates the figures.
- whether the code reflects understanding of the topic.
- whether the code is written using functions (the command def in Python) and can be used to do comparative analysis (analyze a change in a parameter).
- the code uses colors to represent lines (well written in the legend). Arrows should be used indicate dynamics of consumption and capital in the corresponding figures.

Failing to submit the assignment will result in a grade of 0. Replicating main figures will give you 15pts. Bonus will give you an extra 5pts.

The group allocation is as follows:

- Groups 1-10 – Figures in pages 29 and 39 of Chapter 1 (axes will need to be changed as slope is small), and Figure 2 of Tutorial 1
  - Bonus: Figure 2 of Tutorial 2
- Groups 11-20 – Figures in pages 41 of Chapter 1 (axes will need to be changed as slope is small), and a plot like the one in page 29 reflecting the shift in s
  - Bonus: Figure 3 of Tutorial 2
- Groups 21-30 – Figures in pages 42 of Chapter 1 (axes will need to be changed as slope is small), and a plot like the one in page 29 reflecting the shift in n
  - Bonus: Figure 5 of Tutorial 2
- Groups 31-40 – Figures in pages 42 of Chapter 1 (axes will need to be changed as slope is small), and a plot like the one in page 29 reflecting the shift in δ
  - Bonus: Figure 2 of Tutorial 2
- Groups 41-50 – Figures in pages 43 of Chapter 1 (axes will need to be changed as slope is small)
  - Bonus: Figure 2 of Tutorial 2
- Groups 51-60 – Figures in page 55 in Chapter 1 (axes will need to be changed as slope is small)
  - Bonus: Figure 3 of Tutorial 2
- Groups 61-64 – Figures in page 57 in Chapter 1 (axes will need to be changed as slope is small)
  - Bonus: Figure 5 of Tutorial 2

Figure 1 (code in moodle) serves as a reference model for replicating the phase diagram in the Ramsey model.

An example of a Python code that replicates the Golden Rule of capital (in page 53 of Ch. 1) can be found in moodle. This code can be used as a starting point.

Things to note:

- For the Figures that require plotting a phase diagram, you will need to *compute* the stable saddle path (see the Python code in moodle)! For this, you can use Python to solve Ordinary Differential Equations when needed.
- If the change in a single parameter does not move the curve too much, you may want to zoom in to illustrate a change in a parameter (like in Figure 2 of TD2).
- For those groups replicating similar (or same) Figures, please refrain from communicating with one another (as I will be checking your codes).

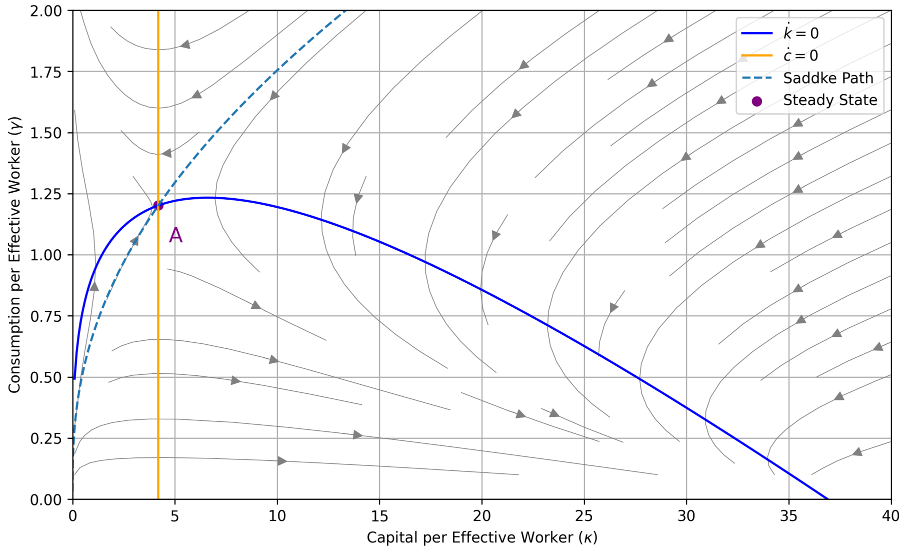

Figure 1: Phase Diagram in CKR model

- You may use ChatGPT to assist you in replicating the figures, but be mindful that it may not always provides correct answers (even when you provide the graphs/code themselves). Your safest bet is to adapt the codes yourselves. You will need to thoroughly check the code.
- You should always verify the output to ensure your understanding of the material. You can for instance ask ChatGPT to check through your code (to make it more efficient for instance). Asking precise questions will help, but be aware of potential feedback loops that may hinder progress. For example, give ChatGPT the graph provided and it will give you something, but will not necessarily work well.
- A very good resource for python codes for economists is https://python.quantecon.org/intro.html.

Good luck.