Written Exam Economics summer 2022

Introduction to Programming and Numerical Analysis

From May 25th 10.00 to May 27th 10.00

This exam question consists of 1 page in total

Answers only in English.

A take-home exam paper cannot exceed 10 pages – and one page is defined as 2400 keystrokes

Submission instructions

In addition to the Jupyter Notebook and .py files containing your exam answers, you must hand in your portfolio of projects completed during the semester. Therefore, place your exam notebook, together with all accompanying files, in a folder called 'exam' in your local git repository. Zip the whole repository into 1 file. Name the file according to your group name (eg. 'myGroup.zip') and upload this file to Digital Exam.

Furthermore: Write which groups you have given peer feed back to (and for which projects) in the main README.md file of your repository. Also write the KU-idents of all group members. (The main README is located together with the 3 projects folders, the .gitignore and the LICENSE file.)

Be careful not to cheat at exams!

Exam cheating is for example if you:

- Copy other people's texts without making use of quotation marks and source referencing, so that it may appear to be your own text
- Use the ideas or thoughts of others without making use of source referencing, so it may appear to be your own idea or your thoughts
- Reuse parts of a written paper that you have previously submitted and for which you have received a pass grade without making use of quotation marks or source references (self-plagiarism)
- Receive help from others in contrary to the rules laid down in part 4.12 of the Faculty of Social Science's common part of the curriculum on cooperation/sparring

You can read more about the rules on exam cheating on your Study Site and in part 4.12 of the Faculty of Social Science's common part of the curriculum.

Exam cheating is always sanctioned by a written warning and expulsion from the exam in question. In most cases, the student will also be expelled from the University for one semester.

In [ ]:
```python
# Standard modules needed
import numpy as np
import pandas as pd
import datetime as dt
from types import SimpleNamespace
%load_ext autoreload
%autoreload 2

%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
plt.style.use('seaborn-whitegrid')
```

Windmill industry in Denmark

This exercise deals with the power generated by windmills in Denmark. As windmills are considered an important element in the transition to a zero emission economy we should take a closer look at the electricity production stemming from them.

To this end, you need to download the official data set containing all windmills in Denmark which produced by the Danish Energy Agency. You can download it using the module `requests` and then save it as the xlsx file `windmills.xlsx`. This is done in the cell below.

In [ ]:
```python
import requests

url = "https://ens.dk/sites/ens.dk/files/Statistik/anlaeg.xlsx"
r = requests.get(url)

with open('windmills.xlsx', 'wb') as xls_file:
    xls_file.write(r.content)
```

Note: `requests` is in the Anaconda distribution, so it should be available to you. If it is not, install it by running `python -m pip install requests` in your terminal.

By inspecting `windmills.xlsx` you'll see it has two sheets: 1) `IkkeAfmeldte-Existing turbines` which holds mills (turbines) currently in use and 2) `Afmeldte-Decommissioned` which are no longer in service. Throughout the rest of the exercise, you need to use data from both of them.

Question 1

1. Load the two sheets of `windmills.xlsx` into your notebook and combine them into one pandas DataFrame. Note that you need to do some data cleaning in the process. For instance, you can disregard the variables which are not present in both sheets. You can decide for yourself whether you want to use English or Danish column names.
2. Plot the development in total electricity production from windmills between 1977-2021. You may use GWh (1 million KWh) as unit.

Question 2

We now want to know how the capacity of electricity production has changed over the years. For this we need two variables `Date of original connection to grid` and `Capacity (kW)`. The first indicates when a mill was initiated and the other its production capacity.

1. Calculate and plot the development in **average** and **maximum** capacity of turbines based on their year of initialization (1977-2021)
   Note: capacity is measured in KW, so you need to multiply with the number of hours pr year to make it comparable with annual production.
2. To get a cleaner view of the trends, compute and plot the 7 year moving average of annual mean capacity and annual max capacity.

   $\mathrm{ma}^7(x_t) = \frac{x_{t-3}+x_{t-2}+\ldots+x_t+\ldots+x_{t+3}}{7}$
3. Finally, compute the **total capacity** of all windmills in Denmark in each year. Plot total capacity together with actual production as calculated in Question 1.
   Note: the capacity of a turbine should only be included for the years when it is connected to the grid.

Question 3

There is a lot of debate about the visual impact on the landscape from windmills. This exercise deals with the relationship between windmill size and productivity to get a sense of the tradeoff. You can solve it in many different ways depending on your preferences. The important thing is that the results are clear.

1. Compute and plot the relationship between height of windmills and their electricity production in 2021. The variable `Hub height (m)` indicates the height of a mill in meters. One possible approach is to discretize the height variable and compute the median electricity production within each bin. But you can also apply a statistical model of your own choosing.
2. Repeat the method you chose above, but now group over the type of location as well. Locations are described by the variable `Type of location` and can be either *off-shore* ("HAV") or *on-shore* ("LAND","Land"). When plotting the results, use common limits on the y-axis for better comparison.
3. Finally, we dig into the productivity of on-shore vs. off-shore mills.
   Compute and plot the *average difference* between annual capacity and annual production for mills on-shore and mills off-shore in each year 1990-2021.

A discrete-continuous consumption-saving model

Here we will consider a modification to the 2-period consumption saving model.
In addition to making a consumption-saving choice in the first period, there is now also a binary choice of whether or not to attend costly education. Taking an education is in this model associated with higher expected earnings in period 2 but it comes at a monetary cost in period 1.

Second period
Household gets utility from **consuming** and **leaving a bequest**:

$$
\begin{aligned}
v_2(m_2) &= \max_{c_2} \frac{c_2^{1-\rho}}{1-\rho} + \nu \frac{(a_2+\kappa)^{1-\rho}}{1-\rho} \\
&\text{s.t.} \\
a_2 &= m_2 - c_2 \\
a_2 &\geq 0
\end{aligned}
$$

First period
Household gets utility from consuming. It takes into account that if choosing to go to school today, expected income will be higher in second period.

$$
\begin{aligned}
v_1(m_1) &= \max_{c_1,s} \frac{c_1^{1-\rho}}{1-\rho} + \beta \mathbb{E}_1\left[v_2(m_2)\right] \\
&\text{s.t.} \\
s &= \begin{cases} 1 & \text{if study in period 1} \\ 0 & \text{otherwise} \end{cases} \\
a_1 &= m_1 - c_1 - \tau s \\
m_2 &= (1+r)a_1 + y_2 \\
y_2 &= \begin{cases} \bar{y} + \gamma s + \Delta & \text{with prob. } p \\ \bar{y} + \gamma s - \Delta & \text{with prob. } 1-p \end{cases} \\
a_1 &\geq 0
\end{aligned}
$$

- s is a binary indicator for whether the agent chooses to study in period 1.
- c is consumption
- γ is the income premium associated with having studied
- τ is the monetary cost of studying, paid in period 1
- m is cash-on-hand
- a is end-of-period assets
- $\bar{y}$ is base income in period 2
- y_2 is total realized income in period 2
- $\Delta \in (0,1)$ is the level of income risk (mean-preserving if $p = 0.5$)
- r is the interest rate
- $\beta > 0$ is the discount factor
- $\mathbb{E}_1$ is the expectation operator conditional on information in period 1
- $a \geq 0$ ensures the household *cannot* borrow

Hint: the study choice is discrete (and thus not differentiable) which means that it cannot be optimized for in the same manner as the continuous consumption choice. Therefore, you need to solve the consumption problem *for each* of the two study choices and pick the **combination** of studying and consuming that yields highest value as the model solution.

In [ ]:
```python
# Parameters
rho = 8.0
nu = 0.1
kappa = 1
beta = 0.90

tau = 0.8
gamma = 1.2
ybar = 1.5
r = 0.04
p = 0.5
Delta = 0.4

m_min = tau+1e-5     # minimum value for m - must be possible to pay for studying
m_max = 5.0          # maximum value for m
```

Question 1

1. Solve the model for the parameters above.
2. Plot $v_1(m_1)$ and $v_2(m_2)$. Comment.
3. Plot the optimal consumption function $c_1^*(m_1)$ and $c_2^*(m_2)$ in one graph. Comment on the shapes of the functions.
4. Plot the optimal study choice function $\mathbb{I}^{s*}(m_1)$. Comment on the shape of the function.

Question 2

1. Given the wage premium on education, compute the **smallest** education cost τ such that an agent with $m_1 = 3.0$ will **no longer** choose to study.
   Hint: there are different ways of obtaining that number. A bisection algorithm is one possibility.

Approximating a function

In this exercise, you will implement an algorithm to approximate a function $f(x)$ if x is on the interval $[-1,1]$.

A degree N approximation of $f(x)$ takes the general form

$$\hat{f}(x) = \sum_{i=1}^{N} a_i T_i(x)$$

for which you need 3 elements:

1. the functions $T_i(x)$
2. M evaluation nodes $\{z_k\}$.
3. N coefficients $\{a_i\}$

1.
The functions $T_i(x)$ take the form

$$T_i(x) = \cos(i \times \arccos(x))$$

2.
The true function f needs to be evaluated on a set of nodes so that we can use these function evaluations for our approximation. The set of nodes where f is evaluated, $\{z_k\}$, has to be chosen wisely such that the approximation error is minimized.
It turns out to be on the form

$$z_k = -cos(\frac{2k-1}{2M}\pi), \quad k = 1,2,3,\ldots,M$$

3.
The N coefficients of the approximation are obtained by what is essentially a least squares regression. They are on the form

$$a_i = \frac{\sum_{k=1}^{M} f(z_k) T_i(z_k)}{\sum_{k=1}^{M} T_i(z_k)^2}, \quad i = 1,2,3,\ldots,N$$

Notes: in general one can let $N < M$.
Observe that we are using M evaluations of $f(z)$ to create **each** of the N approximation coefficients. This can be done up front and needs only to be done once even if you need to approximate f on multiple x's. This is why such an approximation is useful in the context of solving an economic model. For instance, a value function may be very computationally intensive, so you'll benefit from only having to to evaluate it M times in order to get, say, $K >> M$ function approximations.

Question 1

Create an approximator $\hat{f}(x)$ at an $x \in [-1,1]$ by implementing the following algorithm:

1. For each $k = 1,\ldots,M$: compute $z_k = -cos(\frac{2k-1}{2M}\pi)$
2. For each $k = 1,\ldots,M$: compute $y_k = f(z_k)$
3. For each $i = 1,\ldots,N$: compute $a_i = \frac{\sum_{k=1}^{M} y_k T_i(z_k)}{\sum_{k=1}^{M} T_i(z_k)^2}$
4. Return $\sum_{i=1}^{N} a_i T_i(x)$

In [ ]:
```python
def f_approx(x, f, N, M):
    pass
```

Note: you can use the numpy functions `np.arccos` in T_i and `np.cos` in z_k.

Question 2

Evaluate `f_approx` at $x \in \{-0.5, 0.0, 0.98\}$ and report in each case also the deviation from the true value `f(x)`.

Use the following

In [ ]:
```python
f = lambda x: 1/(1+x**2) + x**3 - 0.5*x
N = 5
M = 8
xs = np.array([-0.5, 0.0, 0.98])
```