

Exercise class 4

Introduction to Programming
and Numerical Analysis

Class 3 and 6

Annasofie Marckstrøm Olesen
Spring 2024

UNIVERSITY OF COPENHAGEN



Random numbers

User-defined modules

Problem set 2

Poll

Problem set 2 and using random numbers

Simulating random numbers is an absolutely fundamental tool in numerical economics:

- Models with uncertainty about outcomes
- Models with heterogeneity in preferences, endowments, productivity...
- Simulation of difficult integrals (Monte Carlo Integration)

Take-aways from this weeks' lecture + exercise:

- Drawing random numbers from specified distributions.
- Using seeds and states to ensure reproducibility.
- Solving an exchange economy model with heterogenous preferences.
- Numerical integration by Monte Carlo.

User-defined modules

It is important to keep your code as clean and easy to read as possible. A good way to do so is to collect functions and classes in **modules**.

You can write the functions you need for your project in a .py-file and then **import** it to your notebook the same way you would import eg. numpy.

This keeps your notebooks neat and tidy, and is also a great way to secure yourself against scope bugs!

Problem set 2

Plan for today:

- Now-16: You solve the **Tasks**.
- 16-16.15: Break
- 16.15-16.25: Tips for the **Problem**.
- 16.50-17: Finish together + Poll

$(0,0): x = 0.569$

$(0,1): x = 0.077$

$(1,0): x = 0.569$

$(1,1): x = 0.077$

$(2,0): x = 0.569$

$(2,1): x = 0.077$

Text output in task 1

Solving the exchange economy equilibrium

Think "algorithmically" - what does the solution look like, and which steps do we take to get there?

The solution: A set of prices that result in market clearing.

The steps:

1. Create the agents, their preferences and their endowments using a random number generator.
2. For a given set of prices p , compute each agent's income and demand for each good.
3. Compute aggregate demand, aggregate supply and excess demand at prices p .
4. If markets do not clear, try another value of p and return to step 2. Use a root finding algorithm to update guesses.

More general tips

Break the problem into smaller steps - which parameters do you have? Which functions do you need?

Always test your code as you go! Note down what each function should do and check that they actually do that before continuing to the next step of the problem.

I usually solve these types of problems "inside out" - start with the "building blocks" (utility function, demand function, supply etc.) and check that they work individually before putting them together to compute excess demand and eventually equilibrium.



Poll



Student Login

Room Name

JOIN

 English ▾

Next time

Video lectures

- Structure, debugging and Git
- Consumption
- Labor supply
- Production Economy

Physical lecture:

- Inaugural project

Exercise:

- Introduction to Git
- Work on inaugural project