



UNIVERSITY OF
COPENHAGEN

Fifth exercise class

Class 5

Introduction to numerical programming and analysis

Asker Nygaard Christensen

Spring 2021

KUnet and Absalon are inaccessible from Friday 19th, 17:00 to Sunday 21st, 18:00, because of some security update.

So make sure you have watched the 'Handing-in with Git'-video before the weekend. [This hyperlink](#) might still work during the update, but I'm not certain.

Microsoft Teams and Github will still be working so you can still contact me and hand-in.

Apparently my work email is also closed down. But you can contact me at my personal email: nygaardchristensen7@gmail.com. Only for this weekend however, afterwards I'd prefer to still get it at my work email: hms467@econ.ku.dk

The Inaugural Project

Inaugural Project, tips

Q1: With some algebra, the optimisation problem can be rewritten, by combining the constraints and inserting them into the utility function, such that you're only optimising with respect to p_h (which is the same as h) with no constraints. This makes the computation faster, and also easier to write up in code. Then you just need to maximise utility. `optimize.minimize_scalar` has the easiest syntax when you're only optimising with respect to one variable.

While you don't need a bounded optimiser, having it is very much allowed. There are many different ways to get the correct result. Personally, I think it would be stringently more correct to have h bounded between 0 and the maximum amount allowed given m and housing expenses. But since it gives the same results for the parameters, both approaches are correct.

Q3: You're just solving Q1 multiple times. Then, given optimised choices, you'll be able to calculate avg. taxes.

Inaugural Project, tips

Q5: If you wrap your answer to Q3 in a function, that takes τ^g as an argument, you have a simple root finding problem (make sure that arguments are nested properly such that consumer choices change, depending on τ^g !). You can use the *optimize.root*-function. Alternatively you can also assume that taxes paid correlate positively with τ^g , and loop over guesses that change depending on the distance to the tax-goal. Similar to the exchange economy problem in problem set 3.

The assumptions holds because of the preferences, but you you can also just show it empirically by plotting avg. taxes as a function of τ^g .

Inaugural Project, tips

Don't forget to document and comment your code properly, following the guidelines explained in lecture 5. Good documentation is good coding practice. It helps me and your peer-reviewers if you document now, but most importantly, you'll help your future selves to understand your own code.

Also: Be very mindful of positional and keywords arguments, and making sure to reference them in the right way.

Remember that *optimize.minimize_scalar* minimises with respect to the first argument of the objective-function.

It's always good to make logic-checks, after finding your answers. For example, after finding h^* , maybe try to plot utility as function of h , and make sure your answer looks correct.

Sources of inspiration for the inaugural project:

The deadline for the inaugural problem is 21st of March.

1. PS1 and PS2 contains the tools you need for the project, understanding and using them is crucial.
2. Lecture 3 and 4 also covers the important topics, drawing inspiration and re-applying code from them is allowed.
3. Last year's inaugural project is very similar to this year's. Jeppe showed the solution to this in monday's lecture.

Programming can be frustrating. Remember to take breaks when encountering frustrating bugs. Fresh eyes are much better at spotting bugs, this applies both to asking a friend, but also for taking breaks. It's okay to hand-in something where the final result is wrong, but you've been unable to fix it. Just make sure to document it, and I'll help you. You'll have time to revise for the exam.