Exercise class 4

Introduction to Programming and Numerical Analysis

Class 3 and 6
Annasofie Marckstrøm Olesen
Spring 2024

UNIVERSITY OF COPENHAGEN

Numerical optimization

My thoughts on the problem sets

Problem set 1

Constrained optimization

Problem set 1

Optimization

Objective: find x that minimizes $f(x)$

We are used to solving optimization problems analytically using first and second order conditions.

Different on computer: We can only evaluate the function in one point at a time - how do we find the minimum?

(Note: In computer science, the convention is to *minimize* - if we instead want to *maximize*, we can just minimize the negative of the function)

Optimization: Grid search

One way is to just try a bunch if different values of x and pick the one that gives the smallest value of $f(x)$.

Pros:

- Gives a rough idea of what the function looks like.
- Relatively robust against non-global minima.

Cons:

- Computer-intensive - especially in higher dimensions.
- Does not check outside grid.
- Solution only as precise as grid.

Optimization: Solvers

A solver is an algorithm that looks for a minimum by trying different values of x and updating guesses based on the evaluation of $f(x)$.

Which x's to guess on are determined by the algorithm - except the starting point, which must be provided.
Pros:

- Faster and less intensive than grid search.
- More precise solution.

Cons:

- Solution may depend on starting value.
- May not converge to a solution - then what?

My thoughts on the problem sets

The problem sets will be more challenging than DataCamp, but they train you in everything you need to know for the exam!

If you don't make it through all of the problems at the exercise class, finish them at home.

OK to copy code from lectures or earlier problem sets. Coding is rarely done completely from scratch.

OK to peek at solutions. But you cheat yourselves if you don't make sure you can solve the problems by yourselves.

Extra problems are great for preparing you for assignments, since they often force you to think a little creatively.

Tips for problem set 1

When making mesh grids for plots, remember to put `indexing='ij'` as keyword argument - otherwise the grids will not be in the correct order for the plot to work out.

Remembering the exact syntax for optimizers, plots etc. is difficult - googling is okay!

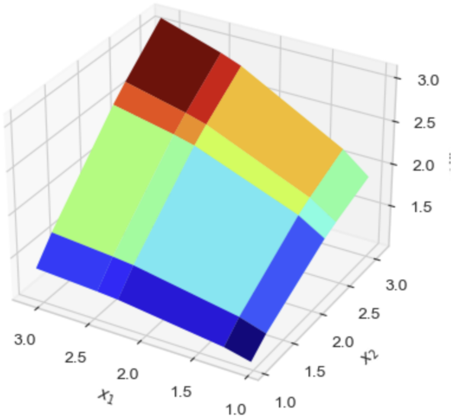
Tips for problem set 1

In task 2, you need to print out a table looking like this:

	0	1	2	3	4
0	1.050	1.205	1.651	1.717	1.885
1	1.121	1.300	1.834	1.916	2.128
2	1.279	1.518	2.300	2.430	2.781
3	1.299	1.545	2.362	2.500	2.872
4	1.343	1.608	2.514	2.671	3.100

Tips for problem set 1

In task 3, you need to recreate this figure:



Problem set 1

Problem set 1 trains you in this week's topics: Optimizing, printing and plotting results.

Plan for today:

- Now-15.16: Work on problem set 1
- 16-16.15: Break
- 16.15-16.20: Constrained optimization
- 16.20-16.55: Work on problem set 1
- 16.55-17.00: Follow-up in class

Constrained optimization

Constrained optimization works in much the same way: you can use grid search or solvers.

OR: You can adjust the objective function with a penalty if the constraint is violated and use unconstrained optimization.

- Pro: Helps "guide" the solver, if it ends up outside the constraint.
- Con: Can introduce new local minima.

Optimization: Solvers

Which algorithm should I use? Depends on the problem. Here are some examples:

Unconstrained optimization:

- `bfgs` (fast, especially if you provide gradient/hessian)
- `nelder-mead` (robust, but slow)

Constrained optimization:

- `slsqp` (fast, especially if you provide gradient/hessian)
- ...or use unconstrained optimization + penalty of constraint is violated.

Next time

Video lectures

- Random numbers
- (Maybe front load a bit on the videos, the week after has both videos and a physical lecture)

Problem set 2:

- Drawing random numbers
- Monte Carlo integration
- User defined modules
- Saving and loading
- ...and of course everything you've learned until now