Computer Laboratory Exercises in Optimization 2023
Lab 1: Line search and unconstrained optimization

The lab is run with the MATLAB package, version R2022a or later, which you can install freely on your computer; see `http://program.ddg.lth.se/en/`. *Make sure you choose to install the Optimization and the Symbolic Math toolboxes.* You should work in a group of 2–3 students.

Preparations to do before the lab session in order to pass

- If you are not familiar with MATLAB you should examine this interactive program with the command `demo`. A faster way may be to type `help demos` and choose a demo from the list, e.g., type `intro`. You might find it useful to look at the MATLAB links on the course Canvas web page under Assignments.

- Read in the textbook and be sure that you understand the line search methods and the multidimensional methods Steepest Descent, Newton and Modified Newton.

- Solve Exercises 3.5, 6 and 8 in the textbook by implementing the line search methods Dichotomous, Golden Section, Bisection and Newton in MATLAB. Write one m-function file per method, for example, starting like

```
function [X,N]=goldensection(F,a,b,tol)
% For example F = @(x) 1-x*exp(-x)
% [a,b] interval
% tol = max ratio of final to initial interval lengths
% X output matrix containing final a, b and b-a from every iteration
% N = number of function evaluations
```

- Download the files `Lab1a.m`, `Lab1b.m`, `Function3.m` and `Function4.m` from the course web page.

- Type `Lab1a` in the MATLAB window to start the program. A new window appears. Select ‘Golden’. The function shown (`Function 1`) should be minimized on the interval $-0.5 \leq x \leq 0.5$. Select an initial interval by clicking two points in the window. For each next click, anywhere in the window, a new smaller interval of uncertainty will be shown. Change the visualized interval to $-10 \leq x \leq 10$. Try several starting intervals. Investigate also `Function 2` on the interval $-10 \leq x \leq 10$.

During the laboratory session

1. Start MATLAB. Be ready to present your implemented line search programs and to test them with a function given by the teacher.

2. Start `Lab1a`. Repeat the investigations (made above for Golden Section) for Newton's method. Try different initial points all over the interval $-10 \leq x \leq 10$ for both functions.

3. Questions to be answered for Golden Section and Newton's method:

   - Does the method always converge? newton no golden

yes

finds local minima but can also find the wrong point

- What kind of points are found? Global minimum, local minimum or something else? we can find global, local minima and maxima depending on where we start

most of the time it finds what it needs to do

- Would you say that it is a reliable method?

not really, it has some unpredictable behaviour

- How far away from a minimum can you start without losing the convergence to it? difficult to say, not that far

quite far, depends on the function

when changing epsilon we change the slope of the line beneath which we find acceptable points.

When changing alpha we change the amount of points.

4. Investigate Armijo's method. Choose `Function 1` and the interval $-0.5 \leq x \leq 0.5$. A single click in the window selects the point $\boldsymbol{x}_0$, then some possible iterations that *may occur*. What is $\boldsymbol{d}$ in the figure? Can you point out the λ-axis and its origin? Interpret and explain the red and blue asterisks. Use the sliders and explain what happens.

blue asterisks is the point which the second rule that F(la) >= T(la)

the lambda axis is the x-axis shifted to start in x0

5. Try two MATLAB methods for minimization of `Function 1` by typing

    `f1=@(x) (-x^4+6*x^2)/(1+(x-0.5)^4)`

    `options=optimset('display','iter');` performs multiple searches

    `fminbnd(f1,-1,1,options)` (the interval is here $[-1, 1]$; try also $[-10, 10]$)

    `[x,fval,exitflag,output]=fminunc(f1,-1)` (initial point is -1; try other)

feels similar to newton methods

Do you see any similarities with Golden Section or Newton's methods? Is the MATLAB result satisfactory? Convergence? Global minimum? Do the methods use derivatives? fminbnd uses iterative steps of multiple points fminunc uses derivaties

6. Start `Lab1b`. In a new window you can minimize functions of two variables with

    Steepest Descent with Armijo's line search

    Newton's method without line search

    Modified Newton's method without line search.

    Investigate first Steepest Descent on `Function 1`. Choose an initial point with the mouse and continue clicking to see the iterations. After you have examined the behaviour for many different starting points, try Newton's method in the same way (how do you choose this method?), and finally the modified Newton for different values of ε. Investigate also `Function 2` in the same way.

7. Questions to be answered for each of the three methods:

    - Does the method always converge? Is it fast or slow? For what kind of functions does it work best?
    - What kind of points are found by the method? Global or local minimizer or something else?
    - How does the size of ε in the Modified Newton method affect the performance?
    - What are the connections between the three methods?

8. For `Function 2`, try to find a good strategy on how to choose a method and an ε on every iteration to find the minimizer with a few number of iterations as possible.

9. Do the same investigations for `Function 3` and `Function 4` first without seeing any contours (`Levels = 2`). Can you be sure that you have found the global minimum? To which points do the methods converge? Later choose more contours with the bar.

10. Download `Function3.m` and `Function4.m` from the course web page. Compare with MATLAB's `fminsearch` by typing

    ```
    [point,val]=fminsearch('Function3',[5 5],options)
    ```

    ```
    [point,val]=fminsearch('Function4',[5 5],options)
    ```

    Here [5 5] is the initial point (try several different).

    Do they find the global minimizer? Which method does MATLAB use when there is no information about the differentiability of the function? Search for animations of this method on the internet and watch how it proceeds in two dimensions.