INSTRUCTIONS

- The assignment consists of 3 questions for a total of 100 marks.
- Submit your assignment **electronically** via `https://odtuclass.metu.edu.tr/`.
- The report should be written via LaTeX.
- Please, do not forget send .m files, .tex file of your report with the .pdf form.
- Please, give a name for your pdf document as **your name, surname, and student ID**. For instance, **mkutuk_2389369**.
- The report for a programming assignment should consist of
  - Brief description of the problem;
  - Results such as data, graphs etc. Try to avoid attaching large set of data unless it is really necessary;
  - Discussion, comments, explanation and conclusion on your numerical observations. This part is equally important as your computer codes and the data you collect, and it helps with understanding concepts, algorithms, or other relevant issues not discussed during lectures.
- Every .m file should be documented, such as:

```
% CA1_numberExercises.m
%
% Author: (Your Name)
%
% Description:
% (Give a brief description of what your program does.)
%
% Input:
% (State what the program inputs are.)
%
% Output:
% (State what the program outputs are.)
%
% Usage:
% (Give an example of how to use your program.)
```

- For one day late submission, you will get 75% of your grade. For two days late submission, you will get 50% of your grade. After 2 days late, your homework will not be graded.

1. (30 *pts.*) In this exercise, use/update Matlab routines provided in **LAB Hour**. Let Beale funtion

$$f(x_1, x_2) = (1.5 - x_1 + x_1 x_2)^2 + (2.5 - x_1 + x_1 x_2^2)^2 + (2.625 - x_1 + x_1 x_2^3)^2$$

be given. For the initial point $x_0 = (3, 0.5)^T$, minimize $f(x_1, x_2)$ by implementing:

a) Steepest Descent using the Armijo condition,

b) Newton method (without line search),

c) Quasi–Newton: update BFGS.

Choosing different tolerance values such as `tol`=1e-2,1e-3,1e-4, create a comparative table describing the number of iterations, norms, i.e., $\|x_k - x^*\|_2$, where the optimal solution $x^* = [3.025, 0.474]^T$. Discuss your observations. Compare the results with Matlab built–in functions `fminunc` or `fminsearch`. Moreover, draw the trace of $\{x_k\}$ for each case.

2. (30 *pts.*) Write a Matlab routine for the **conjugate gradient algorithm** (Fletcher–Reeves form) given in the lecture with initial point `x0`, tolerance `tol` for the termination condition $||r_{i+1}|| <$ `tol`, and `maxit` the maximum number of iterations. Using this routine, solve linear systems in which A is Hilbert matrix, whose elements are

$$A_{\mathrm{i,j}} = 1/(i + j - 1).$$

Set the right-hand-side to $b = (1, 1, ..., 1)^T$ and the initial point to `x0` $= 0$. Try dimensions $n = 5, 8, 12, 20$ and report condition number of the matrix A and the number of iterations required to reduce the residual below 10^{-6} with `maxit=1000`. Discus your observations.

3. (40 *pts.*) Write a Matlab routine called `steihaugCG_TR.m` that implements a trust-region method with trial steps computed from the Steihaug truncated linear CG method given in **Algorithm: Steihaug Method based on linear CG**. The function call should have the form

[`x`, `F`, `G`, `H`, `iter`, `status`] = `steihaugCG_TR.m`(fun, x0, `maxit`, `tol`, `tau1`, `tau2`, `Delta0`),

where `fun` represents the name of a Matlab m–function that computes $f(x)$, $\nabla f(x)$, and $\nabla^2 f(x)$ for some desired function f; it should be of the form

[`F`, `G`, `H`] = `fun.m`(x),

where for a given value x it returns the values of the function, gradient, and Hessian, respectively.

The parameter `x0` is an initial guess at a minimizer of f, `maxit` is the maximum number of iterations allowed, `tol` is the final stopping tolerance, `tau1` and `tau2` are given parameter to check update the radius, and `Delta0` is the initial radius. On output, the parameters `x`, `F`, `G`, and `H` should contain the final iterate, function value, gradient

Algorithm: Steihaug Method based on linear CG

Input symmetric matrix $B \in \mathbb{R}^{nxn}$ and vector g.
Choose stopping tolerance $\tau_{\text{stop}} > 0$.
Set $p_0 = 0$, $r_0 \leftarrow g$, $s_0 \leftarrow -g$, and $k \leftarrow 0$.
while $||r_{\text{k}}|| > \tau_{\text{stop}} ||r_0||$ **do**
  if ${s_{\text{k}}}^T B s_{\text{k}} > 0$ **then**
    Set $\alpha_{\text{k}} \leftarrow ({r_{\text{k}}}^T r_{\text{k}})/({s_{\text{k}}}^T B s_{\text{k}})$.
  else
    Set $p_{\text{k}+1} \leftarrow p_{\text{k}} + \tau s_{\text{k}}$, where τ is the positive root of $||p_{\text{k}} + \tau s_{\text{k}}||_2 = \delta$.
    return $p^{CG} := p_{\text{k}+1}$
  end if
  if $||p_{\text{k}} + \alpha_{\text{k}} s_{\text{k}}||_2 < \delta$ **then**
    Set $p_{\text{k}+1} \leftarrow p_{\text{k}} + \alpha_{\text{k}} s_{\text{k}}$.
  else
    Set $p_{\text{k}+1} \leftarrow p_{\text{k}} + \tau s_{\text{k}}$, where τ is the positive root of $||p_{\text{k}} + \tau s_{\text{k}}||_2 = \delta$.
    return $p^{CG} := p_{\text{k}+1}$
  end if
  Set $r_{\text{k}+1} \leftarrow r_{\text{k}} + \alpha_{\text{k}} B s_{\text{k}}$.
  Set $\beta_{\text{k}+1} \leftarrow ({r_{\text{k}+1}}^T r_{\text{k}+1})/({r_{\text{k}}}^T r_{\text{k}})$.
  Set $s_{\text{k}+1} \leftarrow -r_{\text{k}+1} + \beta_{\text{k}+1} s_{\text{k}}$.
  Set $k \leftarrow k + 1$.
end while
return $p^{CG} := p_{\text{k}}$

vector, and Hessian matrix computed by the algorithm. The parameter `iter` should contain the total number of iterations performed. Finally, `status` should have the value 0 if the final stopping tolerance was obtained and the value 1 otherwise.

In the routine `steihaugCG_TR.m`, call `steihaugCG.m` in order to compute the trial step, having the following form

$$[\texttt{p}, \texttt{iter}, \texttt{flag}] = \texttt{steihaugCG.m}(\texttt{B}, \texttt{g}, \texttt{radius}, \texttt{tol}),$$

where the input `B` is required to be a symmetric (possibly indefinite) matrix, `g` is a vector, $\delta =$ `radius` > 0 is the trust-region radius, and `tol` is the stopping tolerance. On exit, the resulting approximate Steihaug truncated-CG solution should be stored in the vector `p`, the number of iterations performed stored in `iter`, and the parameter `flag` should be set to one of the following values: -1 if the algorithm terminated because of negative curvature, 0 if the stopping tolerance was met, and 1 if the algorithm returned a boundary solution simply because the CG iterations grew larger than the trust-region radius.

Test your routine for the Rosenbrock function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \; x = (x_1, x_2)^T \in \mathbb{R}^2$$

using the following parameters

$$\texttt{x0} = [0, -1]^T, \ \texttt{tol}{=}1.0\text{e-}3, \ \texttt{tau1}{=}1/4, \ \texttt{tau2}{=}3/4, \ \texttt{Delta0} = 1, \text{ and } \texttt{maxit}{=}1000.$$

Comment on the results in the written report. Also, draw the contour lines of the quadratic model

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p$$

assuming that B is the Hessian of f. Compare your result and explain your observations.