EN.553.761: Nonlinear Optimization I
Homework Assignment #3

Starred exercises require the use of MATLAB.

Exercise 3.1: Prove Lemmas 2.2 and 2.3 from the lecture notes on “Conjugate Gradient”.

Exercise 3.2: Solve the trust-region subproblem

\[ \minimize_{s \in \mathbb{R}^n} s^T g + \frac{1}{2} s^T B s \quad \text{subject to} \quad \|s\|_2 \leq \delta \]  

in the following cases:

(a) \[ B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \ g = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \ \text{and} \ \delta = 2, \]

(b) \[ B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}, \ g = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \ \text{and} \ \delta = 5/12 \]

Hint: \( \lambda = 2 \) is a root of the nonlinear equation

\[ \frac{1}{(1 + \lambda)^2} + \frac{1}{(2 + \lambda)^2} = \frac{25}{144}. \]

(c) \[ B = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \ g = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \ \text{and} \ \delta = 5/12, \]

(d) \[ B = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \ g = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ \text{and} \ \delta = 1/2, \ \text{and} \]

(e) \[ B = \begin{pmatrix} -2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \ g = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \ \text{and} \ \delta = \sqrt{2}. \]

Exercise 3.3: Consider the solution of problem (1) with data

\[ B = \begin{pmatrix} 1 & 0 \\ 0 & 3 \end{pmatrix} \ \text{and} \ g = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \]

as a function of the trust-region radius \( \delta \). In which direction does the solution point as \( \delta \) shrinks to zero?

Exercise 3.4*: Write a MATLAB m-function called steihaug_CG.m that implements the truncated linear conjugate gradient method by Steihaug based on Algorithm 4 in the Lecture 06 slides. The function call should have the form
[ p, iters, flag ] = steihaug_CG( B, g, radius, tol )

where the input B is required to be a symmetric (possibly indefinite) matrix, g is a vector, radius > 0 is the trust-region radius, and tol ∈ (0,1) 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 iters, 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.

Exercise 3.5*: Consider the problem

\[
\text{minimize } f(x) \\
\text{for } x \in \mathbb{R}^n
\]

where \( f \) is a twice continuously differentiable function.

(a) Write a MATLAB m-function called unc_TR.m that implements a trust-region method with trial steps computed from the Steihaug truncated linear CG method as given in Problem 4.4 above. The function call should have the form

\[
[x,F,G,H,iter,status] = unc_TR(fun,x0,maxit,printlevel,tol)
\]

where \texttt{fun} is of type \texttt{string} and 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

\[
[F,G,H] = \text{fun}(x)
\]

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 \( f \), \texttt{maxit} is the maximum number of iterations allowed, \texttt{printlevel} determines the amount of printout required, and \texttt{tol} is the final stopping tolerance. In the code, if the parameter \texttt{printlevel} has the value zero, then no printing should occur; otherwise, a single line of output is printed (in column format) per iteration. On output, the parameters \( x \), \( F \), \( G \), and \( H \) should contain the final iterate, function value, gradient vector, and Hessian matrix computed by the algorithm. The parameter \texttt{iter} should contain the total number of iterations performed. Finally, \texttt{status} should have the value 0 if the final stopping tolerance was obtained and the value 1 otherwise.

(b) Write a separate MATLAB m-file with function declaration \([F,G,H] = \text{fun}(x)\) that returns the value \( F \), gradient \( G \), and Hessian \( H \) at the point \( x \in \mathbb{R}^2 \) of the function

\[
f(x) = 10(x_2 - x_1^2)^2 + (x_1 - 1)^2.
\]

Use your m-function \texttt{unc_TR.m} from part (a) to minimize \( f \) with starting point \( x_0 = (0,0) \).