Homework 3 (due on Wednesday 02/19/2020)

Problem 1. Exercise 5.2 in the textbook.

Problem 2. Exercise 5.6 in the textbook.

Problem 3. In this problem we will test the performance of linear conjugate gradient method for minimizing quadratic function $f(x) = \frac{1}{2}x^\intercal Ax - b^\intercal x$, where A is symmetric and positive definite.

- Program CG Algorithm 5.2 in the textbook.
- Apply your program on a symmetric and positive definite matrix A of dimension $10^3 \times 10^3$ with eigenvalues uniformly distributed between 10 and 10^3. Test the convergence of the algorithm and compare your numerical findings with the theoretical result shown in formular (5.36) in the textbook.

  (To set up matrix A with the desired properties, you can follow this procedure: 1) apply QR factorization to create a "random" orthogonal matrix Q using Matlab command [Q,R] = qr(rand(n,n)), where n is the dimension of the matrix. 2) set $A = Q^\intercal DQ$, where D is a diagonal matrix with diagonal entries being the desired eigenvalues.)
- Change the distribution of eigenvalues of A so that some eigenvalues are distributed between 9 and 11, the rest of eigenvalues are distributed between 999 and 1001, i.e., two clusters around 10 and 1000 with radius 1. Test the convergence performance on such matrix. Extra points if you can explain your numerical findings using the theoretical convergence results discussed in the lecture.

Problem 4. Consider the problem of minimizing

$$f(x_1, x_2, \cdots, x_n) \ = \ \sum_{i=1}^{n-1} \left[100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2\right].$$

The global minimum is at $x^* = [1, 1, \cdots, 1]^\intercal$. Numerically solve this problem using nonlinear conjugate gradient algorithms: 1) FR (Algorithm 5.4); 2) FR with restart based on (5.52); 3) PR based on (5.44), and compare their performance. (In the numerical experiments you can set $n = 100$ or any number that is not too small. The initial condition can be chosen as a random vector, for example, 2∗rand(n,1).)