MATH 309 Fall 2014 – Homework 2

The due date is 11:59 pm, October 15th (Wednesday). Please submit your your source codes and the required results via email to TJ Yusun at tyusun@sfu.ca. No late submission will be accepted. Partial points will be deducted for those who copy or duplicate the others' homework or work.

Problem 1: Implement the steepest descent method with the **exact** step length for solving

$$\min_x f(x) = \frac{1}{2}x^T Ax - b^T x,$$

where $b = (1, 1, \ldots, 1)^T$ and A is the $n \times n$ Hilbert matrix, whose elements are $A_{i,j} = 1/(i+j-1)$. Set the initial point to $x^0 = 0$. Try $n = 20$ and terminate the algorithm when $\|\nabla f(x^k)\| \leq 10^{-2}$. Report the objective function value f, the step length α and the norm of ∇f of the **last 10** iterations.

Problem 2: Implement the steepest descent method with the **inexact** step length satisfying the strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 0.1$ for solving the same optimization problem as given in Problem 1. Set the initial step length $\alpha_0 = 1$ and the initial point $x^0 = 0$. Try $n = 20$ and terminate the algorithm when $\|\nabla f(x^k)\| \leq 10^{-2}$. Report the objective function value f, the step length α and the norm of ∇f of the **last 10** iterations. Compared with the algorithm implemented in Problem 1, which one is faster? (**The codes for finding the step length satisfying the strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 0.1$ are posted on the Canvas.**)

Problem 3: Implement the steepest descent method with the **backtrack line search** (that is, Algorithm 3.1) to minimize the Rosenbrock function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

Set $\bar{\alpha} = 1$, $\rho = 0.9$ and $c = 10^{-4}$ and the initial point $x_0 = (-1.2, 1)^T$. Terminate the algorithm once $\|\nabla f(x_k)\| \leq 10^{-4}$. Report the objective function value f, the step length α and the norm of ∇f of the **last 10** iterations.

Problem 4: Implement the steepest descent method with the **inexact** step length satisfying the strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 0.1$ to minimize the above Rosenbrock function. Set the initial step length $\alpha_0 = 1$ and the initial point $x^0 = (-1.2, 1)^T$. Terminate the algorithm once $\|\nabla f(x^k)\| \leq 10^{-4}$. Report the objective function value f, the step length α and the norm of ∇f of the **last 10** iterations. Compared with the algorithm implemented in Problem 3, which one is faster?