MATH 309 Fall 2014 − Homework 4

The due date is 11:59 pm, November 14th (Friday). Please submit 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 Algorithm 5.2 (CG) to solve linear system $Ax = b$, 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 $x^0 = 0$. Try $n = 20$ and terminate the algorithm when $\|\nabla f(x^k)\| \le 10^{-4}$, where $f(x) = \frac{1}{2}x^T Ax - b^T x$. Report the function value f, and the norm of ∇f of the **last 10** iterations.

Problem 2: Resolve Problem 1 by applying Algorithm 5.3 (Preconditioned CG) with $M = LL^T$, where LL^T is an incomplete Cholesky factorization of the Hilbert matrix A. (**Hint:** L can be obtained by calling the MATLAB command L=ichol(sparse(A)); L=full(L).)

Problem 3: Implement the nonlinear conjugate gradient methods FR, PR and PR+ to minimize the Rosenbrock function:

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

Choose the step length to satisfy strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 0.1$. Set the initial step length $\alpha_0 = 1$ and the initial point $x_0 = (0, 1)^T$. Terminate the algorithm once $\|\nabla f(x_k)\| \le 10^{-4}$. Report the objective function value f, the step length α and the norm of gradient ∇f of the **last 10** iterations. Compare the results of these methods and also compare them with those obtained by using line search Newton's method in homework 3. (**Note:** The codes for finding the step length to satisfy strong Wolfe conditions with $c_1 = 10^{-4}$ and $c_2 = 0.1$ are posted on Canvas.)

Problem 4: Implement Algorithm 6.1 (BFGS method) 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 $A_{i,j} = 1/(i + j - 1)$. Choose the step length to satisfy strong Wolfe conditions with $c_1 = 10^{-4}$, $c_2 = 0.9$ and the initial step length $\alpha_0 = 1$. Set the initial point $x_0 = 0$ and the initial inverse Hessian approximation $H_0 = I$. Try $n = 20$ and terminate the algorithm once $\|\nabla f(x^k)\| \le 10^{-4}$. Report the objective function value f, the step length α and the norm of gradient ∇f of the **last 10** iterations. Compared with the method in Problem 1, which one is faster? (**Note:** The parameter for strong Wolfe conditions are $c_1 = 10^{-4}$ and $c_2 = 0.9$.)