MATH 693A Advanced Numerical Methods: Computational Optimization
Fall 2024
Dr. George

Homework 3

Problem 1 (40 points)
Implement the *standard CG algorithm*, and use it to solve linear systems in which A is the Hilbert matrix, whose elements are $a_{ij} = 1/(i + j - 1)$. Set the right-hand-side to be all ones $\vec{b}$ = ones(n,1), and the initial point to be the origin $\vec{x}_0$ = zeros(n,1). In the stopping criteria, use $\|r_k\| > 10^{-6}$.

a. For dimensions n = 5, 8, 12, 20, plot the log of the norm of the residual (i.e., $\log_{10}(\|r_k\|)$ against the iteration (on the same figure); stop when the norm is less than 10^{-6}.
b. Present in a table, the number of iterations for n = 5, 8, 12, 20.
c. Compute the condition number for the Hilbert matrices, present in a table the log of the condition number for the Hilbert matrix for $n = 5, 8, 12, 20$. Use log to base 10.

Note: The Hilbert matrix shows up in the normal equations in least squares approximations and is an example of a matrix with a nasty condition number. Note that the Hilbert matrix is a square matrix, therefore a matrix size n denotes an $n \times n$ matrix.

d. Plot the eigenvalues for n = 5, 8, 12, 20 **on the same figure** in order to show the spread of the eigenvalues. The log of the eigenvalues should be on the y-axis (use log to base 10). For each n, label/order the eigenvalues of the matrix from 1 to n, beginning from the lowest to highest eigenvalue. The eigenvalue label should be on the x-axis.
e. Plot the convergence factors against n for the Conjugate Gradient and Steepest Descent for $n = 2, 3, 4, \cdots, 20$. Using the graph, discuss the performance of both methods.

Problem 2 (15 points)
Construct matrices with different eigenvalue distributions (clustered and non- clustered) and apply the Conjugate Gradient (CG) method to them.

(i) Describe how you generated your matrices.
(ii) Comment on whether the behavior of the CG method can be explained from Theorem 5.5 in the text by Nocedal and Wright 2006. Generate a figure similar to Figure 5.4 in the text by Nocedal and Wright 2006.

Problem 3 (30 points)
Program that *Line Search Newton-CG Metho*d (see Lecture 13) and use it to minimize the function

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

Use a sequence $\{\eta_k\}$ that guarantees *super-linear convergence*. Use $\| \nabla f(\boldsymbol{x}_k) \| < 10^{-8}$ as the stopping criteria for your outer optimization algorithm. Use the *backtracking line search* to find the step length α_k^{LS}. Use the initial point: $\boldsymbol{x}_0 = [-1.2, 1]$ and then try another point $\boldsymbol{x}_0 = [2.8, 4]$. Do the following for each of the initial points.

a. Your program should indicate, at every iteration, whether the method encountered negative curvature in the inner iterations (present your results in the table below)

Iteration number	$\boldsymbol{x}_k$	Did it encounter a negative curvature in the inner iteration? **(yes/no)**
1		
2		

b. Plot the log of the size of the objective function against the iteration number (Use log to base 10).

c. Repeat part (a) with a sequence $\{\eta_k\}$ that guarantees *quadratic convergence*.

Problem 4 (10 points)
Find the Cholesky Factorization of the matrix (show work):

$$B = \begin{pmatrix} 1 & 2 & 4 & 7 \\ 2 & 13 & 23 & 38 \\ 4 & 23 & 77 & 122 \\ 7 & 38 & 122 & 294 \end{pmatrix}$$

$$D = \begin{pmatrix} 4 & 14 & 16 \\ 14 & 50 & 58 \\ 16 & 58 & 132 \end{pmatrix}$$