MATH 437: Principles of Numerical Analysis

Prof. Wolfgang Bangerth // Blocker 507D // bangerth@math.tamu.edu
TA: Youli Mao // Blocker 505E // youlimao@math.tamu.edu

Homework assignment 6 – due Thursday 10/10/2013

Problem 1 (Steepest descent iteration). In class, the prof made the claim that for badly conditioned matrices the solution vector x_k of iteration k wiggles back and forth, rather than making one step towards the main axis of the contour lines of the quadratic function $q(y)$ and then going straight towards the minimum. Let us test this claim:

Take a matrix and right hand side for a two-dimensional problem as follows:

$$A = \begin{pmatrix} 10 & 0 \\ 0 & 1 \end{pmatrix}, \qquad b = \begin{pmatrix} 10 \\ 0 \end{pmatrix}.$$

The solution of the linear system $Ax = b$ is $x = (1, 0)$. Generate graphs that show the surface and contours of the function

$$q(y) = \frac{1}{2} y^T A y - y^T b.$$

Next consider the steepest descent iteration. Start from $x^{(0)} = (2, 10)^T$. Perform 100 iterations, where in each iteration you compute

$$t = b - Ax^{(k)}, \qquad \alpha = \frac{t^T t}{t^T A t},$$

and then set $x^{(k+1)} := x^{(k)} + \alpha t$. Plot the iterates $x^{(k)} = (x_1^{(k)}, x_2^{(k)})^T$ in a 2-dimensional plot and connect them by lines to see their convergence.

How many iterations do you need to achieve an accuracy of $\|x^{(k)} - x\|_2 \leq 10^{-4}$? Repeat the experiment where a_{11} and b_1 both have the values 1, 10, 100, 1000, 10000 (all other elements of A and b unchanged), and starting from $x^{(0)} = (2, a_{11})$. Create a table with the condition number of these matrices and how many iterations it takes to achieve above accuracy. **(8 points)**

Problem 2 (CG iteration). Take the ever-same 100×100 matrix and 100-dimensional vector defined by

$$A_{ij} = \begin{cases} 2.01 & \text{if } i = j, \\ -1 & \text{if } i = j \pm 1, \\ 0 & \text{otherwise,} \end{cases} \qquad b_i = \frac{1}{100} \sin\left(\frac{2\pi i}{50}\right).$$

Implement the Conjugate Gradient algorithm and use it to solve $Ax = b$.

Start with a vector $x^{(0)}$ with randomly chosen elements in the range $0 \leq (x^{(0)})_i \leq 1$ (i.e., with elements generated from what the `rand()` function or a similar replacement returns). Run 100 iterations to obtain a vector $x^{(100)}$

and then do it over again to plot $\|x^{(k)} - x^{(100)}\|$ for the first 100 iterations $k = 0 \dots 100$. (Ideally, one would of course like to plot $\|x^{(k)} - x\|$, but the exact solution $x = A^{-1}b$ is not known here.)

If you ran the algorithm for 200 iterations, does the solution still change significantly from what you had after 100 iterations? If not, why? **(8 points)**