Introduction to Numerical Analysis - Math 104B Summer 2011

Final exam 9/2/2011 Date due: Tuesday, September 6th, at 12:30 am

Name:

Problem 1	Problem 2.1	Problem 2.2	Problem 3	Grade

Solve the following 3 problems. You have to write two programs, a function in Problem 1 and a script in Problem 2.b that produces a graph, and write in paper the answers for Problem 2.1 and Problem 3. To get full credit the written answers have to be correct and concise and your programs have to run and produce correct results (Observe that the result of Problem 2.2 will change everytime you run it because random numbers are generated). The use of the textbook, or notes, is allowed. For the computational part, you must write your own program. It is not allowed to solve the problems in groups. There will be no extensions on the due date.

PROBLEM 1 (40 Points) Write a program (`mygs.m`) that implements the reduced QR decomposition using the Gram-Schmidt algorithm seen in class. The input should be a $m \times n$ matrix with $rank(A) = n \leq m$ and the outputs should be two:

1. An $m \times n$ matrix Q with orthonormal columns and
2. An upper triangular matrix R such that $A = QR$.

PROBLEM 2 (80 Points) We want to test our different Least Squares Solvers: normal equations and three types of QR decompositions: MATLAB `qr.m, myqr.m, mygs.m`, using an array of test problems. Suppose we want to fit an arbitrary function $f(t)$ with a polynomial of degree $n-1$ using the values of the function in an arbitrary set of points $z = [z_1, z_2, \ldots, z_m]$.

You are asked to:

1. (20 Points) Write (in paper) the problem as a Least Squares Problem (LSP) $Ax \approx b$. You have to establish clearly how to get the elements of A and b in terms of $f(t)$ and z, you have also to state the relation between the vector of unknowns x and the problem.
2. (60 Points) Write a MATLAB script in which, given the dimensions n and m, you have to

- generate two random vectors $z \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$ (use the MATLAB command `randn`),
- solve the LSP $Ax \approx b$ in four different ways, getting four different answers `x1, x2, x3, x4`, the first three using the QR decomposition, and the last one with the normal equations:
  - MATLAB `qr.m`
  - `myqr.m`
  - `mygs.m`
  - normal equations,
- establish as "exact" solution the one obtained with the MATLAB command[1]

  `>> x=pinv(a)*b`

  and compute the condition number of the matrix (it can be computed for non-square, non-invertible matrices) with the MATLAB command

  `>> cond(a)`
- get the relative error in the solution given by each method.
- Do this for values of `n=5:25` and `m=3*n`
- plot in a log plot (`semilogy`) the four relative errors and the condition number of the matrix times the rounding error (that is, `cond(a)*eps`) as a function of `n`, that is, the graph should have **five curves** in different colors. This should be the only output of the script.

PROBLEM 3 (20 Points) By looking at the results of Problem 2, try to decide if the following sentences between quotes are true or false, the credit will be given for the reasoning. Be clear and concise, please.

1. When solving LSP, the solution is given by the solution of the normal equations (this is true), "therefore the sensitivity (condition number) of the problem to small changes in the initial data goes like $\kappa(A^TA) = \kappa(A)^2$"

   (We haven´t seen the condition number of the LSP in class, but from the results you have to get in the graphs you will be able to give a reasoned answer.)

2. "When solving LSP the best possible results are obtained by using the QR decomposition"

3. "The conditioning of the LSP problems goes like $\kappa(A)$, at least for the interpolation problems seen here"

4. "Householder QR is stable, while Gram-Schmidt QR is not"

[1] This uses the Singular Value Decomposition (SVD), not seen in class, and the command `pinv(a)` computes the Moore-Penrose pseudoinverse of the matrix `a`