Homework #3 **IAM 961, UNH fall 2014**
Due Monday, Oct 20th in lecture

For this homework, turn in print-outs of your functions, three plots for problem 4, and a Matlab-diary-like listing of your tests. The diary should include extensive comments explaining what you are doing and what the tests reveal. Don't turn in printouts of entire matrices to demonstrate comparisons. Instead, if you want to show that X is very nearly equal to Y, compute the norm of the difference between X and Y.

Please do not look at the pseudo-code for the algorithms in Trefethen & Bau. Instead, start from the mathematical expressions for the algorithms in your lecture class notes and devise the code yourself. Use whatever programming language you like. A scripted high-level language with built-in matrix functionality (e.g. Matlab, Octave, Python) will probably be easiest and most revealing. I suggest the following function names: `qr_cgs, qr_mgs, qr_house`, and `backsolve`, and function signatures of the form `[Q,R] = qr_cgs(A)` and `x = backsolve(Q,b)`.

1. Write functions for computing the QR decomposition of a matrix via
(a) classical Gram-Schmidt orthogonalization,
(b) modified Gram-Schmidt orthogonalization, and
(c) Householder triangularization.

Test that your QR algorithms work correctly on fairly small and well-conditioned matrices (e.g. a 20×20 matrix with normally distributed elements, `A = randn(20,20)` in Matlab). You should test that Q is unitary and that $QR \approx A$. Verify to your own satisfaction that R is upper-triangular.

2. Write an `x = backsolve(R,b)` function for backsolving the system $Rx = b$ for upper-triangular matrices R. Test that your backsolve function works by using it in conjunction with one of your QR algorithms to solve a few $Ax = b$ problem.

3. Write a function `A = randommatrix(m, kappa)` that returns an $m\times m$ random matrix with condition number κ and exponentially graded singular values (i.e. $\sigma_1/\sigma_m = \kappa$ and $\sigma_{j+1}/\sigma_j = \text{const}$). You can use the Matlab code at the top of pg 65 in Trefethen and Bau as a starting point.

4. Solve a large number of random $Ax = b$ problems via QR decomposition using the `randommatrix`, `qr_cgs` and `backsolve` functions from problems 1,2 and 3, and produce a scatter plot of the normalized solution error $\|\hat{x} - x\|/\|x\|$ versus κ. To form a random $Ax = b$ problem, construct a random A matrix with $\kappa = 10^n$ where n is a random real-valued number uniformly distributed between 0 and 16. Select a random x vector with `x = randn(m,1)`, and then set $b = Ax$. Compute the numerical solution $\hat{x}$ of $Ax = b$ via QR, and then plot $\|\hat{x} - x\|/\|x\|$ using log-log axes. Do this for few hundred or

thousand random $Ax = b$ problems and for a fairly small value of m (perhaps 10 or 20). Repeat with `qr_mgs` and `qr_house`. Turn in three scatter plots, one for each QR decomp algorithm.

5. Comment on your results. What can you explain about the scatter plots based on the algorithms and their implementation in finite-precision arithmetic? Or, contrariwise, what can you say about the algorithms based on the scatter plots?

6. If you are curious, repeat problem 4 for a different value of m (perhaps $m = 100$). Does the dimensionality of the matrix (the value of m) make any difference?