Math 5553, Homework 2, Due on 2/21/2013

1. (4 points) You may use the facts that $det(AB) = det(A)det(B)$ and $det(A^*) = \overline{det(A)}$.
   (a) What are the singular values of a unitary matrix?
   (b) Let $A \in \mathbb{C}^{m\times m}$ be an invertible matrix with singular value decomposition $A = U\Sigma V^*$. What are the singular values of A^{-1}?
   (c) Prove that for any unitary matrix U, we have $|det(U)| = 1$.
   (d) If $A \in \mathbb{C}^{m\times m}$ has singular values $\sigma_1, \sigma_2, \ldots, \sigma_m$, prove that $|\det(A)| = \sigma_1\sigma_2\cdots\sigma_m$. What if A is a real-valued matrix?

2. (3 points) Find an orthogonal projector that maps all 3-dim vectors onto the plane $2x - 4y + 3z = 0$.

3. (3 points) Prove that the Householder reflection $F = I - 2\frac{\mathbf{v}\mathbf{v}^*}{\mathbf{v}^*\mathbf{v}}$ is unitary.

4. (4 points) Calculate the computational cost of the Householder QR factorization (Algorithm 10.1).

5. (6 points) Matlab command `rand(m,n)` generates an $m \times n$ random matrix. Here we always assume $m \geq n$. To get a random $m \times m$ unitary matrix, one can simply use `[Q,R] = qr(rand(m,n))`, where `qr` is the Matlab built-in routine for QR factorization, or `[Q, S, V] = svd(rand(m,n))`, where `svd` is the Matlab built-in routine for SVD. Then, `Q` returns an $m \times m$ unitary matrix.

   Next, we manually write an upper triangular matrix R. Let `R= diag(r) * triu(ones(m,n))`, where $r = [r_1, r_2, \ldots, r_m]$ is a given m-dim vector. This generates an $m \times n$ matrix R which contains the first n columns of the upper triangular matrix

$$\begin{bmatrix} r_1 & r_1 & r_1 & \cdots & r_1 \\ 0 & r_2 & r_2 & \cdots & r_2 \\ 0 & 0 & r_3 & \cdots & r_3 \\ & & & \cdots & \\ 0 & 0 & 0 & \cdots & r_m \end{bmatrix}$$

   In the following experiment, we will set $r_i = 2^{-i}$, for $i = 1, ...m$, and generate R.

   Now we have an $m \times m$ unitary matrix Q and an $m \times n$ upper triangular matrix R. Define $A = QR$. Such a matrix A can be used to test the stability of different algorithms for computing the QR factorization, because we know its exact QR factorization. We have covered three algorithms in class, the classical Gram-Schmidt, the modified Gram-Schmidt, and the Householder triangulization algorithms. Implement these three algorithms and test them on matrix A, with $m = n = 48$. For each algorithm, you get the computed QR factorization $\tilde{Q}$ and $\tilde{R}$. Calculate and report the matrix 2-norm of $I - \tilde{Q}^*\tilde{Q}$, using Matlab command `norm`. Notice that if $\tilde{Q}$ is exactly a unitary matrix, this norm should be 0.

   Draw the diagonal entries of $\tilde{R}$ using command `semilogy`. (Plot in discrete dots, circles, x-marks or whatever instead of continuous curve. Type `help plot` to learn more about line type options.) Compare the results with the diagonal entries of R. What do you observe for these three algorithms?