CS 6210: Assignment 6 (Revised)

Due: Wednesday, November 3, 2010 (In Class or in Upson 5153 by 4pm)

Scoring for each problem is on a 0-to-5 scale (5 = complete success, 4 = overlooked a small detail, 3 = good start, 2 = right idea, 1 = germ of the right idea, 0 = missed the point of the problem.) One point will be deducted for insufficiently commented code. Unless otherwise stated, you are expected to utilize fully MATLAB'S vectorizing capability subject to the constraint of being flop-efficient. Test drivers and related material are posted at `http://www.cs.cornell.edu/courses/cs6210/2010fa/`. For each problem submit output and a listing of all scripts/functions that *you* had to write in order to produce the output. You are allowed to discuss *background* issues with other students, but the codes you submit must be your own.

P1. (Skew-Symmetric Tridiagonalization)

We say that $A \in \mathbb{R}^{n\times n}$ is skew-symmetric if $A^T = -A$. Note that if this is the case then $x^TAx = 0$ for all $x \in \mathbb{R}^n$. It is possible to compute Householder reflections $P_1, \ldots, P_{n-2}$ such that if $Q = P_1 \cdots P_{n-2}$ then $Q^TAQ = T$ is tridiagonal:

for $k = 1{:}n-2$
 Determine a unit 2-norm vector $v \in \mathbb{R}^{n-k}$ such that
  $(I_{n-k} - 2vv^T)A(k+1{:}n, k)$ is zero except in the first component.

 $A \leftarrow P_k^TAP_k$ where $P_k = \begin{bmatrix} I_k & 0 \\ 0 & I_{n-k} - 2vv^T \end{bmatrix}$.
end

Write an efficient MATLAB function `[Q,T] = SkewReduce(A)` that does this. Note that $Q = Q_1(Q_2(Q_3Q_4))$ is more effcient than $Q = ((Q_1Q_2)Q_3)Q_4$. Also observe that if M is skew-symmetric and u is a vector, then $u^TMu = u^TM^Tu = -u^TMu = 0$. Finally, if you have a low rank skew-symmetric update of the form $M = M + zv^T - vz^T$, then it is okay to use

```
M = M +z*v'-v*z'
```

instead of the more flop-efficient

```
n = length(z)
for i=1:n
   for j=i+1:n
      M(i,j) = M(i,j) + z(i)*v(j) - v(i)*z(j);
      M(j,i) = -M(i,j);
   end
end
```

Test with the script `P1`.

P2. (An Arrow Matrix Eigenproblem)

A matrix of the form

$$A = \text{arrow}(d, v) = \begin{bmatrix} d_1 & 0 & 0 & 0 & v_1 \\ 0 & d_2 & 0 & 0 & v_2 \\ 0 & 0 & d_3 & 0 & v_3 \\ 0 & 0 & 0 & d_4 & v_4 \\ v_1 & v_2 & v_3 & v_4 & d_5 \end{bmatrix}$$

is an *arrow matrix* if the d_i are positive and distinct and the v_i are nonzero. It can be shown that an arrow matrix has a unique dominant eigenvalue $\lambda_{max} > \max\{d_i\}$. Complete the following function so that it performs as specified

```
  function [lambda,x,its] = DomArrow(d,v,tol,itMax)
% d is a column n-vector with distinct positive values
% v is a column (n-1)-vector with nonzero entries
% lambda and x are an approximate dominant eigenpair for A = Arrow(d,v)
% that satisfies norm(A*x - lambda*x,2) <= tol*norm(A,1). It is assumed that
% x has unit 2-norm.
% its is the number of required iterations until convergence. itMax is an
% upper bound on the number of iterations.
```

Your implementation should make effective use of the power method. In choosing your starting vector, assume that $\| d \|$ is considerably larger than $\| v \|$. Test your implementation with `P2`.

P3. (Eigenvalues of an Orthogonal Matrix)

The eigenvalues of a real orthogonal matrix are on the unit circle. Suppose $Q \in \mathbb{R}^{n\times n}$ is orthogonal and $Qx = \lambda x$. It follows that

$$\frac{Q+Q^T}{2}x \;=\; \frac{1}{2}\left(Q+Q^{-1}\right)x \;=\; \frac{1}{2}\left(\lambda+\frac{1}{\lambda}\right)x \;=\; \frac{1}{2}\left(\lambda+\bar{\lambda}\right)x \;=\; \mathrm{Re}(\lambda)x$$

Complete the following function so that it performs as specified:

```
 function nPos= OrthoEig(Q)
% Q is an n-by-n real orthogonal matrix with distinct eigenvalues mu_1,...,mu_n.
% nPos is the number of Q's eigenvalues that are strictly to the right of the
% imaginary axis.
```

Test your implementation with the script `P3`.

C6. (The Centrosymmetric Eigenvalue Problem)

Let E_n be the n-by-n exchange matrix, i.e.,

$$E_3 \;=\; \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

A matrix $A \in \mathbb{R}^{n\times n}$ is *centrosymmetric* if $A = A^T$ and $A = E_n A E_n$. If $n = 2m$ and

$$A \;=\; \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

is centrosymmetric, then it is easy to verify that $A_{22} = EA_{11}E$ and $A_{21} = EA_{12}E$ where $E = E_m$. Note that if $(A_{11} + A_{12}E)y = \lambda y$, then

$$\begin{bmatrix} A_{11} & A_{12} \\ EA_{12}E & EA_{11}E \end{bmatrix}\begin{bmatrix} y \\ Ey \end{bmatrix} \;=\; \lambda \begin{bmatrix} y \\ Ey \end{bmatrix}$$

while $(A_{11} - A_{12}E)y = \lambda y$ implies

$$\begin{bmatrix} A_{11} & A_{12} \\ EA_{12}E & EA_{11}E \end{bmatrix}\begin{bmatrix} y \\ -Ey \end{bmatrix} \;=\; \lambda \begin{bmatrix} y \\ -Ey \end{bmatrix}.$$

It follows that A has a Schur decomposition $Q^TAQ = \mathrm{diag}(\lambda_i)$ where Q has the form

$$Q \;=\; \frac{1}{\sqrt{2}}\begin{bmatrix} Q_{11} & Q_{12} \\ EQ_{11} & -EQ_{12} \end{bmatrix}$$

Thus, the Schur Decomposition for A can be "built-up" from the half-sized Schur Decompositions for $A_{11} + A_{12}E$ and $A_{11} - A_{12}E$. Work out the corresponding case when $n = 2m + 1$. Hint: If

$$A = \begin{bmatrix} A_{11} & v & A_{13} \\ v^T & \alpha & v^T E \\ A_{31} & Ev & A_{33} \end{bmatrix}$$

then you can build Q and D from the from the Schur decompositions of $A_{11} - A_{13}E$ and

$$\begin{bmatrix} A_{11} + A_{13}E & \sqrt{2}v \\ \sqrt{2}v^T & \alpha \end{bmatrix}$$

Implement a function

```
function [Q,D] = CentroSymSchur(A)
```

that makes efficient use of the MATLAB function `schur` to compute the structured Schur decomposition of a centrosymmetric matrix. Clearly state the form of Q. Test your implementation with the script `C6`.