CS 4220: Assignment 4

Due: Friday, March 19, 2010 (or via single pdf by Tuesday Mar 23)

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 on the course website `http://www.cs.cornell.edu/courses/cs4220/2010sp/`. 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. Submit listing and the output when the test script `P1` is applied. A function `v = House(x)` is available on the website.

P2. (Special Schur Decomposition)

Suppose $A = I_n + XBX^T$ where $X \in \mathbb{R}^{n\times p}$ $(p < n)$ and $B \in \mathbb{R}^{p\times p}$ is symmetric. Note that A has at most p eigenvalues that do not equal one. Thus, if $Q^TAQ = T$ is A's tridiagonalization, then T will have a lot of zeros along its subdiagonal. One could exploit this property when getting T's eigenvalues. A better approach starts by computing X's QR factorization. Think about this and implement the following function accordingly.

```
  function [Q,D] = SpecialSchur(X,B)
% X is an n-by-p matrix, 1<=p<n
% B is a symmetric p-by-p matrix
% A = I + X*B*X'
% Q'*A*Q = D is the schur decomposition of A.
```

Test your implementation on the script `P2.m` that is available on the website. Submit output and listing of `SpecialSchur`.

P3. (A Rank-1 Adjustment Eigenproblem)

If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite and $v \in \mathbb{R}^n$, then $A+vv^T$ is positive definite. Making effective use of the MATLAB function `schur` and the power method, complete the following function so that it performs as specified.

```
 function [Q,D,mu,x] = SchurPlus(A,v)
% A is an n-by-n symmetric positive definite matrix.
% v is a column n-vector
% Q'*A*Q = D is the Schur decomposition of A.
% mu is the largest eigenvalue of A + v*v' and x is an associated eigenvector,
% normalized so  norm(x,2) = 1. Assume that the largest eigenvalue of A and
% A + v*v' are unique
```

Compute the Schur decomposition of A first and then go after x and μ using the power method. In the power method, exploit as much as possible the fact that you have the Schur decomposition of A. (Hint: Think about a change of variable via Q and think about a good starting vector.) Terminate the power iteration when

$$\| (A + vv^T)x^{(k)} - \mu^{(k)}x^{(k)} \|_2 \leq 10^{-14} \left(\| A + vv^T \|_\infty \right)$$

where x^k is the k-th (unit vector) iterate and $\mu^{(k)} = x^{(k)T}(A + vv^T)x^{(k)}$.

Test your implementation on the script `P3`. Submit output and a listing of `SchurPlus`.

P4. (A Constrained Max Problem)

Consider the function ϕ:$\mathbb{R} \rightarrow \mathbb{R}$ defined by

$$\phi(x) \;=\; \frac{x^T Ax}{x^T x}$$

where $A \in \mathbb{R}^{n \times i}$s symmetric. The function ϕ is maximized by setting x to be an eigenvector associated with A's most positive eigenvector. Complete the following function so that it performs as advertised:

```
function [xMax,phiMax] = QuadForm(A)
% A is n-by-n and symmetric.
% xMax is a column vector that maximizes x'*A*x/x'*x  subject to the constraint
% that x(1) + ... + x(n) = 0.
% phiMax = xMax'*A*xMax/xMax'*xMax.
```

Submit a listing of `QuadForm` and the output that is obtained when the script `P4` is run.

Hint. Note that the constraint says that we must optimize over all x that are orthogonal to $e = ones(n, 1)$. If $Q \in \mathbb{R}^{n \times n}$ is orthogonal and $Q^T e = \sqrt{n} I_n(:, 1)$, then the last $n-1$ columns of Q define the search subspace. Our goal amounts to finding $z \in \mathbb{R}^{n-1}$ so that $\phi(Q(:, 2{:}n)z)$ is maximized.