CS 621: Assignment 2

Due: Wednesday, October 2, 2002 (In Lecture or 4130 Upson by 4pm)

Scoring on each problem is on a 0-1-2-3 scale. 3 = complete success, 2 = overlooked a small point, 1 = germ of the right idea, 0 = missed the point of the problem. One point will be deducted for insufficiently commented code. Test drivers and related material will be posted on the course website. For each problem submit output and a listing of all the scripts/functions that you had to write/modify 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.

Problem 1. Consider the n-by-n matrix

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

where $A_{11} \in \mathbb{R}^{p\times p}$, $A_{12} \in \mathbb{R}^{p\times m}$, $A_{21} \in \mathbb{R}^{m\times p}$, and $A_{22} \in \mathbb{R}^{m\times m}$. Assume that A^T is strictly diagonally dominant. Write a MATLAB function `z = SchurCompSolver(A,c)` that solves $(A_{22} - A_{21}A_{11}^{-1}A_{12})z = c$ where $c \in \mathbb{R}^m$. Make effective use of the MATLAB `LU` function. You may use \ to solve triangular systems. Test your implementation by running the script `P1`. Hint: Spend some time thinking about the following block matrix equation:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}.$$

Problem 2. Develop a MATLAB function `[kappaHat,y] = CondEst(T)` that implements Algorithm 3.5.1. Note that the algorithm tries to pick a unit ∞-norm vector y so that $\| Ty \|_\infty$ has small norm.

Write a script `P2` that sheds light on the reliability of `CondEst`. It should run `CondEst` on a range of random upper triangular matrices. Report on average values and/or worst-case values of the quantity $\mu(T) = \hat{\kappa}/\kappa_\infty(T)$. How does this quantity vary with n and actual condition? Your script should produce tables and/or graphs that speak to the issue. For your information, here is a script that generates an upper triangular matrix with 2-norm condition 10^i.

```
D = diag(logspace(0,-i,n);
[U,R] = qr(randn(n,n));
[V,R] = qr(randn(n,n));
[Q,T] = qr(U*D*V');
```

Problem 3. Complete the following function so that it performs as advertised:

```
function X = CollectionSolve(A,B)
% A is an m-by-4 matrix and B is an m-by-2 matrix.
% X is an m-by-2 matrix with the property that for i=1:m
% [X(i,1);X(i,2)] = [A(i,1) A(i,2) ; A(i,3) A(i,4) ] \ [B(i,1) ; B(i,2)]
% Assume that these linear systems are nonsingular.
```

Thus, `CollectionSolve` uses Gaussian elimination with pivoting to solve the 2-by-2 linear systems

$$\begin{bmatrix} A_{i1} & A_{i2} \\ A_{i3} & A_{i4} \end{bmatrix} \begin{bmatrix} X_{i1} \\ X_{i2} \end{bmatrix} = \begin{bmatrix} B_{i1} \\ B_{i2} \end{bmatrix} \qquad i = 1{:}m$$

Your implementation should involve NO loops. Test your implementation by running the script `P3`. Hint: Look at 2-by-2 Gaussian elimination with pivoting and vectorize. You'll need to use the `find` function.

Problem 4. Suppose $A \in \mathbb{R}^{n\times n}$ is such that A^T is strictly diagonally dominant. Let e_k be the k-th column of I_n and define the function

$$f_k(\alpha) = \det(A + \alpha e_k e_k^T)$$

Note that the matrix is just A with α added to the kth diagonal entry. Write a MATLAB function `beta = DerDet(A,k)` that returns $f_k'(0)$. You may want to make use of these facts which are based on the factorization $A = LU$:

- $\det(A) = \det(U) = u_{11} \cdots u_{nn}$.
- If $L(\alpha)U(\alpha) = A + \alpha e_k e_k^T$ is the LU factorization, then by taking $d/d\alpha$ of both sides we get

$$\dot{L}U + L\dot{U} = e_k e_k^T \qquad \leftrightarrow \qquad L^{-1}\dot{L} + \dot{U}U^{-1} = L^{-1}e_k e_k^T U^{-1}$$

- $\dot{L}$ has a zero diagonal.

Test your implementation by running the script `P4`.