CS 621: Assignment 3

Due: Friday, October 14, 2005 (In Lecture or 4130 Upson by 4pm)

Scoring for each problem is on a 0-to-3 scale (3 = complete success, 2 = overlooked a small detail, 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/cs621/2005fa/`. 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. (Unsymmetric Positive Definite Tridiagonal Systems)

Refer to GVL §4.2 where the stability of the factorization $A = LDM^T$ is discussed for the case when A is positive definite but not symmetric. The quantity

$$\Omega = \frac{\| ST^{-1}S \|}{\| A \|} \qquad T = \frac{A + A^T}{2},\ S = \frac{A - A^T}{2}$$

is critical. This problem is about estimating the numerator of this quotient when A is tridiagonal. In particular, complete the following function so that it performs as specified.

```
  function omega = MaxElement(a,c,f)
% a is a column n-vector, c and f are column (n-1)-vectors.
% Let A = diag(c,-1) + diag(a) + diag(f,1)and assume T = (A+A')/2 is positive definite.
% omega is the largest entry in abs( S*inv(T)*S ) where S = (A-A')/2.
```

Test your implementation with the script `P1`.

P2. (Block Toeplitz System Solver)

Suppose $R_1, \ldots, R_p$ are given m-by-m matrices. For $k = 1{:}p$ define $T_k \in \mathbb{R}^{km \times km}$ by

$$T_k = \begin{bmatrix} I_m & R_1 & R_2 & R_3 & R_4 \\ R_1 & I_m & R_1 & R_2 & R_3 \\ R_2 & R_1 & I_m & R_1 & R_2 \\ R_3 & R_2 & R_1 & I_m & R_1 \\ R_4 & R_3 & R_2 & R_1 & I_m \end{bmatrix} \qquad (k = 5)$$

These *block Toeplitz* matrices are *block symmetric.* (They are not actually symmetric unless the R_i are symmetric.) Define the *block exchange* permutation E_k by

$$E_k = I_k(:, k{:}-1{:}1) \otimes I_m = \begin{bmatrix} 0 & 0 & 0 & 0 & I_m \\ 0 & 0 & 0 & I_m & 0 \\ 0 & 0 & I_m & 0 & 0 \\ 0 & I_m & 0 & 0 & 0 \\ I_m & 0 & 0 & 0 & 0 \end{bmatrix} \qquad (k = 5)$$

Notice that the T_k are *block persymmetric* ($T_k = E_k T_k E_k$) and that

$$T_{k+1} = \begin{bmatrix} T_k & E_k R \\ R^B E_k & I_m \end{bmatrix}$$

where

$$R = \begin{bmatrix} R_1 \\ R_2 \\ \vdots \\ R_k \end{bmatrix}$$

and

$$R^B = \begin{bmatrix} R_1 & R_2 & \cdots & R_k \end{bmatrix} \qquad \text{(block transpose)}$$

Assume that T_p is strictly diagonally dominant so that all the T_k are nonsingular.

By generalizing the derivations in GVL §4.7, figure out how to solve

$$T_{k+1} \begin{bmatrix} Z \\ A \end{bmatrix} = -\begin{bmatrix} R \\ R_{k+1} \end{bmatrix} \qquad Z \in \mathbb{R}^{km\times m},\ A \in \mathbb{R}^{m\times m}$$

efficiently given that you have the solution to $T_k Y = -R$. Likewise, figure out how to solve efficiently the linear system

$$T_{k+1} \begin{bmatrix} w \\ \mu \end{bmatrix} = -\begin{bmatrix} b \\ b_{k+1} \end{bmatrix} \qquad w, b \in \mathbb{R}^{km},\ \mu, b_{k+1} \in \mathbb{R}^m$$

given that you have the solution to $T_k x = b$.

By exploiting these ideas, implement the following function so that it performs as specified:

```
  function x = BlockToeplitz(R,b)
% R is a length p cell array with each R{i} an m-by-m matrix.
% b is a pm-by-1 vector.
% x solves Tx = b where T is a block Toeplitz matrix with
%                  T(1:m,:) = [eye(m,m) R{1} R{2} ... R{p-1}]
% Assume that p>>m and that T is strictly diagonally dominant.
```

It should involve $O(p^2m^2)$ flops and $O(pm^2)$ storage. Test your implementation with the script P2.

Go-the-Distance 3. (Hostile Pivoting)

Suppose $A \in \mathbb{R}^{n\times n}$ is symmetric but possibly indefinite and consider the factorization

$$P_1AP_1^T = \begin{bmatrix} E & C^T \\ C & B \end{bmatrix} = \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix}\begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{bmatrix}\begin{bmatrix} I_s & E^{-1}C^T \\ 0 & I_{n-s} \end{bmatrix}$$

where $s = 1$ or $s = 2$ and P_1 is a permutation. In "classical" symmetric pivoting, $s = 1$ and and P_1 is chosen so that E is maximal. In other words, the most positive diagonal entry is brought up to the (1,1) position. In "hostile" symmetric pivoting $s = 1$ and P_1 is chosen so that E is minimal. In other words, the least positive diagonal entry is brought up to the (1,1) position. If E is negative then we know A is indefinite and there exists a unit 2-norm vector x so $x^TAx < 0$. If E is positive then by choosing it as small as possible encourages the update $B - CE^{-1}E$ to have small (possibly negative) diagonal elements. Thus, hostile symmetric pivoting is a heuristic way of exposing indefiniteness as soon as possible as the factorization unfolds.

Develop a hostile version of the Bunch-Kaufman pivot strategy that is discussed in GVL p.169. The idea here is to choose efficiently P_1 so that E has small, possibly negative eigenvalues. Note that if E has a negative eigenvalue then A is indefinite. Using these ideas, implement the following function so that it performs as advertised

```
  function x = NegVec(A)
% A is an n-by-n symmetric matrix.
% If A is indefinite, then x is a unit 2-norm column n-vector so x'*A*x < 0.
% Otherwise, x is the empty vector.
```

Submit implementation electronically. With the rest of your assignment, submit a hardcopy write-up of your method with justifications.