CS 6210: Assignment 3

Due: Monday, September 27, 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. (I-Plus-Triagonal Skew Solver) Read about unsymmetric positive definite systems in §4.2.2. Write a MATLAB function

```
function [x,relerr] = iSkewSolve(b,c)
```

that solves systems of the form

$$Ax = \begin{bmatrix} 1 & c_1 & 0 & 0 & 0 \\ -c_1 & 1 & c_2 & 0 & 0 \\ 0 & -c_2 & 1 & c_3 & 0 \\ 0 & 0 & -c_3 & 1 & c_4 \\ 0 & 0 & 0 & -c_4 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}$$

Your method should work with the factorization $A = LU$. Do not pivot, do not use the MATLAB `lu` function, and do not use \. The input vectors `b` and `c` should be column vectors. The value of output parameter `relerr` should provide an estimate of $\| x - \hat{x} \| / \| x \|$ where $\hat{x}$ is the computed solution. Submit listing and the output when the test script `P1` is executed.

P2. (A Symmetric Indefinite Problem) If A is symmetric but indefinite, then the MATLAB function `ldl` function offers a Cholesky-speed method for solving $Ax = b$. It computes a permutation P, a lower triangular L, and a block diagonal D so $P^TAP = LDL^T$. The matrix D is block diagonal with 1-by-1 and 2-by-2 diagonal blocks. Here is the solution framework:

```
[L,D,P] = ldl(A)
w = L'\(P'*b);
z = D\w;
y = L\z;
x = P*y
```

Now suppose $A \in \mathbb{R}^{n \times n}$ has $n-1$ positive eigenvalues and one negative eigenvalue. It follows that a unit vector x exists so $x^TAx < 0$. In this problem you are to write a MATLAB function `x = FindNegVec(A)` that returns such a vector. The factorization $P^TAP = LDL^T$ is relevant because by applying Sylvestor's Law of Inertia, we can assert that exactly one of D's eigenvalues is negative. This suggests that `FindNegVec` can be organized as follows:

- Compute $P^TAP = LDL^T$ using `ldl`.
- Find a nonzero vector v so $v^TDv < 0$.
- Use v to find x.

In step 2 you will have to find that block of D that has the negative eigenvalue. If $D(k,k)$ is negative and a 1-by-1 block, then $e_k^T D e_k = D(k,k) < 0$ where $e_k = I_n(:,k)$. But if the negative eigenvalue sits in a 2-by-2 block, then you have a little work! You might want to use `eig`. Submit a listing of your implementation of `FindNegVec` and the output from the test script `P2`.

P3. (The Diagonal of a Cholesky Factor) Suppose $u \in \mathbb{R}^n$ is nonzero and that $GG^T = I_n + uu^T$ is the Cholesky factorization of the symmetric positive definite matrix $I_n + uu^T$. Write a function `g = CholDiag(u)` that returns the diagonal of G in the column n-vector g. For full credit, your implementation must require just $O(n)$ flops.

Start by considering this equation:

$$\begin{bmatrix} G_1 & 0 \\ v^T & g_{k+1,k+1} \end{bmatrix} \begin{bmatrix} G_1 & 0 \\ v^T & g_{k+1,k+1} \end{bmatrix}^T = \begin{bmatrix} I_k + ww^T & u_{k+1}w \\ u_{k+1}w^T & 1 + u_{k+1}^2 \end{bmatrix}$$

where $G_1 = G(1{:}k, 1{:}k)$, $v^T = G(k+1, 1{:}k)$, and $w = u(1{:}k)$. Develop a recipe for $g_{k+1,k+1}$ that does not require G_1. You might find the Sherman Morrison formula useful.

Submit your implementation of `CholDiag` together with the output when the test script `P3` is applied.

Challenge Problem 3. The Block Diagonal Part of a Cholesky Factor)

Suppose $U \in \mathbb{R}^{n \times r}$ has full column rank, $n = mr$, and $r << n$. Partition the Cholesky factorization of $A = I + UU^T$ as follows

$$I + UU^T = \begin{bmatrix} G_{11} & 0 & \cdots & 0 \\ G_{21} & G_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{m1} & G_{m2} & \cdots & G_{mm} \end{bmatrix} \begin{bmatrix} G_{11} & 0 & \cdots & 0 \\ G_{21} & G_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ G_{m1} & G_{m2} & \cdots & G_{mm} \end{bmatrix}^T$$

Write a MATLAB function `Gblocks = CholBlockDiag(U)` that returns in the m-by-1 cell array `GBlocks` the diagonal blocks $G_{11}, \ldots, G_{mm}$. Submit listing and the output from the test script `C3`.