Math 170A, Fall 2022 HOMEWORK #2 due Friday, October 20

Homework problems that will be graded (Q1 - Q5, 30pts in total):

Q1. Let A be as follows:

$$A = \begin{bmatrix} 0 & 1 & 1 & 2 \\ 2 & 0 & 0 & 0 \\ 1 & 1 & 1 & 3 \\ 1 & 0 & 2 & 4 \end{bmatrix}$$

Does the matrix A have an LU decomposition? If yes, compute the LU decomposition. Otherwise, explain why it does not have one, and then compute the PLU decomposition of A (as we learned in class, swapping the rows each time to bring the largest entry to the diagonal).

Q2. Determine whether or not the following matrices have a Cholesky factorization; if they do, compute (by hand) the Cholesky factor R:

$$A = \begin{bmatrix} 1 & -2 & 0 \\ -2 & 13 & 6 \\ 0 & 6 & 5 \end{bmatrix}, \quad B = \begin{bmatrix} 4 & -4 & 0 \\ -4 & 4 & 0 \\ 0 & 0 & 5 \end{bmatrix}$$

Q3. Let A be an $n \times n$ positive definite and symmetric matrix, and X an $n \times n$ invertible matrix. Show that $B = X^T AX$ is positive definite and symmetric.

Q4. Let P be an $n \times n$ permutation matrix, corresponding to a permutation p of $\{1, 2, \ldots, n\}$.

In class, we showed that, given any matrix A which is $n \times n$, PA permutes the rows of A according to p.

a) Show that AP^T permutes *the columns* of A according to p. **Hint:** you could use transpose properties.

b) Conclude that $PP^T = I_n$.

Q5. Write a MATLAB code that does Gauss elimination with partial pivoting (row switching). The input to your code should be

- an $n \times n$ matrix A
- a column vector b of size $n \times 1$

and the output should be the solution x to $Ax = b$.

You can use the Gauss elimination code (ge_solve.m), the backward substitution code (BackSub.m) and include the row switching into the first loop (i-loop) as we did in class for PLU. The functions ge_solve.m and BackSub.m are on the next page.

```
function x = ge_solve(A,b)

n = size(A,1);

if (size(A,2) ~=n) || (size(b,1) ~=n) || (size(b,2) ~=1)
    error('cannot solve this system')
end

for i=1:n
    if (A(i,i) == 0)
        error('cannot do GE without swaps')
    end

    for j=(i+1):n
        l = A(j,i)/A(i,i);
        A(j,i) = 0;
        for k=(i+1):n
            A(j,k) = A(j,k) - l*A(i,k);
        end
        b(j) = b(j) - l*b(i);
    end

    U = A;

    x = BackSub(U,b);
```

```
function x = BackSub(A,b)

n = size(A,1);

if (size(A,2) ~= n) || (size(b,1) ~=n) || (size(b,2) ~=1)
    error('cannot solve')
end

if(tril(A) ~= zeros(n))  %checking that U is upper triangular
    error('input matrix not upper triangular')
end

x = b;
for i = n:-1:1
    for j=(i+1):n
        x(i) = x(i) - A(i,j)*x(j);
    end;
    if (A(i,i) == 0)
        error('input matrix not invertible')
    end
    x(i) = x(i)/A(i,i);
end;
```