MATH 8600 (FALL 2018) HOMEWORK 3

Instructor: Dr. Fei Xue, Martin O-203, fxue@clemson.edu.

Assigned 09/28/18, due 10/09/18 in class.

1. (By hand) Let $A = \begin{bmatrix} 2 & 1 & 1 & 4 \\ -3 & -1 & 3 & 2 \\ -5 & -1 & 2 & 5 \\ 4 & 2 & 3 & 1 \end{bmatrix}$, and $b = \begin{bmatrix} 4 \\ 3 \\ 8 \\ 1 \end{bmatrix}$.

   (a) Solve the linear system $Ax = b$ by Gaussian elimination without pivoting.

   (b) Find the LU factorization of A without pivoting, then solve $Ax = b$ by two triangular linear solves.

2. (By hand) Find the LU factorization of A in Problem 1 with partial pivoting. Give detailed work as shown in class, and specify the matrices L_1, L_2, L_3, P_1, P_2, P_3, $\widetilde{L}_1$, $\widetilde{L}_2$, $\widetilde{L}_3$, $L = \widetilde{L}_1^{-1}\widetilde{L}_2^{-1}\widetilde{L}_3^{-1}$ and $P = P_3P_2P_1$. Make sure that they satisfy $PA = LU$.

3. (MATLAB) Implement MATLAB code for LU factorization without and with partial pivoting, respectively (let them be `HW3_lu.m` and `HW3_lupp.m`). Test your code on Problems 1 and 2 to verify the results. Then run the following MATLAB commands

```
n = 9;
u = -cos((0:n)/n*pi);
A = vander(u);
[L1,U1] = HW3_lu(A);        (call your LU factorization without pivoting)
[L2,U2,P2] = HW3_lupp(A);   (call your LU factorization with partial pivoting)
display(norm(A-L1*U1)/norm(A));
display(norm(P2*A-L2*U2)/norm(A));
display([max(abs(nonzeros(L1))) max(abs(nonzeros(U1)))]);
display([max(abs(nonzeros(L2))) max(abs(nonzeros(U2)))]);
```

   Now let $n = 19, 29$ and 39 and repeat the experiments. What do you see? In exact arithmetic, we should have $A = LU$ (without pivoting) and $PA = LU$ (with pivoting), but do they hold numerically for each n? Is it a good idea to solve a generic linear system by LU factorization without pivoting, and why?

4. (MATLAB) For certain matrices, LU factorization without pivoting is stable (i.e., no pivoting is needed); for some other matrices, LU factorization with partial pivoting is not stable (i.e., more stable pivoting is necessary).

   (a) Starting with Problem 3, after the command `A = vander(u)`, add the command `A = A+diag([5:5+n]')` and repeat the experiment without pivoting. What do you see? Read the textbook on "Matrices requiring no pivoting" on textbook page 109, and numerically verify whether the updated matrices are diagonally dominant.

   (b) Consider the matrix A of order $n = 100$ in Example 5.9, on page 109. Define `b = A*ones(n,1)`, and solve the linear system $Ax = b$ by MATLAB's backslash directly, and then by your code `HW3_lupp.m`. How large are $\frac{\|PA-\hat{L}\hat{U}\|}{\|A\|}$ and $\frac{\|b-A\hat{x}\|}{\|b\|}$?

(c) Use the uploaded code `HW3_lucp.m` (LU factorization with complete pivoting) to factorize A in (b), which computes matrices satisfying $PAQ = LDU$, where P and Q are permutation matrices, L and U are unit lower and upper triangular, and D is diagonal. Use these matrices to solve $Ax = b$ (need to explain how to do this in details), and report $\frac{\|PAQ-\hat{L}\hat{D}\hat{U}\|}{\|A\|}$ and $\frac{\|b-A\hat{x}\|}{\|b\|}$.

(d) Draw conclusions from part (a)(b)(c).

5. (By hand) For matrices of limited bandwidth, the arithmetic cost for the LU factorization can be reduced considerably.

   (a) Assume that A is such that all its entries below the k-th subdiagonal are zero, and all entries elsewhere (on and above the k-th subdiagonal) are nonzero. Evaluate the arithmetic cost for the LU factorization for A.

   (b) Assume that A is such that all its entries below the k-the subdiagonal and above the k-the super diagonal are zero, and all entries elsewhere are nonzero. Evaluate the arithmetic cost for the LU factorization for A.

6. (By hand and MATLAB) About Cholesky factorization

   (a) Show that the diagonal elements of a real symmetric positive definite (SPD) matrix must be positive.

   (b) Why are the elements in the L factor uniformly bounded without using pivoting?

   (c) Evaluate the arithmetic cost of Cholesky factorization of a full SPD matrix.

   (d) Implement your MATLAB code of Cholesky factorization. Let A be constructed from the command `A = delsq(numgrid('S',5))` and show your computed L factor.