Math 1080: Spring 2011

Homework #6

SOLUTIONS

Problem 1:

Let **A** and **B** be two nonsingular lower triangular $m \times m$ matrices. Show that the product **AB** is also lower triangular.

SOLUTION :

The matrix **A** is lower triangular if and only if $a_{ij} = 0$ whenever $i < j$. Similarly, for **B**, $b_{ij} = 0$ whenever $i < j$. The elements of the product **C** = **AB** obey $c_{ij} = \sum_{k=1}^{m} a_{ik} b_{kj}$

Suppose that $i < j$. Then $c_{ij} = \sum_{k=1}^{i} a_{ik} b_{kj} + \sum_{k=i+1}^{j-1} a_{ik} b_{kj} + \sum_{k=j}^{m} a_{ik} b_{kj}$. In the first sum, $k < j$ and hence $b_{kj} = 0$. In the second sum $i < k < j$ and hence $a_{ik} = b_{kj} = 0$. In the third sum $i < k$ and hence $a_{ik} = 0$. Thus, $c_{ij} = 0$ whenever $i < j$ and the product **AB** is lower triangular.

Problem 2:

Compute the LU factorization with partial pivoting, (i.e., find **P**, **L**, **U** such that $\mathbf{PA} = \mathbf{LU}$) for the following matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 \\ 3 & -2 & -7 \\ 3 & 2 & -1 \end{bmatrix}$$

SOLUTION 1:

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & -2 & -7 \\ 1 & 1 & 1 \\ 3 & 2 & -1 \end{bmatrix}, \quad \mathbf{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1/3 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}, \quad \mathbf{L}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & -2 & -7 \\ 0 & 5/3 & 10/3 \\ 0 & 4 & 6 \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & -2 & -7 \\ 0 & 4 & 6 \\ 0 & 5/3 & 10/3 \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -5/12 & 1 \end{bmatrix},$$

$$\mathbf{U} = \mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & -2 & -7 \\ 0 & 4 & 6 \\ 0 & 0 & 5/6 \end{bmatrix}, \quad \mathbf{P} = \mathbf{P}_2\mathbf{P}_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{L} = \mathbf{P}_2^{-1}\mathbf{L}_1^{-1}\mathbf{P}_2^{-1}\mathbf{L}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1/3 & 5/12 & 1 \end{bmatrix}$$

SOLUTION 2:

$$\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & 2 & -1 \\ 3 & -2 & -7 \\ 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{L}_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1/3 & 0 & 1 \end{bmatrix}, \quad \mathbf{L}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & 2 & -1 \\ 0 & -4 & -6 \\ 0 & 1/3 & 4/3 \end{bmatrix}$$

$$\mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & 2 & -1 \\ 0 & -4 & -6 \\ 0 & 1/3 & 4/3 \end{bmatrix}, \quad \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/12 & 1 \end{bmatrix},$$

$$\mathbf{U} = \mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A} = \begin{bmatrix} 3 & 2 & -1 \\ 0 & -4 & -6 \\ 0 & 0 & 5/6 \end{bmatrix}, \quad \mathbf{P} = \mathbf{P}_2\mathbf{P}_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{L} = \mathbf{P}_2^{-1}\mathbf{L}_1^{-1}\mathbf{P}_2^{-1}\mathbf{L}_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1/3 & -1/12 & 1 \end{bmatrix}$$

Computer Assignment 4:

a) Write a MATLAB function **`[L,U,P]=gausspivot(A)`** that computes the LU factorization of a square $m \times m$ matrix **A** using Gaussian elimination with partial pivoting. The output variables are a lower triangular $m \times m$ matrix **L**, an upper triangular $m \times m$ matrix **U**, and an $m \times m$ permutation matrix **P**.

b) For the following matrix

$$\mathbf{A} = \begin{bmatrix} -8 & -9 & 7 & 19 & 3 \\ 9 & 10 & 9 & 1 & -16 \\ 7 & 2 & -8 & -2 & 3 \\ 19 & 8 & -18 & -8 & -3 \\ 15 & 10 & 16 & -16 & -18 \end{bmatrix}$$

compute two LU factorizations: (1) using Gaussian elimination *without* pivoting **`gauss`**, and (2) using Gaussian elimination with partial pivoting **`gausspivot`**. Compute the relative accuracy of each method and compare them :

`Delta1 = norm(L*U-A)/norm(A);` in case (1)

`Delta2 = norm(L*U-P*A)/norm(A);` in case (2)

PROGRAM:

```
function [L,U,P] = gausspivot(A)
m = length(A);
L = eye(m);
P = eye(m);
for k = 1:m-1
    i = k;
    for j = k+1:m
        if abs(A(j,k)) > abs(A(i,k))
            i = j;
        end
    end
    [A(k,k:m), A(i,k:m)] = swap(A(k,k:m), A(i,k:m));
    [P(k,:), P(i,:)] = swap(P(k,:), P(i,:));
    [L(k,1:k-1), L(i,1:k-1)] = swap(L(k,1:k-1), L(i,1:k-1));
    for j = k+1:m
        L(j,k) = A(j,k)/A(k,k);
        A(j,k:m) = A(j,k:m) - L(j,k)*A(k,k:m);
    end
end
U = A;

function [y,x] = swap(x,y)
```

OUTPUT:

```
>>A = [-8 -9 7 19 3; 9 10 9 1 -16; 7 2 -8 -2 3; 19 8 -18 -8 -3; 15 10
16 -16 -18]

A =

    -8    -9     7    19     3
     9    10     9     1   -16
     7     2    -8    -2     3
    19     8   -18    -8    -3
    15    10    16   -16   -18

>>[L,U] = gauss(A)

L =

    1.0000         0         0         0         0
   -1.1250    1.0000         0         0         0
   -0.8750   47.0000    1.0000         0         0
   -2.3750  107.0000    2.2730    1.0000         0
   -1.8750   55.0000    1.1308 -692.7727    1.0000


U =

   1.0e+03 *

   -0.0080   -0.0090    0.0070    0.0190    0.0030
         0   -0.0001    0.0169    0.0224   -0.0126
         0         0   -0.7950   -1.0370    0.5990
         0         0    0.0000    0.0001   -0.0065
         0         0         0         0   -4.4988

>>Delta1 = norm(L*U-A)/norm(A)

Delta1 =

   3.5604e-12

>>[L,U,P] = gausspivot(A)

L =

    1.0000         0         0         0         0
    0.4737    1.0000         0         0         0
    0.7895    0.5932    1.0000         0         0
   -0.4211   -0.9068    0.7729    1.0000         0
    0.3684   -0.1525    0.0659    0.0844    1.0000
```

```
U =

   19.0000    8.0000  -18.0000   -8.0000   -3.0000
         0    6.2105   17.5263    4.7895  -14.5789
         0         0   19.8136  -12.5254   -6.9831
         0         0         0   29.6553   -6.0860
         0   -0.0000   -0.0000         0    2.8550


P =

     0     0     0     1     0
     0     1     0     0     0
     0     0     0     0     1
     1     0     0     0     0
     0     0     1     0     0

>>Delta2 = norm(L*U-P*A)/norm(A)

Delta2 =

   9.8817e-17
```

The resulting values of `Delta1` and `Delta2` show that for this particular matrix **A** the Gaussian elimination with partial pivoting is $3\text{x}10^5$ times more accurate than Gaussian elimination without pivoting.