W.D. Henshaw **Math 6800: Solutions for Problem Set 7**

1. A matrix D is block tridiagonal if it is of the form

$$D = \begin{bmatrix} B_1 & C_1 & & & & \\ A_2 & B_2 & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & A_4 & B_4 & C_4 & \\ & & & \ddots & \ddots & \ddots \\ & & & & A_n & B_n \end{bmatrix}$$

where each A_i, B_i and C_i is a small matrix of size $p \times p$ (p is the *block size*). Derive a block LU decomposition (i.e. an LU decomposition that uses operations involving $p \times p$ matrices instead of scalars), assuming no pivoting. **next time: Write a psueo-code algorithm to compute the block LU factorization. What are the conditions you need for this algorithm to successfully complete?**

Solution:

Solution: In the first stage in the block-factorization we will multiply D by the block lower triangular matrix L_1 given by

$$L_1 = \begin{bmatrix} I_{p\times p} & & & & \\ -A_2B_1^{-1} & I_{p\times p} & & & \\ & & I_{p\times p} & & \\ & & & \ddots & \\ & & & & I_{p\times p} \end{bmatrix} \quad \Longrightarrow \quad L_1D = \begin{bmatrix} B_1 & C_1 & & & & \\ 0 & \tilde{B}_2 & C_2 & & & \\ & A_3 & B_3 & C_3 & & \\ & & A_4 & B_4 & C_4 & \\ & & & \ddots & \ddots & \ddots \\ & & & & A_n & B_n \end{bmatrix}$$

where $\tilde{B}_2 = B_2 - A_2B_1^{-1}C_1$. At stage 2 we multiply L_1D by the block lower triangular matrix L_2 given by

$$L_2 = \begin{bmatrix} I_{p\times p} & & & & & \\ & I_{p\times p} & & & & \\ & -A_3\tilde{B}_2^{-1} & I_{p\times p} & & & \\ & & & \ddots & & \\ & & & & I_{p\times p} \end{bmatrix} \quad \Longrightarrow \quad L_2L_1D = \begin{bmatrix} B_1 & C_1 & & & & \\ 0 & \tilde{B}_2 & C_2 & & & \\ & 0 & \tilde{B}_3 & C_3 & & \\ & & A_4 & B_4 & C_4 & \\ & & & \ddots & \ddots & \ddots \\ & & & & A_n & B_n \end{bmatrix}$$

where $\tilde{B}_3 = B_3 - A_3\tilde{B}_2^{-1}C_2$

This process can be repeated to give an upper block triangular matrix U,

$$L_{n-1}\cdots L_2L_1D = \begin{bmatrix} B_1 & C_1 & & & & \\ 0 & \tilde{B}_2 & C_2 & & & \\ & 0 & \tilde{B}_3 & C_3 & & \\ & & 0 & \tilde{B}_4 & C_4 & \\ & & & \ddots & \ddots & \ddots \\ & & & & 0 & \tilde{B}_n \end{bmatrix} = U,$$

where $\tilde{B}_i = B_i - A_i\tilde{B}_{i-1}^{-1}C_{i-1}$ Note that the inverse of any L_i is determined by negating the subdiagonal element. For example,

$$L_1^{-1} = \begin{bmatrix} I_{p\times p} & & & & \\ A_2B_1^{-1} & I_{p\times p} & & & \\ & & I_{p\times p} & & \\ & & & \ddots & \\ & & & & I_{p\times p} \end{bmatrix}.$$

Thus, as for standard GE, the matrix $L = L_1^{-1}L_2^{-1}\cdots L_{n-1}^{-1}$ will consist of negatives of the subdiagonal entries from each L_i. Therefore the block LU decomposition is of the form

$$D = \begin{bmatrix} I_{p\times p} & & & & \\ \tilde{L}_2 & I_{p\times p} & & & \\ & \tilde{L}_3 & I_{p\times p} & & \\ & & \ddots & \ddots & \\ & & & \tilde{L}_n & I_{p\times p} \end{bmatrix} \begin{bmatrix} \tilde{B}_1 & C_1 & & & & \\ 0 & \tilde{B}_2 & C_2 & & & \\ & 0 & \tilde{B}_3 & C_3 & & \\ & & 0 & \tilde{B}_4 & C_4 & \\ & & & \ddots & \ddots & \ddots \\ & & & & 0 & \tilde{B}_n \end{bmatrix} = LU,$$

where $\tilde{B}_1 = B_1$, $\tilde{L}_i = A_i\tilde{B}_{i-1}^{-1}$, $\tilde{B}_i = B_i - \tilde{L}_iC_{i-1}$, $i = 2, 3, \ldots, m$.

The LU decomposition exists provided $\tilde{B}_i$, $i = 1, 2, \ldots, n-1$ are nonsingular.

2. NLA 20.3 *Suppose an $m \times m$ matrix A ...*

Solution:

(a) By block matrix multiplication we find

$$\begin{bmatrix} I & 0 \\ -A_{21}A_{11}^{-1}I & \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ -A_{21}A_{11}^{-1}A_{11} + A_{21} & -A_{21}A_{11}^{-1}A_{12} + A_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} - A_{21}A_{11}^{-1}A_{12} \end{bmatrix}$$

as required.

(b) Using the first n steps of Gaussian elimnation to reduce A_{11} to upper triangular form and eliminate block A_{21} would give

$$L_{n-1}\cdots L_2L_1 \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} U_{11} & U_{12} \\ 0 & A_{22}^{(2)} \end{bmatrix}$$

or

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & I \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & A_{22}^{(2)} \end{bmatrix}$$

where L_{11} is an $n \times n$ unit lower triangular matrix and L_{21} is an $(m-n) \times n$ matrix. Multiplying the block matrices gives the four equations,

$$A_{11} = L_{11}U_{11}, \tag{1}$$
$$A_{12} = L_{11}U_{12}, \tag{2}$$
$$A_{21} = L_{21}U_{11}, \tag{3}$$
$$A_{22} = L_{21}U_{21} + A_{22}^{(2)}. \tag{4}$$

Thus

$$\begin{aligned} A_{22}^{(2)} &= A_{22} - L_{21}U_{12}, && \text{(from (4))}, \\ &= A_{22} - A_{21}U_{11}^{-1}U_{12}, && \text{(replace } L_{21} \text{ from } A_{21} = L_{21}U_{11}), \\ &= A_{22} - A_{21}U_{11}^{-1}L_{11}^{-1}A_{12}, && \text{(replace } U_{12} \text{ from } A_{12} = L_{11}U_{12}), \\ &= A_{22} - A_{21}A_{11}^{-1}A_{12}, && \text{(from } A_{11} = L_{11}U_{11}), \end{aligned}$$

and this completes the proof.

3. The matrix $A \in \mathbb{C}^{m\times m}$ is *diagonally dominant* if

$$|a_{ii}| > \sum_{j=1, j\neq i}^{m} |a_{ij}|, \quad i = 1, 2, \ldots, m.$$

(a) Prove that if A is diagonally dominant, then any principal submatrix of A is diagonally dominant.
(b) Prove that if A is diagonally dominant, then A is nonsingular.
(c) Prove that if A is diagonally dominant then it will have an LU decomposition (you may use the result of NLA 20.1).

Solution:

(a) Since

$$|a_{ii}| > \sum_{j=1, j\neq i}^{m} |a_{ij}|, \quad i = 1, 2, \ldots, m.$$

it follows immediately that for any principal $(k+1)\times(k+1)$ submatrix $A^{(k+1)}$ with upper left element a_{jj},

$$A^{(k+1)} = \begin{bmatrix} a_{jj} & \cdots & a_{j,j+k} \\ \vdots & & \vdots \\ a_{j+k,j} & \cdots & a_{j+k,j+k} \end{bmatrix}$$

then the magnitude of the diagonal entry is bigger than the sum of the magnitudes of all the off diagonal entries in the orginial matrix and thus larger than the sum of the magnitudes of the off diagonal entries in the principal sub-matrix,

$$|a_{j+p,j+p}| > \sum_{q=1, q\neq j+p}^{m} |a_{j+p,q}| \;\geq \sum_{q=j, q\neq j+p}^{j+k} |a_{j+p,q}|, \qquad p = 0, 1, \ldots, k.$$

(b) To show that A is nonsingular we will show that $Ax = 0$ implies $x = 0$. Therefore assume there is an $x \neq 0$ such that $Ax = 0$. Then

$$\sum_{j=1^m} a_{i,j}x_j = 0, \quad i = 1, 2, \ldots, m,$$

Let $|x_p| = \|x\|_\infty$, then by assumption $x \neq 0$ and thus $x_p \neq 0$. Now

$$|a_{pp}x_p| = |\sum_{j\neq p} a_{i,j}x_j| \leq \sum_{j\neq p} |a_{p,j}||x_j|$$

Since $|x_j|/|x_p| \leq 1$ it follows that after dividing both sides by $|x_p|$, that

$$|a_{pp}| \leq \sum_{j\neq p} |a_{p,j}| \frac{|x_j|}{|x_p|} \leq \sum_{j\neq p} |a_{p,j}|$$

But this contradicts the fact that A is diagonally dominant. Therefore $Ax = 0$ implies $x = 0$ and thus A is nonsingular.

(c) Since all principal submatrices of A are diagonally dominant, they are also nonsingular. Thus by the results of NLA exercise 20.1, A has an LU decomposition.

4. Write a Matlab code `[L,U,P]=lufactor(A)` that takes an $m \times m$ matrix A and computes the LU factorization, $PA = LU$, using partial pivoting. Write a second Matlab code `x = lusolve( b, L,U,P)` that solves the system $Ax = b$, for x, given b, using the output from `lufactor`. For this exercise you should only use elementary arithmetic, vector and matrix operations (e.g. no backslash operators to solve the triangular systems).

(a) Test your `lufactor` routine using the matrix

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

Output A, L, U and P. Check that $PA = LU$.

(b) Test your function `lusolve` by solving $Ax = b$ where A is from part (a) and $b = [7, 23, 69, 79]^T$. Output x and check that $Ax = b$.

Solution:

(a) The codes are given below. Here are the results for part (a)

```
A =

     2     1     1     0
     4     3     3     1
     8     7     9     5
     6     7     9     8


L =

    1.0000         0         0         0
    0.7500    1.0000         0         0
    0.5000   -0.2857    1.0000         0
    0.2500   -0.4286    0.3333    1.0000
```

```
U =

    8.0000    7.0000    9.0000    5.0000
         0    1.7500    2.2500    4.2500
         0         0   -0.8571   -0.2857
         0         0         0    0.6667


P =

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

 norm(P*A-L*U)=1.11e-16
```

(b) Here are the results for part (a)

```
x =

    1.0000
    2.0000
    3.0000
    4.0000

 norm(A*x-b)=1.78e-15
```

Listing 1: lufactor.m

```
1  function [L,U,P] = lufactor( A )
2  %
3  %   Compute an LU factorization with partial pivoting
4  %           P A = L U
5  %
6  %   A (input) : m x m matrix
7  %   L (output) : m x m lower triangular matrix with unit diagonal
8  %   U (output) : m x m upper diagonal matix.
9  %   P (output) : m x m permutation matrix
10 %
11
12 [m,n]=size(A);
13
14 U=A; L=eye(m,m); P=L;
15
16 for k=1:m-1
17   % find the pivot: (row i)
18   i=k;
19   for j=k+1:m
20     if abs(U(j,k))> abs(U(i,k))
21       i=j;
22     end;
23   end;
24   % fprintf('step k=%d, pivot=%d\n',k,i);
25   if i ~= k
26     % swap rows
```

```
    temp=U(k,k:m); U(k,k:m)=U(i,k:m); U(i,k:m)=temp;
    temp=L(k,1:k-1); L(k,1:k-1)=L(i,1:k-1); L(i,1:k-1)=temp;
    temp=P(k,:); P(k,:)=P(i,:); P(i,:)=temp;
  end;

  for j=k+1:m
    L(j,k)=U(j,k)/U(k,k);
    U(j,k:m)=U(j,k:m)-L(j,k)*U(k,k:m);
  end;

end;
```

Listing 2: lusolve.m

```
function x = lusolve( b, L,U,P )
%
%   Solve Ax = b using the results from LU factor
%            P A = L U
%
%   b (input) : m x 1 right hand side
%   L, U , P (input) : results from lufactor.
%   x (output) : m x 1 solution.
%

[m,n]=size(L);

% Solve LU = P*b;

% c=P*b;
c=zeros(m,1);
for i=1:m
  for j=1:m
    if P(i,j)~=0
      c(i)=b(j);
    end
  end
end;

% y = L\(P*b);
y=zeros(m,1);
for i=1:m
  y(i) = c(i);
  for j=1:i-1
    y(i) = y(i) - L(i,j)*y(j);
  end
end


% x= U\y;
x=zeros(m,1);
for i=m:-1:1
  x(i) = y(i);
  for j=i+1:m
    x(i) = x(i) - U(i,j)*x(j);
```

```
41    end
42    x(i) = x(i)/U(i,i);
43  end
```