LECTURE 10

LU Factorizations

Suppose that a matrix $\mathbf{A}$ has a factorization

$$\mathbf{A} = \mathbf{L}\mathbf{U}$$

where $\mathbf{L}$ and $\mathbf{U}$ are respectively lower triangular and upper triangular matrices. Then the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{10.1}$$

can be solved quite easily. For we can rewrite (10.1) as

$$\mathbf{L}(\mathbf{U}\mathbf{x}) = \mathbf{b}$$

But then $\mathbf{U}\mathbf{x}$ must be the solution of

$$\mathbf{L}\mathbf{z} = \mathbf{b}$$

(which we can solve explicitly for $\mathbf{z}$ since $\mathbf{L}$ is lower triangular), and so

$$\mathbf{U}\mathbf{x} = \mathbf{z}$$

(which we can solve explicitly for $\mathbf{x}$ since $\mathbf{U}$ is upper triangular).

We shall now develop an algorithm for finding a LU factorization of a given matrix $\mathbf{A}$. Not all matrices will have LU factorizations. However, it will turn out that the lack of an LU factorization can be attributed to the fact that the application of the algorithm given below is eventually nullified by an illegal operation; namely a division by 0. In other words, if a matrix has an LU factorization our algorithm will find it. If not, our algorithm will instead churn out a divide by zero error.

To develop this algorithm we start with the matrix multiplication formula

$$a_{ij} = \sum_{k=1}^{n} l_{ik} u_{kj} \tag{10.2}$$

Now the components of $\mathbf{L}$, being lower triangular, satisfy

$$l_{ik} = 0 \quad \text{if} \quad k > i$$

and the components of $\mathbf{U}$, being upper triangular, satisfy

$$u_{kj} = 0 \quad \text{if} \quad k > j$$

Therefore the sum (10.2) can be written

$$a_{ij} = \sum_{k=1}^{\min(i,j)} l_{ik} u_{kj} \tag{10.3}$$

Now before we actually try soving equations (10.3) for the components l_{ik} of $\mathbf{L}$ and the components u_{kj} of $\mathbf{U}$, let's first observe that the matrix $\mathbf{A}$ has n^2 components a_{ij}; so we have a total of n^2 equations in (10.3). Now the number of (possibly) non-zero components of the matrix $\mathbf{L}$ is determined by observing that it has the same number, n, of diagonal components as matrix $\mathbf{A}$, but only half the number of off diagonal components as $\mathbf{A}$. Therefore, the number of nontrivial entries of $\mathbf{L}$ is

$$n + \frac{1}{2}\left(n^2 - n\right) = \frac{n(n+1)}{2}$$

Similarly, the upper triangular matrix $\mathbf{U}$ has $\frac{1}{2}n(n+1)$ components. Thus the total number of unknowns in the system (10.3) is

$$\frac{n(n+1)}{2} + \frac{n(n+1)}{2} = n^2 + n$$

We thus have n more unknowns than we have equations. The general solution of (10.3) will thus contain n free parameters. We can remove these extra degrees of freedom, without destroying the possibility of finding a solution, by imposing n additional conditions on the components of $\mathbf{L}$ and/or $\mathbf{U}$. We shall do so in a manner that simplifies the subsequent solution of (10.3); namely we shall require

$$l_{ii} = 1 \quad , \quad i = 1, 2, \ldots, n \tag{10.4}$$

In other words, we force the lower diagonal matrix $\mathbf{L}$ to have only 1's along its diagonal. (The text refers to such a matrix as **unit lower triangular**.)

Let's now separate the n^2 equations (10.3) into three subsets corresponding to the cases when $i = j$, $i < j$, and $i > j$.

$$a_{ii} = \sum_{k=1}^{i} l_{ik} u_{ki} \quad , \quad i = 1, 2, \ldots, n \tag{10.5}$$

$$a_{ij} = \sum_{s=1}^{i} l_{ik} u_{kj} \quad , \quad i < j \tag{10.6}$$

$$a_{ij} = \sum_{k=1}^{j} l_{ik} u_{kj} \quad , \quad j < i \tag{10.7}$$

Setting $i = 1$ in (10.4) and (10.5) we have

$$a_{11} = l_{11} u_{11} = u_{11} \quad \Rightarrow \quad u_{11} = a_{11} \tag{10.8}$$

Setting $i = 1$ in (10.6) yields

$$a_{1j} = \sum_{k=1}^{1} l_{ik} u_{kj} = l_{11} u_{1j} = u_{1j} \quad \Rightarrow \quad u_{1j} = a_{1j} \quad , \quad j = 2, 3, \ldots, n \tag{10.9}$$

Note that equations (10.8) and (10.9) completely determine the first row of $\mathbf{U}$.

Setting $j = 1$ in (10.7) we obtain

$$a_{i1} = \sum_{k=1}^{1} l_{ik} u_{k1} = l_{i1} u_{11} \quad \Rightarrow \quad l_{i1} = \frac{1}{u_{11}} a_{i1} = \frac{a_{i_1}}{a_{11}} \quad , \quad i = 2, 3, \ldots, n \tag{10.10}$$

Since $l_{11} = 1$ by hypothesis, equations (10.10) fix all the elements of the first column of the matrix $\mathbf{L}$.

Let's now set $i = 2$ in (10.5). This yields

$$a_{22} = \sum_{k=1}^{2} l_{2k} u_{k2} = l_{21} u_{12} + l_{11} u_{22} = l_{21} u_{12} + u_{22} \quad \Rightarrow \quad u_{22} = a_{22} - l_{21} u_{12} \tag{10.11}$$

Since l_{21} (an element of the first column of $\mathbf{L}$) and u_{12} (an element of the first row of $\mathbf{U}$) have already been determined (10.11) determines u_{22}. Setting $i = 2$ in (10.6) yields

$$(10.12) \qquad a_{2j} = \sum_{k=1}^{2} l_{2k} u_{kj} = l_{21} u_{1j} + l_{22} u_{2j} = l_{21} u_{1j} + u_{2j} \quad \Rightarrow \quad u_{2j} = a_{2j} - l_{21} u_{1j} \quad , \quad j = 3, 4, \ldots, n$$

Since $u_{21} \equiv 0$, equations (10.11) and (10.12) completely fix the second row of $\mathbf{U}$.

Now set $j = 2$ in (10.7). This yields

$$(10.13) \qquad a_{i2} = \sum_{k=1}^{2} l_{ik} u_{k2} = l_{i1} u_{12} + l_{i2} u_{22} \quad \Rightarrow \quad l_{i2} = \frac{1}{u_{22}} (a_{i2} - l_{i1} u_{12}) \quad , \quad i = 3, 4, \ldots, n$$

Since $l_{12} \equiv 0$, $l_{22} \equiv 1$, and because u_{22}, l_{i1}, and u_{12} have all been previously determined, these relations suffice to fix the second column of $\mathbf{L}$.

Let me now review the steps taken so far so that we can bring to life the general algorithm.

1. We set all the diagonal elements of $\mathbf{L}$ equal to 1.
2. We determined u_{11} from equation (10.5) with $i = 1$.
3. We determined the first row of $\mathbf{U}$ from the preceding results and the equations (10.6) with $i = 1$ and $j = 2, 3, \ldots, n$.
4. We determined the first column of $\mathbf{L}$ from the preceding results and the equations (10.7) with $j = 1$. and $i = 2, 3, \ldots, n$.
5. We determined u_{22} from the preceding results and equation (10.5) with $i = 2$.
6. We determined the second row of $\mathbf{U}$ from the preceding results and the equations (10.6) with $i = 2$. and $j = 3, 4, \ldots, n$.
7. We determined the second column of $\mathbf{L}$ from the preceding results and the equations (10.7) with $j = 2$ and $i = 3, 4, \ldots, n$.

The general algorithm can now be stated.

For each k from 1 to n

1. Set $l_{ik} = 0$ for $i = 1, 2, \ldots, k-1$ (so that $\mathbf{L}$ is lower triangular).
2. Set $u_{ki} = 0$ for $i = 1, 2, \ldots, k-1$ (so that $\mathbf{U}$ is upper triangular).
3. Set $l_{kk} = 1$.
4. Determine u_{kk} from the equation (10.5)
$$a_{kk} = \sum_{s=1}^{k} l_{ks} u_{sk} \quad \Rightarrow \quad u_{kk} = \frac{1}{l_{kk}} \left(a_{kk} - \sum_{s=1}^{k-1} l_{ks} u_{sk} \right)$$
Note that the expression on the far right involves only the first $k-1$ columns of $\mathbf{L}$ and the first $k-1$ rows of $\mathbf{U}$.
5. Determine the remaining elements of the k^{th} row of $\mathbf{U}$ from the equations (10.6) with $i = k$ and $j = k+1, k+2, \ldots, n$
$$a_{kj} = \sum_{s=1}^{k} l_{ks} u_{sj} \quad \Rightarrow \quad u_{kj} = \frac{1}{l_{kk}} \left(a_{kj} - \sum_{s=1}^{k-1} l_{ks} u_{sj} \right)$$
6. Determine the remaining elements of the k^{th} column of $\mathbf{L}$ from the equations (10.7) with $j = k$ and $i = k+1, k+2, \ldots, n$
$$a_{ik} = \sum_{s=1}^{k} l_{is} u_{sk} \quad \Rightarrow \quad l_{ik} = \frac{1}{u_{kk}} \left(a_{ik} - \sum_{s=1}^{k-1} l_{is} u_{sk} \right)$$

EXAMPLE 10.1. Consider the following 3×3 matrix:

$$\mathbf{A} = \begin{pmatrix} 5 & 6 & 7 \\ 10 & 20 & 23 \\ 15 & 50 & 67 \end{pmatrix}$$

Write a Maple program that carries out an LU factorization of $\mathbf{A}$.

The following code works.

```
n := 3;  # all matrices are nxn=3x3
A := array(1..n,1..n);
L := array(1..n,1..n);
U := array(1..n,1..n);
A := [[5,6,7],[10,20,23],[15,50,67]];

for k from 1 to n do   # calculate kth column of L and kth row of U
    for s from 1 to k-1 do
        L[s,k] := 0;  # so that L is lower triangular
        U[k,s] := 0;  # so that U is upper triangular
        od;
    L[k,k] := 1;       # by convention
    k1 := k-1;
    # calculate the kth element of kth row of U
    U[k,k] := A[k,k] -  sum(L[k,j0]*U[j0,k],j0=1..k1);
    for t from k+1 to n do
        # calculate remaining elements in kth column of L
        L[t,k] := (A[t,k] - sum(L[t,j1]*U[j1,k],j1=1..k1))/U[k,k];
        # calculate remaining elements in kth row of U
        U[k,t] := A[k,t] -  sum(L[k,j2]*U[j2,t],j2=1..k1);
        od;
     od;
print(L);
print(U);
```