# LU and Cholesky decompositions. J. Demmel, Chapter 2.7

Solving $Ax = b$ using Gaussian elimination.

1. Factorize $A$ into $A = PLU$

   Permutation   Unit lower triangular   Non-singular upper triangular

Solving $Ax = b$ using Gaussian elimination.

1. Factorize $A$ into $A = PLU$
   Permutation Unit lower triangular Non-singular upper triangular
2. Solve $PLUx = b$ (for $LUx$) :

$$LUx = P^{-1}b$$

Solving $Ax = b$ using Gaussian elimination.

1. Factorize $A$ into $A = PLU$
   Permutation    Unit lower triangular    Non-singular upper triangular
2. Solve $PLUx = b$ (for $LUx$) :
$$LUx = P^{-1}b$$
3. Solve $LUx = P^{-1}b$ (for $Ux$) by forward substitution:
$$Ux = L^{-1}(P^{-1}b).$$

Solving $Ax = b$ using Gaussian elimination.

1. Factorize $A$ into $A = PLU$
   Permutation  Unit lower triangular  Non-singular upper triangular
2. Solve $PLUx = b$ (for $LUx$) :

$$LUx = P^{-1}b$$

3. Solve $LUx = P^{-1}b$ (for $Ux$) by forward substitution:

$$Ux = L^{-1}(P^{-1}b).$$

4. Solve $Ux = L^{-1}(P^{-1}b)$ by backward substitution:

$$x = U^{-1}(L^{-1}P^{-1}b).$$

# Example of LU factorization

We factorize the following 2-by-2 matrix:

$$\begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix}. \tag{1}$$

One way to find the LU decomposition of this simple matrix would be to simply solve the linear equations by inspection. Expanding the matrix multiplication gives

$$\begin{aligned} l_{11} \cdot u_{11} + 0 \cdot 0 &= 4, \\ l_{11} \cdot u_{12} + 0 \cdot u_{22} &= 3, \\ l_{21} \cdot u_{11} + l_{22} \cdot 0 &= 6, \\ l_{21} \cdot u_{12} + l_{22} \cdot u_{22} &= 3. \end{aligned} \tag{2}$$

This system of equations is underdetermined. In this case any two non-zero elements of $L$ and $U$ matrices are parameters of the solution and can be set arbitrarily to any non-zero value. Therefore, to find the unique LU decomposition, it is necessary to put some restriction on L and U matrices. For example, we can conveniently require the lower triangular matrix $L$ to be a unit triangular matrix (i.e. set all the entries of its main diagonal to ones).

Then the system of equations has the following solution:

$$\begin{aligned} l_{21} &= 1.5, \\ u_{11} &= 4, \\ u_{12} &= 3, \\ u_{22} &= -1.5. \end{aligned} \tag{3}$$

Substituting these values into the LU decomposition above yields

$$\begin{bmatrix} 4 & 3 \\ 6 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1.5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 3 \\ 0 & -1.5 \end{bmatrix}. \tag{4}$$

**Definition**

The leading j-by-j principal submatrix of $A$ is $A(1:j, 1:j)$.

**Theorem 2.4.**

*The following two statements are equivalent:*
*1. There exists a unique unit lower triangular L and non-singular upper triangular $U$ such that $A = LU$.*
*2. All leading principal submatrices of $A$ are non-singular.*

## Algorithm 2.2

*LU factorization with pivoting:*

*for* $i = 1$ *to n-1*

*apply permutations so* $a_{ii} \neq 0$*(permute L, U)*

*/* for example for GEPP, swap rows j and i of A and of L where* $|a_{ji}|$ *is the largest entry in* $|A(i : n, i|$*;*
*for GECP, swap rows j and i of A and of L, and columns k and i of A and U, where* $|a_{jk}|$ *is the largest entry in* $|A(i : n, i : n)|$**/*

*/* compute column i of L */*

*for j=i+1 to n*

$l_{ji} = \frac{a_{ji}}{a_{ii}}$

*end for*

*/* compute row j of U */*

*for j=i to n*

$u_{ij} = a_{ij}$ *end for*

**Algorithm 2.2**

```
    /* update A_22 */
    for j=i+1 to n
        for k=i+1 to n
            a_jk = a_jk - l_ji * u_ik
        end for
    end for
end for
```

**Algorithm 2.3**

*LU factorization with pivoting, overwriting L and U on A:*

```
for i=1 to n-1
    apply permutations (see Algorithm 2.2.)
    for j=i+1 to n
        a_ji = a_ji / a_ii
    end for
    for j=i+1 to n
        for k=i+1 to n
            a_jk = a_jk - a_ji * a_ik
        end for
    end for
end for
```

$$a_{ji} = \frac{a_{ji}}{a_{ii}}$$

$$a_{jk} = a_{jk} - a_{ji} * a_{ik}$$

**Algorithm 2.4**

*LU factorization with pivoting, overwriting L and U on A, using Matlab notations:*

```
for i=1 to n-1
    apply permutations(see algorithm 2.2. )
    A(i+1:n,i)=A(i+i:n,i)/A(i,i)
    A(i+1:n,i,i+1:n)=
        A(i+1:n,i+1:n)-A(i+1:n,i)*A(i,i+1:n)
end for
```

# 2.7.1. Real Symmetric Positive Definite Matrices

Recall that a real matrix $A$ is s.p.d. if and only if $A = A^T$ and $x^T Ax > 0$ for all $x \neq 0$. In this section we will show how to solve $Ax = b$ in half the time and half the space of Gaussian elimination when $A$ is s.p.d.

PROPOSITION 2.2.

1. *If $X$ is nonsingular, then $A$ is s.p.d. if and only if $X^T AX$ is s.p.d.*
2. *If $A$ is s.p.d. and $H$ is any principal submatrix of $A$($H = A(j : k, j : k)$ for some $j \leq k$), then $H$ is s.p.d.*
3. *$A$ is s.p.d. if and only if $A = A^T$ and all its eigenvalues are positive.*
4. *If $A$ is s.p.d., then all $a_{ii} > 0$, and $\max_{ij} |a_{ij}| = \max_i a_{ii} > 0$.*
5. *$A$ is s.p.d. if and only if there is a unique lower triangular nonsingular matrix $L$, with positive diagonal entries, such that $A = LL^T$. $A = LL^T$ is called the Cholesky factorization of $A$, and $L$ is called the Cholesky factor of $A$.*

ALGORITHM 2.11. *Cholesky algorithm:*

*for* $j = 1$ *to* $n$

$l_{jj} = (a_{jj} - \sum_{k=1}^{j-1} l_{jk}^2)^{1/2}$

*for* $i = j + 1$ *to* $n$

$l_{ij} = (a_{ij} - \sum_{k=1}^{j-1} l_{ik} l_{jk}) / l_{jj}$

*end for*

*end for*

If $A$ is not positive definite, then (in exact arithmetic) this algorithm will fail by attempting to compute the square root of a negative number or by dividing by zero; this is the cheapest way to test if a symmetric matrix is positive definite.