

Linear equations: LU decomposition

Consider solving a linear system of equations, say with 3 variables

$$A \cdot x = b \rightarrow A = L \cdot U$$

Matrix **A** is factorized or **decomposed** into a product of *lower triangular* **L** and *upper triangular* **U** matrices,

$$L = \begin{pmatrix} \ell_{00} & 0 & 0 \\ \ell_{10} & \ell_{11} & 0 \\ \ell_{20} & \ell_{21} & \ell_{22} \end{pmatrix} \quad U = \begin{pmatrix} u_{00} & u_{01} & u_{02} \\ 0 & u_{11} & u_{12} \\ 0 & 0 & u_{22} \end{pmatrix}$$

Looks formidable but very useful and not nearly as hard.
Multiplying

$$\begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \ell_{00}u_{00} & \ell_{00}u_{01} & \ell_{00}u_{02} \\ \ell_{10}u_{00} & \ell_{10}u_{01} + \ell_{11}u_{11} & \ell_{10}u_{02} + \ell_{11}u_{12} \\ \ell_{20}u_{00} & \ell_{20}u_{01} + \ell_{21}u_{11} & \ell_{20}u_{02} + \ell_{21}u_{12} + \ell_{22}u_{22} \end{pmatrix}$$

But without proper ordering, the factorization may fail!

LU Decomposition

The matrix multiplication $\mathbf{L} \cdot \mathbf{U}$ yields

$$\begin{array}{lll} \ell_{00} u_{00} = a_{00} & \ell_{00} u_{01} = a_{01} & \ell_{00} u_{02} = a_{02} \\ \ell_{10} u_{00} = a_{10} & \ell_{10} u_{01} + \ell_{11} u_{11} = a_{11} & \ell_{10} u_{02} + \ell_{11} u_{12} = a_{12} \\ \ell_{20} u_{00} = a_{20} & \ell_{20} u_{01} + \ell_{21} u_{11} = a_{21} & \ell_{20} u_{02} + \ell_{21} u_{12} + \ell_{22} u_{22} = a_{22} \end{array}$$

A catch : $3 \times 3 = 9$ equations but $3 \times (3 + 1)$ variables!!

Trick : Either all three $\ell_{ij} = 1$, called *Doolittle* or all three $u_{ij} = 1$, called *Crout* decomposition.

Any of the decomposition of \mathbf{L} and \mathbf{U} can proceed iteratively.

Doolittle : $\ell_{11} = \ell_{22} = \ell_{33} = 1$ and *Crout* : $u_{11} = u_{22} = u_{33} = 1$

This straight away implies

$$\begin{array}{ll} \text{Doolittle :} & u_{00} = a_{00}, u_{01} = a_{01}, u_{02} = a_{02} \\ \text{Crout :} & \ell_{00} = a_{11}, \ell_{10} = a_{10}, \ell_{20} = a_{20} \end{array}$$

The rest of the ℓ_{ij} or u_{ij} can be solved from the remaining equations to achieve LU decomposition.

Doolittle LU

Take Doolittle LU factorization,

1. Set $\ell_{ii} = 1 \quad \forall i = 0, 1, \dots, N-1$ implying

$$u_{00} = a_{00}, \quad u_{01} = a_{01} \text{ and } u_{02} = a_{02} \Rightarrow u_{0j} = a_{0j}, \quad (j = 0, 1, \dots, N-1)$$

2. Do the calculation in the order they appear

$$u_{10} = 0,$$

$$\ell_{10} = (a_{10})/u_{00}$$

$$u_{20} = 0,$$

$$\ell_{20} = (a_{20})/u_{00}$$

$$u_{11} = a_{11} - \ell_{10}u_{01},$$

$$\ell_{11} = 1$$

$$u_{21} = 0,$$

$$\ell_{21} = (a_{21} - \ell_{20}u_{01})/u_{11}$$

$$u_{12} = a_{12} - \ell_{10}u_{02},$$

$$\ell_{12} = 0$$

$$u_{22} = a_{22} - \ell_{20}u_{02} - \ell_{21}u_{12},$$

$$\ell_{22} = 0$$

3. Generic form for each $j = 0, 2, \dots, N-1$, in the order they appear

$$u_{ij} = a_{ij} - \sum_{k=0}^{i-1} \ell_{ik} u_{kj} \quad \text{for } i = 2, \dots, j$$

$$\ell_{ij} = \left(a_{ij} - \sum_{k=0}^{j-1} \ell_{ik} u_{kj} \right) / u_{jj} \quad \text{for } i = j+1, j+2, \dots, N-1$$

LU storage

Important : Every a_{ij} is used only once and never again.

$\Rightarrow u_{ij}, \ell_{ij}$ can be stored in the same location / memory of a_{ij} .
Hence memory requirement is half.

Storing **Doolittle LU** decomposed matrix

$$LU = \begin{pmatrix} u_{00} & u_{01} & u_{02} & u_{03} \\ \ell_{10} & u_{11} & u_{12} & u_{13} \\ \ell_{20} & \ell_{21} & u_{22} & u_{23} \\ \ell_{30} & \ell_{31} & \ell_{32} & u_{33} \end{pmatrix} \rightarrow A$$

No need to store $\ell_{ii} = 1$, modify loop over indices accordingly.

Otherwise, numerical cost of Gauss-Jordan and LU are same,
 $\mathcal{O}(N^3)$.

But, can LU be always done?

Can we always LU decompose?

- If $a_{11} = 0$, then either L or U is singular \rightarrow impossible if A is not.

Solution : Row pivot

- Guaranteed if all *leading submatrices* have nonzero determinant

$$A = \begin{pmatrix} 1 & 0 & 2 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \Rightarrow A_1 = 1, A_2 = \begin{pmatrix} 1 & 0 \\ -2 & 1 \end{pmatrix} \text{ and } A_3 = \begin{pmatrix} 1 & 0 & 2 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$\det[A_1] = 1, \det[A_2] = 1 \text{ and } \det[A_3] = -3$$

- Additional pivoting if determinant of any leading submatrix is zero but the matrix itself is invertible.

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 0 & 2 \\ -1 & 1 & 2 \end{pmatrix} \Rightarrow A_1 = 1, A_2 = \begin{pmatrix} 1 & 0 \\ 3 & 0 \end{pmatrix} \text{ and } A_3 = \begin{pmatrix} 1 & 0 & 2 \\ 3 & 0 & 2 \\ -1 & 1 & 2 \end{pmatrix}$$

$$\det[A_1] = 1, \det[A_2] = 0 \text{ and } \det[A_3] = 4$$

- Otherwise, no solution exists and your are doomed!

U in Gauss-Jordan

Recall the **U** matrix needed for determinant calculation in Gauss-Jordan elimination

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \Rightarrow \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \bar{b}_1 \\ \bar{b}_2 \\ \bar{b}_3 \end{pmatrix}$$

Solutions x_i starts from last row i.e. by *backward substitution*

$$u_{33}x_3 = \bar{b}_3 \quad x_3 = \frac{\bar{b}_3}{u_{33}}$$

$$u_{22}x_2 + u_{23}x_3 = \bar{b}_2 \quad x_2 = \frac{\bar{b}_2 - u_{23}x_3}{u_{22}}$$

$$u_{11}x_1 + u_{12}x_2 + u_{13}x_3 = \bar{b}_1 \quad x_1 = \frac{\bar{b}_1 - u_{12}x_2 - u_{13}x_3}{u_{11}}$$

A generic solution for $N \times N$ matrix by backward substitution is

$$x_i = \frac{1}{u_{ii}} \left(\bar{b}_i - \sum_{j=i+1}^N u_{ij}x_j \right), \text{ where } x_N = \frac{\bar{b}_N}{u_{NN}} \text{ and } i = N-1, N-2, \dots, 1$$

LU forward-backward

To solve linear system of equations using LU decomposition it is advisable to begin with **partial pivoting**.

★ In the next step, consider the following split up

$$A \cdot x = b \Rightarrow L \cdot (U \cdot x) = b \quad \Bigg| \quad U \cdot x = y \Rightarrow L \cdot y = b$$

★ First solve for y from $L \cdot y = b$ using *forward substitution*, then use it to solve for x by *backward substitution* from $U \cdot x = y$

Forward substitution :

$$\begin{pmatrix} 1 & 0 & 0 \\ \ell_{01} & 1 & 0 \\ \ell_{20} & \ell_{21} & 1 \end{pmatrix} \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix}$$

Starting from the **first row** i.e. moving **forward** we solve for y_i

$$\begin{array}{ll} y_0 = b_0 & y_0 = b_0 \\ \ell_{10}y_0 + y_1 = b_1 & y_1 = b_1 - \ell_{10}y_0 \\ \ell_{20}y_0 + \ell_{21}y_1 + y_2 = b_2 & y_2 = b_2 - \ell_{20}y_0 - \ell_{21}y_1 \end{array}$$

Backward substitution :

$$\begin{pmatrix} u_{00} & u_{01} & u_{02} \\ 0 & u_{11} & u_{12} \\ 0 & 0 & u_{22} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix}$$

Starting from the **last row** i.e. moving **backward** to solve for x_i

$$u_{33}x_3 = y_3 \quad x_3 = y_3 / u_{33}$$

$$u_{22}x_2 + u_{23}x_3 = y_2 \quad x_2 = (y_2 - u_{23}x_3) / u_{22}$$

$$u_{11}x_1 + u_{12}x_2 + u_{13}x_3 = y_1 \quad x_1 = (y_1 - u_{12}x_2 - u_{13}x_3) / u_{11}$$

In generic form, the solutions for y_i and subsequently x_i are

$$y_i = b_i - \sum_{j=0}^{i-1} \ell_{ij}y_j, \quad \text{where } y_0 = b_0 \text{ and } i = 2, 3, \dots, N$$

$$x_i = \frac{1}{u_{ii}} \left(y_i - \sum_{j=i+1}^N u_{ij}x_j \right), \quad \text{where } x_N = \frac{y_N}{u_{NN}} \text{ and } i = N-1, N-2, \dots, 1$$

★ Get determinant of **A** for free

$$\det A = \det LU = \det L \times \det U = (-1)^n \prod_i u_{ii}$$

★ For inverse, iterate through each column of the identity matrix.

Cholesky decomposition

Properties and symmetries of A are often used to simplify the process of solving linear equations.

Cholesky decomposition : factorization of **Hermitian, positive definite** matrix (which often is the case in physics e.g. covariance matrix) into a product of L and L^T .

For a 3×3 system,

$$A = LL^\dagger \xrightarrow{\text{real}} LL^T \Rightarrow \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} \ell_{00} & 0 & 0 \\ \ell_{10} & \ell_{11} & 0 \\ \ell_{20} & \ell_{21} & \ell_{22} \end{pmatrix} \begin{pmatrix} \ell_{00} & \ell_{01} & \ell_{02} \\ 0 & \ell_{11} & \ell_{12} \\ 0 & 0 & \ell_{22} \end{pmatrix}$$

where $\ell_{ij} = \ell_{ji}$. When A is real and positive definite,

$$\ell_{ii} = \pm \sqrt{a_{ii} - \sum_{j=0}^{i-1} \ell_{ij}^2} \quad \text{and} \quad \ell_{ij} = \frac{1}{\ell_{ii}} \left(a_{ij} - \sum_{k=0}^{i-1} \ell_{ik} \ell_{kj} \right) \quad \text{for } i < j$$

Signs before square roots are inconsequential. **DIY** the decomposition for complex matrix.

Cholesky

Cholesky is about **TWICE** as efficient as the LU decomposition for solving system of linear equations.

An example of Cholesky decomposition of a real, symmetric matrix is (taken from Wikipedia),

$$\begin{pmatrix} 4 & 12 & -16 \\ 12 & 37 & -43 \\ -16 & -43 & 98 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 6 & 1 & 0 \\ -8 & 5 & 3 \end{pmatrix} \begin{pmatrix} 2 & 6 & -8 \\ 0 & 1 & 5 \\ 0 & 0 & 3 \end{pmatrix}$$

Apart from being used for numerical solution of linear equations, Cholesky decomposition is also used in non-linear optimization for multiple variable, monte carlo simulation for decomposing covariance matrix, inversion of Hermitian matrices etc.