LU Decomposition Method

Lecture 17 | Direct methods for linear systems

Core idea	Factor a square matrix A into a lower and an upper triangular matrix.
Solve efficiently	Replace Ax = b by two triangular systems: Ly = b and Ux = y.
Learning focus	Doolittle factorization, worked examples, pivoting and implementation.

Learning outcomes

By the end of this lecture, students should be able to:

Factor	Compute L and U using elimination multipliers and Doolittle formulas.
Solve	Apply forward and backward substitution to obtain x.
Evaluate	Verify the factorization and residual; explain cost and pivoting.

Why factor a matrix?

One factorization can serve many right-hand sides.

Original system	$Ax = b$, where A is $n \times n$ and x and b are column vectors.
Factorization	$A = LU \Rightarrow L(Ux) = b$. Introduce $y = Ux$.
Workflow	$A \to L, U$ | $b \to$ solve $Ly = b \to$ solve $Ux = y \to x$

Structure of L and U

Doolittle convention: the diagonal entries of L are all 1.

Lower triangular L

$$\begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix}$$

Upper triangular U

$$\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Interpretation

L stores elimination multipliers; U is the resulting upper triangular matrix.

Doolittle formulas

Compute the kth row of U, then the kth column of L.

Upper row	$u(k,j) = a(k,j) - \Sigma\, l(k,s)u(s,j),\ \ s = 1,\ldots,k-1;\ \ j \geq k.$
Lower column	$l(i,k) = [a(i,k) - \Sigma\, l(i,s)u(s,k)] \,/\, u(k,k),\ \ i > k.$
Pivot requirement	Set $l(k,k) = 1$. A zero pivot prevents this unpivoted procedure.

Example 1 — factor a 2 × 2 matrix

Solve 2x + y = 5 and 4x + 3y = 11.

Input — $A = [[2, 1], [4, 3]]$; $b = [5, 11]^T$.

Eliminate — $m_{21} = 4/2 = 2$; $R_2 \leftarrow R_2 - 2R_1$. Second row becomes [0, 1].

Factors — $L = [[1, 0], [2, 1]]$; $U = [[2, 1], [0, 1]]$.

Example 1 — solve and verify

Use the factors from the previous slide.

Forward solve	$Ly = b$: $y_1 = 5$; $2y_1 + y_2 = 11 \Rightarrow y_2 = 1$.
Backward solve	$Ux = y$: $x_2 = 1$; $2x_1 + x_2 = 5 \Rightarrow x_1 = 2$.
Verification	$A[2, 1]^T = [5, 11]^T$. The residual $b - Ax$ is zero.

Example 2 — a 3 × 3 system

Find the solution using Doolittle factorization.

Coefficient matrix	$A = [[2, 1, 1], [4, 3, 3], [8, 7, 9]]$
Right-hand side	$b = [7, 17, 43]^T$
Goal	Compute $A = LU$, solve $Ly = b$, then solve $Ux = y$.

Example 2 — first pivot

Pivot $a_{11} = 2$. Eliminate entries beneath it.

Row 2	$m_{21} = 4/2 = 2$; $R_2 - 2R_1 = [0, 1, 1]$.
Row 3	$m_{31} = 8/2 = 4$; $R_3 - 4R_1 = [0, 3, 5]$.
Intermediate U	$[[2, 1, 1], [0, 1, 1], [0, 3, 5]]$

Example 2 — second pivot

Pivot $u_{22} = 1$. Eliminate the remaining lower entry.

Multiplier	$m_{32} = 3/1 = 3$; $R_3 \leftarrow R_3 - 3R_2 = [0, 0, 2]$.
Final U	[[2, 1, 1], [0, 1, 1], [0, 0, 2]]
Final L	[[1, 0, 0], [2, 1, 0], [4, 3, 1]]

Example 2 — forward substitution

Solve Ly = b, starting at the top row.

First row	$y_1 = 7$
Second row	$2(7) + y_2 = 17 \Rightarrow y_2 = 3$
Third row	$4(7) + 3(3) + y_3 = 43 \Rightarrow y_3 = 6$

Example 2 — backward substitution

Solve $Ux = [7, 3, 6]^T$ from the bottom row.

Third row	$2x_3 = 6 \Rightarrow x_3 = 3$
Second row	$x_2 + x_3 = 3 \Rightarrow x_2 = 0$
First row	$2x_1 + x_2 + x_3 = 7 \Rightarrow x_1 = 2$

Verification and residual

Check both the factorization and the computed solution.

Factor check	LU = [[2, 1, 1], [4, 3, 3], [8, 7, 9]] = A.
Solution check	$Ax = [2(2)+3, 4(2)+9, 8(2)+27]^T = [7, 17, 43]^T$.
Residual	$r = b - Ax = [0, 0, 0]^T$. A small residual alone does not guarantee small forward error.

Pivoting and existence

A nonsingular matrix may still require row exchanges.

Counterexample	A = [[0, 1], [1, 1]] is nonsingular but its first pivot is zero.
Partial pivoting	Choose the largest absolute entry in the active column; swap rows.
Permuted factorization	PA = LU. Solve Ly = Pb and Ux = y. Swap previously stored L entries as needed.

Cost and practical applications

LU is especially useful for repeated solves.

Factorization	Dense LU costs about $2n^3/3$ floating-point operations.
Triangular solves	Forward and backward substitution together cost about $2n^2$ operations per right-hand side.
Applications	Circuit models, structural analysis, repeated simulations and implicit time stepping.

Python implementation

Use a pivoted LU routine for general dense matrices.

Factor once

```
from scipy.linalg import lu_factor, lu_solve
A = [[2,1,1],[4,3,3],[8,7,9]]
```

Solve

```
fac = lu_factor(A)
x = lu_solve(fac, [7,17,43])
```

Result

x ≈ [2., 0., 3.]; reuse fac for a new right-hand side.

Practice questions

Try the exercises before viewing the solutions.

Q1 Factor A = [[3, 2], [6, 5]] using Doolittle and solve b = [7, 16]^{T}.

Q2 Why is unpivoted LU impossible for [[0, 2], [1, 3]]?

Q3 Explain why LU is preferable to repeating elimination for ten right-hand sides.

Solutions and recap

LU factorization → forward solve → backward solve.

Q1	L = [[1,0],[2,1]], U = [[3,2],[0,1]]; y = $[7,2]^T$; x = $[1,2]^T$.
Q2 / Q3	First pivot is zero; swap rows. Reusing the factors saves repeated cubic-cost factorization.
Next lecture	NC_18 — Jacobi Iterative Method.