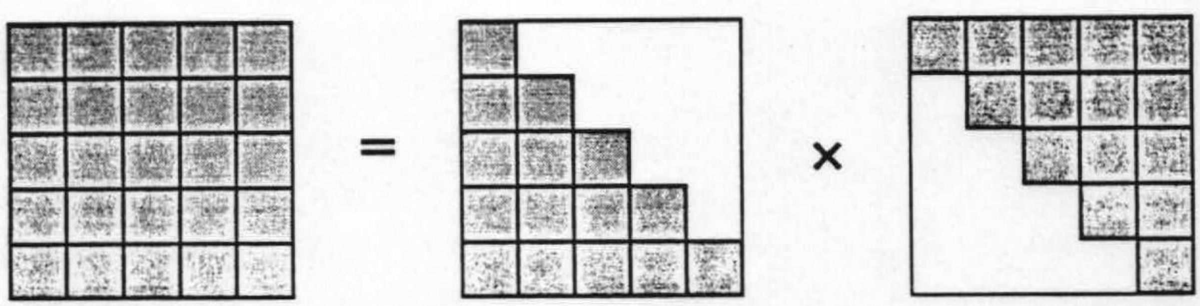
LU Decomposition

What is L U decomposition?

L U decomposition is the factorization of a square matrix into the product of a **lower-triangular** matrix and an **upper-triangular** matrix.

$$A = L \times U$$



Where does A come from?

A is the coefficient matrix from a system of linear equations, which may arise in many kinds of scientific application:

- physical systems
- biological systems
- economic models

$$\left\{\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + \ldots + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + \ldots + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + \ldots + & a_{nn}x_n & = & b_n \end{array}\right\}$$

$$\text{or} \qquad A\,x = b$$

$$\text{where} \qquad A = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

How is the decomposition useful?

$$\begin{array}{c}\text{Decomposing } A = LU \\ \downarrow \\ \text{Solving the system } A\,x = b\end{array}$$

Two-step procedure:

1. Solve $L\,c = b$:

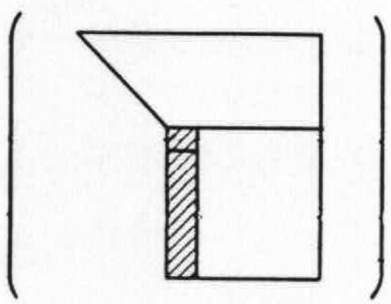
$$\begin{bmatrix} \ell_{11} & & \\ \vdots & \ddots & \\ \ell_{n1} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

2. Solve $U\,x = c$:

$$\begin{bmatrix} u_{11} & \cdots & u_{1n} \\ & \ddots & \vdots \\ & & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$$

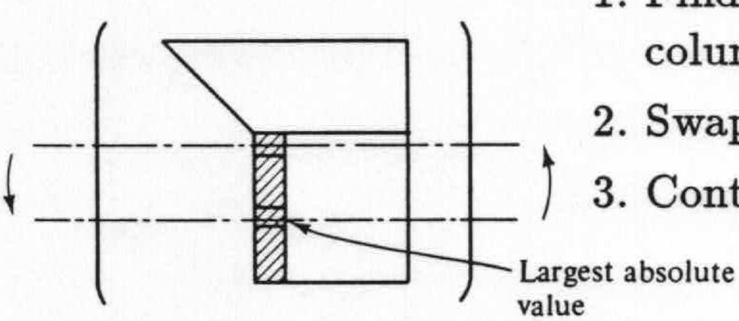
How to perform L U decomposition: Use Gaussian Elimination

Without pivoting:



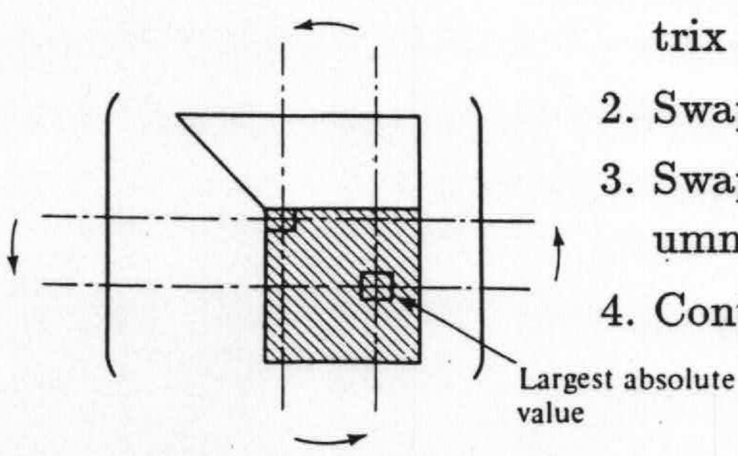
1. Divide column by pivot value to get row multipliers.
2. Subtract row multiplier × pivot's row from each lower row.

Partial pivoting:



1. Find largest magnitude in pivot's column at or below pivot.
2. Swap its row with pivot's row.
3. Continue as without pivoting.

Full pivoting:



1. Find largest magnitude in submatrix at or "southeast" of pivot.
2. Swap its row with pivot's row.
3. Swap its column with pivot's column.
4. Continue as without pivoting.

Gaussian Elimination Without Pivoting: An Example

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\downarrow$$

$$\begin{array}{c} \\ \frac{c}{a} \end{array} \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\downarrow$$

$$\begin{bmatrix} a & b \\ (c - \frac{c}{a} \times a) & (d - \frac{c}{a} \times b) \end{bmatrix}$$

$$\swarrow \qquad \qquad \searrow$$

$$U = \begin{bmatrix} a & b \\ 0 & (d - \frac{c}{a} \times b) \end{bmatrix} \qquad \qquad L = \begin{bmatrix} 1 & 0 \\ \frac{c}{a} & 1 \end{bmatrix}$$

Why Pivot?

Without pivoting, Gaussian elimination can be unstable – it can amplify the range of values, leading to round-off error in computers with limited precision.

In the example below, whose exact solution is $x_1 = \frac{10{,}000}{9{,}999}$ and $x_2 = \frac{9{,}998}{9{,}999}$, a hypothetical computer precise to 3 decimal digits rounds off the boxed coefficients. Without pivoting, an error results.

$$\left\{\begin{array}{rcrcl} .0001 & x_1 & + & x_2 & = 1 \\ & x_1 & + & x_2 & = 2 \end{array}\right\}$$

$$\swarrow \qquad\qquad \searrow$$

(no pivoting done) $$\left\{\begin{array}{rcrcl} & x_1 & + & x_2 & = 2 \\ .0001 & x_1 & + & x_2 & = 1 \end{array}\right\}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\left\{\begin{array}{rcrcl} .0001 & x_1 & + & x_2 & = 1 \\ & & - \boxed{10{,}000} & x_2 & = \boxed{\text{-}10{,}000} \end{array}\right\} \left\{\begin{array}{rcrcl} x_1 & + & x_2 & = & 2 \\ & & \boxed{x_2} & = & \boxed{1} \end{array}\right\}$$

$$\downarrow \qquad\qquad \downarrow$$

$x_2 = 1$ (correct to 3 places) $x_2 = 1$ (correct to 3 places)

$$\downarrow \qquad\qquad \downarrow$$

$x_1 = 0$ (wrong) $x_1 = 1$ (correct to 3 places)

Related Methods

- L-U Decomposition with Neighbor Pivoting
- Q-R Factorization by:
  - [Windowed] Householder Transformations
  - [Pipelined] Givens Rotations

Suggested Readings

- Dongarra, J. J., Gustavson, F. G., and Karp, A. Implementing Linear Algebra Algorithms for Dense Matrices on a Vector Pipeline Machine. *SIAM Review*, **26,** 1 (January 1984).
- Johnsson, S. Lennart. Communication Efficient Basic Linear Algebra Computations on Hypercube Architectures. *Journal of Parallel and Distributed Computing*, **4,** pp. 133-172 (1987).
- Trefethen, Lloyd N. and Schreiber, Robert S. Average-Case Stability of Gaussian Elimination. M.I.T. Department of Mathematics, *Numerical Analysis Report 88-3* (May 1988).
- Vavasis, Stephen. Implementation of QR factorization on the CM-2. Thinking Machines Corporation (17 September 1988).