Solving System of Linear Equations

System of linear equations.

n unknowns, m linear equations

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \end{cases}$$

In matrix notations, it can be written as

$$Ax = b,$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

coefficient matrix — unknown vector — right-hand side (RHS)

1. Solvability of system of linear equations

- The solution of $Ax=b$ must fall into one of the 3 cases.

  Case I: $Ax=b$ has no solution.

  Example: $\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ has no solution.

  Case II: $Ax=b$ has a unique solution.

  Example: $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$ has a unique solution $x = \begin{pmatrix} 2 \\ 3 \end{pmatrix}$.

  Case III: $Ax=b$ has infinitely many solutions.

Example: $\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$ has solutions in the form of $x = \begin{pmatrix} 1 \\ 1 \\ a \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ -a \end{pmatrix}$ $\forall a \in \mathbb{R}$.

- Thm: Let $A \in \mathbb{R}^{m \times n}$ and $b \in \mathbb{R}^m$. Then the linear equation $Ax = b$ is solvable (i.e., $\exists$ at least one $x \in \mathbb{R}^n$ s.t. $Ax = b$) if and only if $b \in Ran(A)$.

Proof. By the definition of $Ran(A)$.

Some results on the solutions of $Ax = b$

- $Ax = b$ has at least one solution, (i.e. either (Case II) or (Case III)) $\iff$ $b \in Ran(A)$ (By definition of A).
- $Null(A) \neq \{0\} \Rightarrow Ax = b$ cannot has a unique solution (i.e, either (Case I) or (Case III)) (proved by the definition of $Null(A)$)
- $m < n$ (i.e., less equations than unknowns) $\Rightarrow Ax = b$ cannot has a unique solution (i.e., either (Case I) or (Case III)) (proved by using the last statement)
- $m > n$ (i.e., more equations than unknowns) $\Rightarrow$ $\exists b$ s.t. $Ax = b$ has no solution. (i.e., $\exists b$ in (Case I))

We focus only on the case when A is nonsingular.

- Definition: An $n \times n$ square matrix A is non-singular if $Rank(A) = n$.
- A is nonsingular $\iff$ $Ran(A) = \mathbb{R}^n$
- Definition: An $n \times n$ square matrix A is invertible if there exists an $n \times n$ matrix, denoted by A^{-1}, such that $AA^{-1} = A^{-1}A = I$.
- Thm: Let $A \in \mathbb{R}^{n \times n}$ be an $n \times n$ square matrix. Then,

A is nonsingular $\iff$ A is invertible $\iff$ $Ax = b$ has a unique solution for any $b \in \mathbb{R}^n$.

proof. From a standard linear algebra textbook.

2. Gassian Elimination

An 3 X 3 Example:

$$\begin{cases} 2x_1+x_2-x_3=1 & ① \\ 4x_1+5x_2-3x_3=-3 & ② \\ -2x_1+5x_2-2x_3=-8 & ③ \end{cases}$$

Step 1: Eliminate x_1 in ② & ③

Row operations

① → ①

② − ① × 2 → ②

③ + ① × 1 → ③

$$\begin{cases} 2x_1+x_2-x_3=1 & ① \\ 3x_2-x_3=-5 & ② \\ 6x_2-3x_3=-7 & ③ \end{cases}$$

Step 2: Eliminate x_2 in ③

① → ①

② → ②

③ − ② × 2 → ③

$$\begin{cases} 2x_1+x_2-x_3=1 & ① \\ 3x_2-x_3=-5 & ② \\ -x_3=3 & ③ \end{cases}$$

easy to solve.

Step3: Solve the reduced linear equation

Solve ③, since it contains only x_3:

$$-x_3=3 \Rightarrow x_3=-3$$

solve ②, since it contains only x_2, x_3 and x_3 is solved.

$$3x_2-x_3=3x_2-(-3)=-5 \Rightarrow x_2=-\frac{8}{3}$$

Solve ①, sine x_2 and x_3 are already solved.

$$2x_1+x_2-x_3=2x_1-\frac{8}{3}-(-3)=1 \Rightarrow x_1=\frac{1}{3}$$

Matrix Reformulation:

$$\left[\begin{array}{ccc|c} 2 & 1 & -1 & 1 \\ 4 & 5 & -3 & -3 \\ -2 & 5 & -2 & -8 \end{array}\right] \quad \begin{matrix} & & & b \end{matrix}$$
(A)

Row operations are equivalent to left-multiplications

Step 1:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}}_{L_1} \left[\begin{array}{ccc|c} 2 & 1 & -1 & 1 \\ 4 & 5 & -3 & -3 \\ -2 & 5 & -2 & -8 \end{array}\right] \longrightarrow \left[\begin{array}{ccc|c} 2 & 1 & -1 & 1 \\ 0 & 3 & -1 & -5 \\ 0 & 6 & -3 & -7 \end{array}\right]$$

Step 2:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix}}_{L_2} \left[\begin{array}{ccc|c} 2 & 1 & -1 & 1 \\ 0 & 3 & -1 & -5 \\ 0 & 6 & -3 & -7 \end{array}\right] \longrightarrow \underbrace{\left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & 3 & -1 & -5 \\ 0 & 0 & -1 & 3 \end{array}\right]}_{\text{easy to solve}}$$

Step 3: Solve

$$\left[\begin{array}{ccc|c} 2 & 1 & 1 & 1 \\ 0 & 3 & -1 & -5 \\ 0 & 0 & -1 & 3 \end{array}\right] \longrightarrow \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{3} \\ -\frac{8}{3} \\ -3 \end{pmatrix}$$

last to solve

first to solve

The coefficient matrix is upper triangular.

$$Ax = b \longrightarrow L_1 A x = L_1 b \longrightarrow L_2 L_1 A x = L_2 L_1 b$$

$$\begin{bmatrix} x & x & x \\ x & x & x \\ x & x & x \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad L_1 \qquad \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{bmatrix} \quad L_2 \qquad \begin{bmatrix} 2 & 1 & 1 \\ 0 & 3 & -1 \\ 0 & 0 & -1 \end{bmatrix}$$

Gaussian Elimination as LU decomposition

We first show that

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

Proof: Let $v_1 = \begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix}$. Then $L_1 = I - v_1 e_1^T$.

Then $(I - v_1 e_1^T)(I + v_1 e_1^T) = I - v_1 e_1^T + v_1 e_1^T - v_1 e_1^T v_1 e_1^T$

$= I - (e_1^T v_1)\, v_1 e_1^T$

$e_1^T v_1 = (1\ 0\ 0)\begin{pmatrix} 0 \\ 2 \\ -1 \end{pmatrix} = 0$

$\Rightarrow (I - v_1 e_1^T)(I + v_1 e_1^T) = I$.

Similarly, $(I + v_1 e_1^T)(I - v_1 e_1^T) = I$.

So, $L_1^{-1} = I + v_1 e_1^T$. ☒

Similarly, $L_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{bmatrix}$

(Let $v_2 = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$. Then $L_2 = I - v_2 e_2^T$, $L_2^{-1} = I + v_2 e_2^T$)

Let $U = \begin{bmatrix} 2 & 1 & 1 \\ 0 & 3 & -1 \\ 0 & 0 & -1 \end{bmatrix}$

Since $L_2 L_1 A = U$, we have

$A = L_1^{-1} L_2^{-1} U$.

Let $L = L_1^{-1} L_2^{-1} = (I + v_1 e_1^T)(I + v_2 e_2^T)$

$= I + v_1 e_1^T + v_2 e_2^T + v_1 e_1^T v_2 e_2^T$

$= I + v_1 e_1^T + v_2 e_2^T + (e_1^T v_2)\, v_1 e_2^T$ ($e_1^T v_2 = (1\ 0\ 0)\begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} = 0$)

$= I + v_1 e_1^T + v_2 e_2^T$

$= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$

$$\Rightarrow \quad L = \begin{bmatrix} 2 & 1 & 0 \\ -1 & 2 & 1 \end{bmatrix}$$

Therefore,

$$A = \underbrace{\begin{bmatrix} 1 & & \\ 2 & 1 & \\ -1 & 2 & 1 \end{bmatrix}}_{\text{Lower triangular}} \underbrace{\begin{bmatrix} 2 & 1 & 1 \\ & 3 & -1 \\ & & -1 \end{bmatrix}}_{\text{upper triangular.}}$$

L $\qquad$ U

and $\quad L^{-1} = L_2 L_1$.

With these, the Gaussian Elimination can be viewed as:

Step I: Compute $A = LU$, where L is lower triangular with diagonals 1.
U is upper triangular,

(so that $Ax = b \iff LUx = b$
$\iff Ux = y$ and $Ly = b$.)

Step II: Solve $Ly = b$.
Step III: Solve $Ux = y$.
} all are easy to solve.

3. LU decomposition and Triangular systems.

We will present Steps I, II, and III for general n.
We start with Steps II and III.

Forward Substitution for lower triangular system

A matrix $L \in \mathbb{R}^{n \times n}$ is called Lower Trangular if $l_{ij} = 0 \ \forall \ i < j$.

i.e.

$$L = \begin{bmatrix} l_{11} & & & \\ l_{21} & l_{22} & & \\ \vdots & \vdots & \ddots & \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix}$$

To solve the linear equation $Lx = b$ where L is low triangular

We solve x_1 first, and then x_2, $\cdots$, and x_n the last.

For x_1, use the first equation

$$l_{11} x_1 = b_1 \Rightarrow x_1 = b_1 / l_{11}.$$

For x_2, use the 2nd eqution

$$l_{21} x_1 + l_{22} x_2 = b_2 \Rightarrow x_2 = (b_2 - l_{21} x_1) / l_{22}$$

$\vdots$

For x_i, use the i-th equation

$$\sum_{k=1}^{i-1} l_{ik} x_k + l_{ii} x_i = b_i \Rightarrow x_i = (b_i - \sum_{k=1}^{i-1} l_{ik} x_k) / l_{ii}$$

(x_k, $k = 1, \cdots, i-1$, already solved)

$\vdots$

This procedure is called Forward Substitution.

Example:

$$\begin{bmatrix} 1 & & \\ 2 & 1 & \\ -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -3 \\ -8 \end{bmatrix}$$

$x_1 = 1$

$2x_1 + x_2 = -3 \Rightarrow x_2 = -3 - 2x_1 = -3 - 2 \cdot 1 = -5$.

$-x_1 + 2x_2 + x_3 = -8 \Rightarrow x_3 = -8 + x_1 - 2x_2 = -8 + 1 - 2 \cdot (-5) = 3$.

Pseudo code:

```
for  i = 1:n
     for  k = 1: i-1
          b_i = b_i - l_ik x_k
     end
     x_i = b_i / l_ii
end
```

Other loops available. (e.g., ki-loop).

Computational complexity:

$$\sum_{i=1}^{n}\left(\left(\sum_{k=1}^{i-1} 2\right)+1\right) = \sum_{i=1}^{n}(2(i-1)+1) = \sum_{i=1}^{n}(2i-1) = 2\sum_{i=1}^{n} i - \sum_{i=1}^{n} 1$$

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

$$\sim O(n^2)$$

Back substitution for upper triangular system

A matrix $U \in \mathbb{R}^{n\times n}$ is called Upper Trangular if $u_{ij}=0 \ \forall\ i>j$.

i.e.

$$U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

To solve the linear equation $Ux=b$ where U is upper triangular

We solve x_n first, x_{n-1} the second, ---, and x_1 the last.

For x_n, use the last equation

$$u_{nn}x_n = b_n \Rightarrow x_n = b_n/u_{nn}.$$

For x_{n-1}, use the 2nd eqution

$$u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n = b_{n-1} \Rightarrow x_{n-1} = (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1}$$

$$\vdots$$

For x_i, use the i-th equation

$$u_{ii}x_i + \sum_{k=i+1}^{n} u_{ik}x_k = b_i \Rightarrow x_i = (b_i - \sum_{k=i+1}^{n} u_{ik}x_k)/u_{ii}$$

(x_k, $k=i+1,\cdots,n$, already solved)

$$\vdots$$

This procedure is known as Back Substitution.

Example:

$$\begin{bmatrix} 2 & 1 & 1 \\ & 3 & -1 \\ & & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -5 \\ 3 \end{bmatrix}$$

For x_3: $\quad -x_3 = 3 \Rightarrow x_3 = -3$

For x_2: $3x_2 - x_3 = -5 \Rightarrow x_2 = (-5+x_3)/2 = (-5+(-3))/3 = -\frac{8}{3}$

For x_1: $2x_1 + x_2 + x_3 = 1 \Rightarrow x_1 = (1 - x_2 - x_3)/2 = (1-(-\frac{8}{3})+3)/2 = \frac{10}{3}$

pseudo code:

```
for i = n : -1 : 1
    for k = i+1 : n
        b_i = b_i - u_ik x_k
    end
    x_i = b_i / u_ii
end
```

Other implementation available.

Computational cost:

$$\sum_{i=n:-1:1} \left(\left(\sum_{k=i+1}^{n} 2 \right) + 1 \right) = \sum_{i=n:-1:1} \left(2(n-i)+1 \right) = 2\sum_{i=n:-1:1} (n-i) + \sum_{i=n:-1:1} 1$$

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

$$\sim O(n^2)$$

Now let's present Step I — LU decomposition.

LU decomposition

Let $A \in \mathbb{R}^{n\times n}$ be an $n\times n$ matrix. Then there exist a decomposition,

$$A = LU,$$

where $L \in \mathbb{R}^{n\times n}$ is lower triangular with diagonals 1, and $U \in \mathbb{R}^{n\times n}$ is upper triangular.

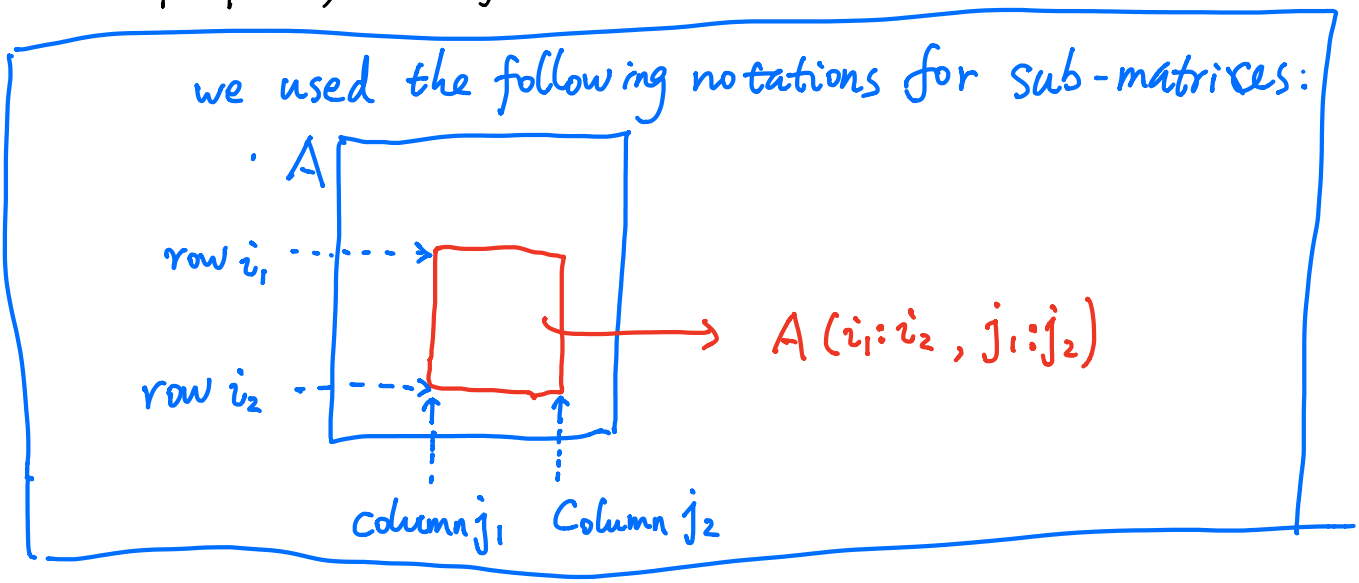
$$L = \begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ l_{31} & l_{32} & 1 & \\ \vdots & \vdots & \ddots & \ddots \\ l_{n1} & l_{n2} & \cdots & l_{n,n-1} & 1 \end{bmatrix} \qquad U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

How to find LU decomposition?

The Gaussian Elimination provide an algorithm.

Here we derive the LU decomposition directly, where L is obtained column-by column and U is obtained row by row

For this purpose, we first introduce some notations.



- At Sub-step 1:

$$\begin{bmatrix} 1 & & & \\ l_{21} & 1 & & \\ \vdots & l_{32} & \ddots & \\ l_{n1} & l_{n2} & \cdots & 1 \end{bmatrix} \qquad \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ & u_{22} & \cdots & u_{2n} \\ & & \ddots & \vdots \\ & & & u_{nn} \end{bmatrix}$$

— entries to be computed.

- Patition L and U as

$$L = \begin{bmatrix} 1 & 0 \\ L(2{:}n,1) & L(2{:}n,2{:}n) \end{bmatrix} \qquad U = \begin{bmatrix} U(1,1) & U(1,2{:}n) \\ 0 & U(2{:}n,2{:}n) \end{bmatrix}$$

Then,

$$LU = \begin{bmatrix} U(1,1) & U(1,2{:}n) \\ U(1,1)\,L(2{:}n,1) & L(2{:}n,1)\,U(1,2{:}n) + L(2{:}n,2{:}n)\,U(2{:}n,2{:}n) \end{bmatrix}$$

Comparing it with

$$A = \begin{bmatrix} A(1,1) & A(1,2{:}n) \\ A(2{:}n,1) & A(2{:}n,2{:}n) \end{bmatrix}$$

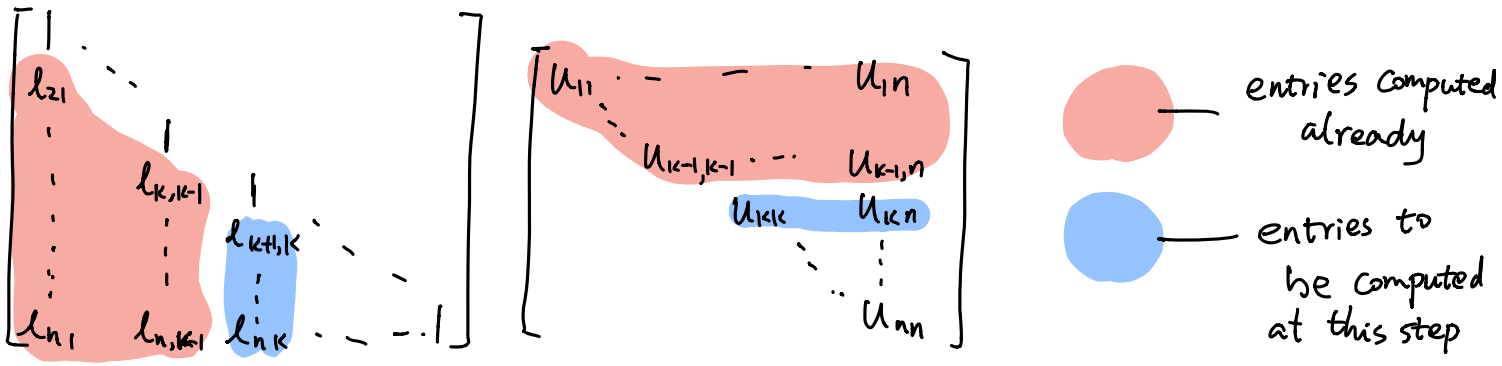
We obtain

$$\begin{cases} U(1,1) = A(1,1) \end{cases}$$

$$\begin{cases} U(1,2{:}n) = A(1,2{:}n) \\ U(1,1)L(2{:}n,1) = A(2{:}n,1) \Rightarrow L(2{:}n,1) = A(2{:}n,1)/U(1,1); \end{cases}$$

$\vdots$

- At sub-step k:



- Partition L and U as

$$L = \begin{bmatrix} L(1{:}k-1,1{:}k-1) & 0 & 0 \\ L(k,1{:}k-1) & 1 & 0 \\ L(k+1{:}n,1{:}k-1) & L(k+1{:}n,k) & L(k+1{:}n,k+1{:}n) \end{bmatrix}$$

$$U = \begin{bmatrix} U(1{:}k-1,1{:}k-1) & U(1{:}k-1,k) & U(1{:}k-1,k+1{:}n) \\ 0 & U(k,k) & U(k,k+1{:}n) \\ 0 & 0 & U(k+1{:}n,k+1{:}n) \end{bmatrix}$$

Then, by direct calculation,

$$LU = \begin{bmatrix} X & X & X \\ X & L(k,1{:}k-1)U(1{:}k-1,k) + U(k,k) & L(k,1{:}k-1)U(1{:}k-1,k+1{:}n) + U(k,k+1{:}n) \\ X & L(k+1{:}n,1{:}k-1)U(1{:}k-1,k) + U(k,k)L(k+1{:}n,k) & \bigstar \end{bmatrix}$$

Compare it with

$$A = \begin{bmatrix} A(1{:}k-1,1{:}k-1) & A(1{:}k-1,k) & A(1{:}k-1,k+1{:}n) \\ A(k,1{:}k-1) & A(k,k) & A(k,k+1{:}n) \\ A(k+1{:}n,1{:}k-1) & A(k+1{:}n,k) & A(k+1{:}n,k+1{:}n) \end{bmatrix}$$

We get

① $\begin{cases} L(k,1:k-1)\,U(1:k-1,k) + U(k,k) = A(k,k) \\ L(k,1:k-1)\,U(1:k-1,k+1:n) + U(k,k+1:n) = A(k,k+1:n) \end{cases}$, i.e.,

$$L(k,1:k-1)\;U(1:k-1,\,k:n) + U(k,k:n) = A(k,k:n),$$

which implies

$$U(k,k:n) = A(k,k:n) - L(k,1:k-1)\;U(1:k-1,\,k:n)$$

② $L(k+1:n,1:k-1)\,U(1:k-1,k) + U(k,k)\,L(k+1:n,k) = A(k+1:n,k)$,

which gives

$$L(k+1:n,k) = \left[A(k+1:n,k) - L(k+1:n,1:k-1)\;U(1:k-1,\,k)\right] / U(k,k)$$

⋮

In summary, the LU decomposition is done by

for $k = 1, 2, \cdots, n$

$$U(k,k:n) = A(k,k:n) - L(k,1:k-1)\;U(1:k-1,\,k:n)$$

$$L(k+1:n,k) = \left[A(k+1:n,k) - L(k+1:n,\,1:k-1)\;U(1:k-1,k)\right] / U(k,k)$$

end

In practice, a_{ij} can be overwritten by

$$\begin{cases} u_{ij} & \text{if } i \le j. \\ l_{ij} & \text{if } i > j. \end{cases}$$

pseudo-code for LU decomposition.

for $k = 1, 2, \cdots, n$

$$A(k,k:n) = A(k,k:n) - A(k,1:k-1)\;A(1:k-1,\,k:n)$$

$$A(k+1:n,k) = \left(A(k+1:n,k) - A(k+1:n,\,1:k-1)\;A(1:k-1,k)\right) / A(k,k)$$

end

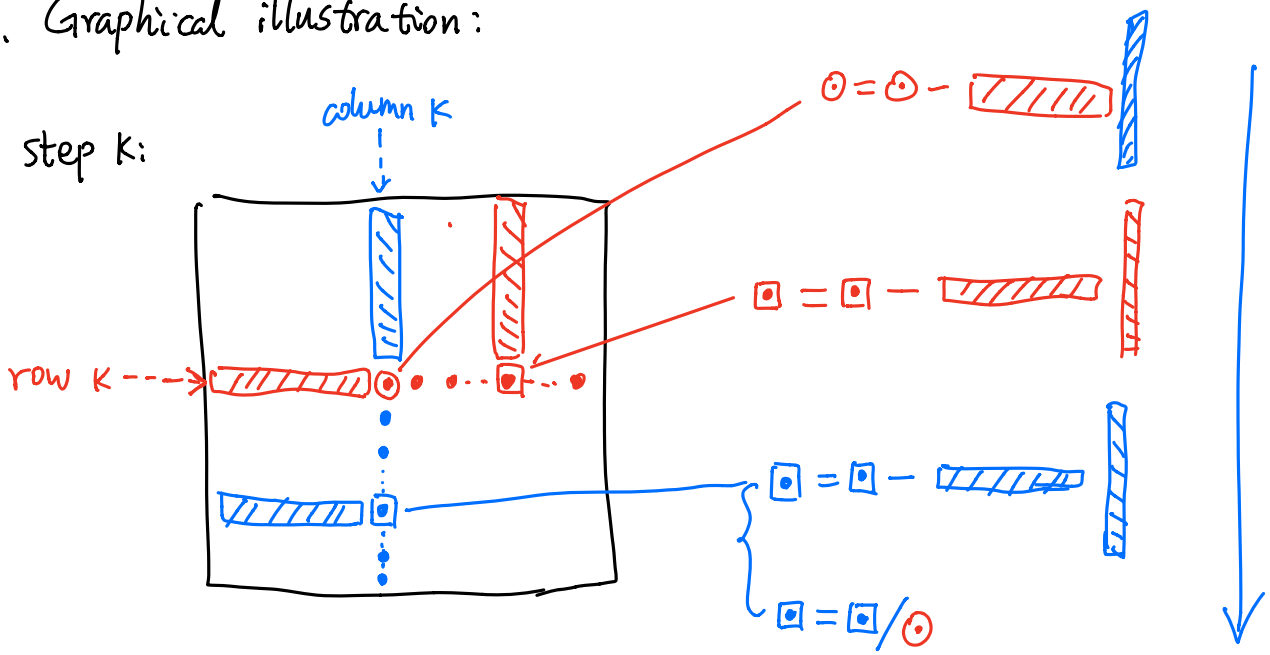
Remarks:

1. Due to the overwritting, the output of the above code is

$$L = \begin{pmatrix} 1 & & & \\ a_{21} & 1 & & \\ a_{31} & a_{32} & \ddots & \\ a_{n1} & a_{n2} & \cdots & 1 \end{pmatrix} \quad \text{and} \quad U = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix}$$

2. There are other implementations.

3. Graphical illustration:



Computational complexity:

In the first line:

$$A(k, k{:}n) = A(k, k{:}n) - \underbrace{A(k, 1{:}k-1)}_{1\times(k-1)}\,\underbrace{A(1{:}k-1, k{:}n)}_{(k-1)\times(n-k+1)}$$

So the complexity is $\underbrace{2(k-1)(n-k+1)}_{\text{for matrix-matrix product}} + \underbrace{(n-k+1)}_{\text{for subtraction}} = (2k-1)(n-k+1)$

In the 2nd line:

$$A(k+1{:}n, k) = \left(A(k+1{:}n, k) - \underbrace{A(k+1{:}n, 1{:}k-1)}_{(n-k)\times(k-1)}\,\underbrace{A(1{:}k-1, k)}_{(k-1)\times 1}\right) / A(k,k)$$

So the complexity is $\underbrace{2(k-1)(n-k)}_{\text{for m-m product}} + \underbrace{2(n-k)}_{\text{for div and subtraction}} = 2k(n-k)$

The total complexity is

$$\sum_{k=1}^{n}\left[(2k-1)(n-k+1)+2k(n-k)\right]$$
$$=\sum_{k=1}^{n}\left(4nk-4k^2+3k-n-1\right)$$
$$=4n\sum_{k=1}^{n}k-4\sum_{k=1}^{n}k^2+3\sum_{k=1}^{n}k-n\sum_{k=1}^{n}1-\sum_{i=1}^{n}1$$
$$=4n\frac{n(n+1)}{2}-4\cdot\frac{n(2n+1)(n+1)}{6}+3\frac{n(n+1)}{2}-n^2-n$$
$$=\frac{2}{3}n^3+\text{lower order terms (e.g. } n^2,\ n,\ \text{constant})$$
$$\sim O(n^3)$$

Example: Find LU Decomposition of

$$A=\begin{bmatrix}2&1&-1\\4&5&-3\\-2&5&-2\end{bmatrix}$$

$k=1$.
$$\begin{bmatrix}2&1&-1\\2&5&-3\\-1&5&-2\end{bmatrix}$$

$k=2$
$$\begin{bmatrix}2&1&-1\\2&3&-1\\-1&2&-2\end{bmatrix}$$

$k=3$
$$\begin{bmatrix}2&-1&-1\\2&3&-1\\-1&2&-1\end{bmatrix}=-2-(-1)\cdot(-1)-(2)\cdot(-1)$$

So
$$L=\begin{bmatrix}1&&\\2&1&\\-1&2&1\end{bmatrix}\qquad U=\begin{bmatrix}2&1&-1\\&3&-1\\&&-1\end{bmatrix}$$

Complete Algorithm for solving $Ax=b$.

for $k=1,2,\cdots,n$

$A(k,k:n)=A(k,k:n)-A(k,1:k-1)\,A(1:k-1,k:n)$

$A(k+1:n,k)=\left(A(k+1:n,k)-A(k+1:n,1:k-1)\,A(1:k-1,k)\right)/A(k,k)$

end

```
for i=1:n
    b_i = b_i - A(i,1:i-1) b(1:i-1,1)
end
```

LU decomposition

$A=LU$, where

a_{ij} is overwritten by $\begin{cases} l_{ij} & \text{if } i>j \\ u_{ij} & \text{if } i\le j. \end{cases}$

So the result is

$L=\begin{pmatrix} 1 & & \\ a_{21} & \ddots & \\ a_{n1} & \cdots & 1 \end{pmatrix}$ $U=\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ & a_{22} & \cdots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix}$

```
for i=n:-1:1
    b_i = b_i - A(i,i+1:n) b(i+1:n,1)
    b_i = b_i / A(i,i)
end
```

Back substitution for solving

$UX=y$,

where y is overwritten by X, i.e.,

$X=\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$ — the output of the solution of $AX=b$.

forward substitution for solving

$Ly=b$,

where b is overwritten by y.

So $y=\begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}$

Example: Solve

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & -1 & -5 \\ 2 & 7 & 9 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

Step 1: LU decomposition.

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & -1 & -5 \\ 2 & 7 & 9 \end{bmatrix} \to \begin{bmatrix} 2 & 1 & -1 \\ 2 & -1 & -5 \\ 1 & 7 & 9 \end{bmatrix} \to \begin{bmatrix} 2 & 1 & -1 \\ 2 & -3 & -3 \\ 1 & -2 & 9 \end{bmatrix} \to \begin{bmatrix} 2 & 1 & -1 \\ 2 & -3 & -3 \\ 1 & -2 & 4 \end{bmatrix}$$

So $L=\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & -2 & 1 \end{bmatrix}$ $U=\begin{bmatrix} 2 & 1 & -1 \\ & -3 & -3 \\ & & 4 \end{bmatrix}$

Step 2: Forward substitution

$$\begin{bmatrix} 1 & & \\ 2 & 1 & \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

$$y_1 = 0$$

$$2y_1 + y_2 = 0 \Rightarrow y_2 = 0 - 2y_1 = 0$$

$$y_1 - 2y_2 + y_3 = 4 \Rightarrow y_3 = 4 - y_1 + 2y_2 = 4$$

Step 3: Back substitution

$$\begin{bmatrix} 2 & 1 & -1 \\ & -3 & -3 \\ & & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix}$$

$$4x_3 = 4 \Rightarrow x_3 = 1$$

$$-3x_2 - 3x_3 = 0 \Rightarrow x_2 = (0 + 3x_3)/(-3) = -1$$

$$2x_1 + x_2 - x_3 = 0 \Rightarrow x_1 = (0 + x_3 - x_2)/2 = 1$$

Advantages of L.U decomposition formulation of GE (over standard GE)

1. No need to calculate $A = LU$ again if we solve linear systems of the same coefficient matrix A.
2. There are other usages of LU decomposition
   - —— Fast inversion of A.
     - ① $A = LU$
     - ② $A^{-1} = U^{-1}L^{-1}$
   - —— Determinate of A
     - ① $A = LU$
     - ② $\det(A) = \det(LU) = \det(L)\det(U) = u_{11}u_{22}\cdots u_{nn}$ (where $\det(L) = 1$)

— Others

3. Lead to potentially more efficient implementation.
4. LU can help on the theoretical analysis of GE.
5. Can utilize the structure of A. to obtain more efficient algorithms than the original GE.

Pivoting

In the k-th substep, $A(k,k)$ is used as a denominator.

- If $A(k,k) = 0$, then the LU decomposition cannot continue, even if A is invertible.

  Example: $$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

- Also, if $A(k,k)$ is very small, then the division by $A(k,k)$ will make L and U contrains large entries. Consequently, we may get large error in the solution due to round-off error of digital computers.

Example: $\begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$

Assume the computation is on a computer with 3 correct digits.

The exact solution is $x_{true} = \begin{pmatrix} 1.000100\ldots \\ 0.999900\ldots \end{pmatrix} \xrightarrow{\text{keep 3 digits}} \begin{pmatrix} 1 \\ 0.999 \end{pmatrix}$

LU decomposition:

$$\begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 10^{-4} & 1 \\ 10^{4} & 1 \end{bmatrix} \longrightarrow \begin{bmatrix} 10^{-4} & 1 \\ 10^{4} & 1-10^{4} \end{bmatrix}$$

$1 - 10^4 = -9999$, keep 3 digits to -9990.

So the computed LU decomposition is

$$L \approx \begin{bmatrix} 1 & \\ 10^4 & 1 \end{bmatrix} \quad U \approx \begin{bmatrix} 10^{-4} & 1 \\ & -9990 \end{bmatrix}$$

Forward substitution:

$$\begin{bmatrix} 1 & \\ 10^4 & 1 \end{bmatrix}\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \Rightarrow \begin{aligned} y_1 &= 1 \\ y_2 &= (2-10^4) = -9998 \xrightarrow{\text{keep 3 digits}} -9990 \end{aligned}$$

Back substitution:

$$\begin{pmatrix} 10^{-4} & 1 \\ & -9990 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ -9990 \end{pmatrix} \Rightarrow \begin{aligned} x_2 &= \frac{-9990}{-9990} = 1 \\ x_1 &= (1-x_2)/10^{-4} = \frac{1-1}{10^{-4}} = 0 \end{aligned}$$

So the computed solution is $x_{computed} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$,

which has a relative error (in 2-norm)

$$\frac{\|x_{computed} - x_{true}\|_2}{\|x_{true}\|_2} = \frac{\left\|\begin{pmatrix} 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0.999 \end{pmatrix}\right\|_2}{\left\|\begin{pmatrix} 1 \\ 0.999 \end{pmatrix}\right\|_2} \approx \frac{1}{\sqrt{2}} \approx 70.7\%$$

To overcome these, we introduce a technique called "pivoting".

We call $A(k,k)$ in the denominator in the k-th substep "pivot".

Partial Pivoting:

- permute the equations so that the pivot is as large as possible, i.e.,

we find a permutation matrix P s.t. the LU decomposition of PA

$$PA = LU,$$

where the "pivot" is as large as possible.

- We say $P \in \mathbb{R}^{n \times n}$ is a permutation matrix if there is only exactly a "1" in each row and each column and all other entries are 0.

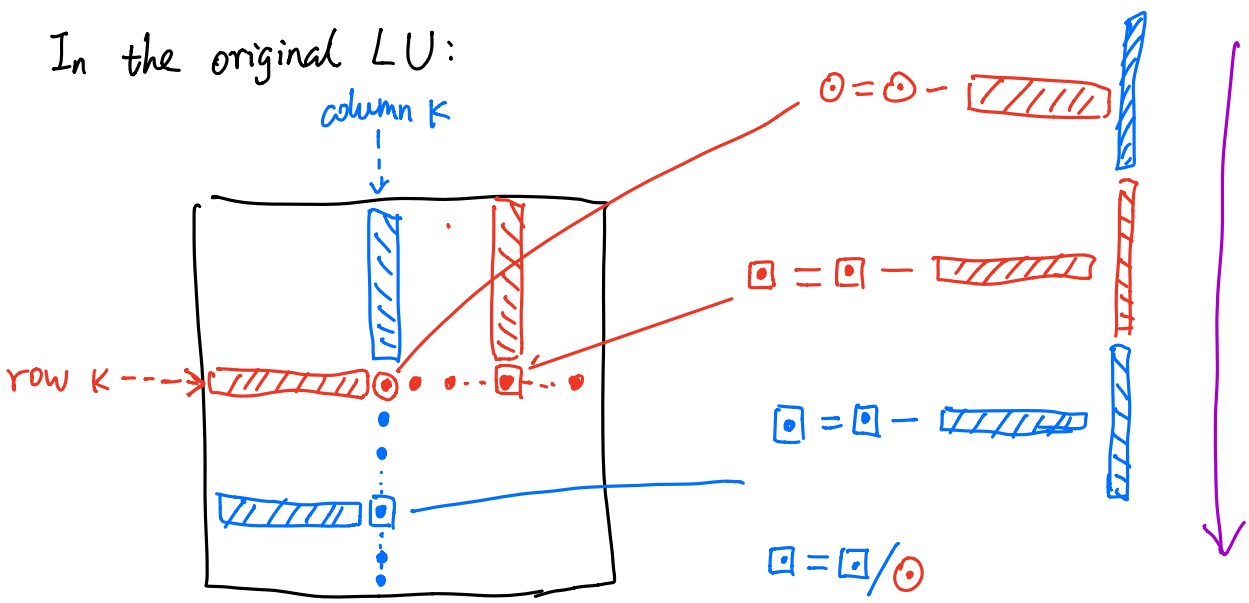
For examples:

$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ (we have $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} a_2 \\ a_1 \end{bmatrix}$, i.e., it permutes $\begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$ to $\begin{pmatrix} a_2 \\ a_1 \end{pmatrix}$)

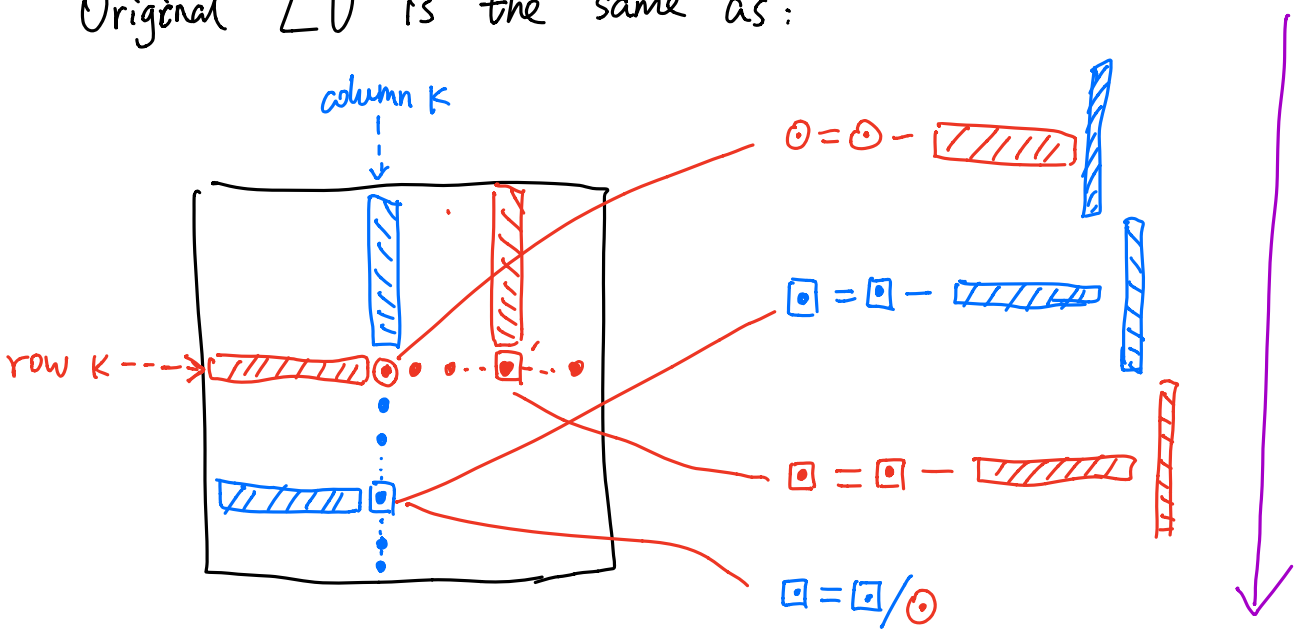
$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$ (we have $\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} a_2 \\ a_3 \\ a_1 \end{bmatrix}$, i.e., it permutes $\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ to $\begin{pmatrix} a_2 \\ a_3 \\ a_1 \end{pmatrix}$).

- To avoid a small $A(k,k)$ at sub-step k, we choose the max entry in $A(k{:}n, k)$ in magnitude as "pivot".

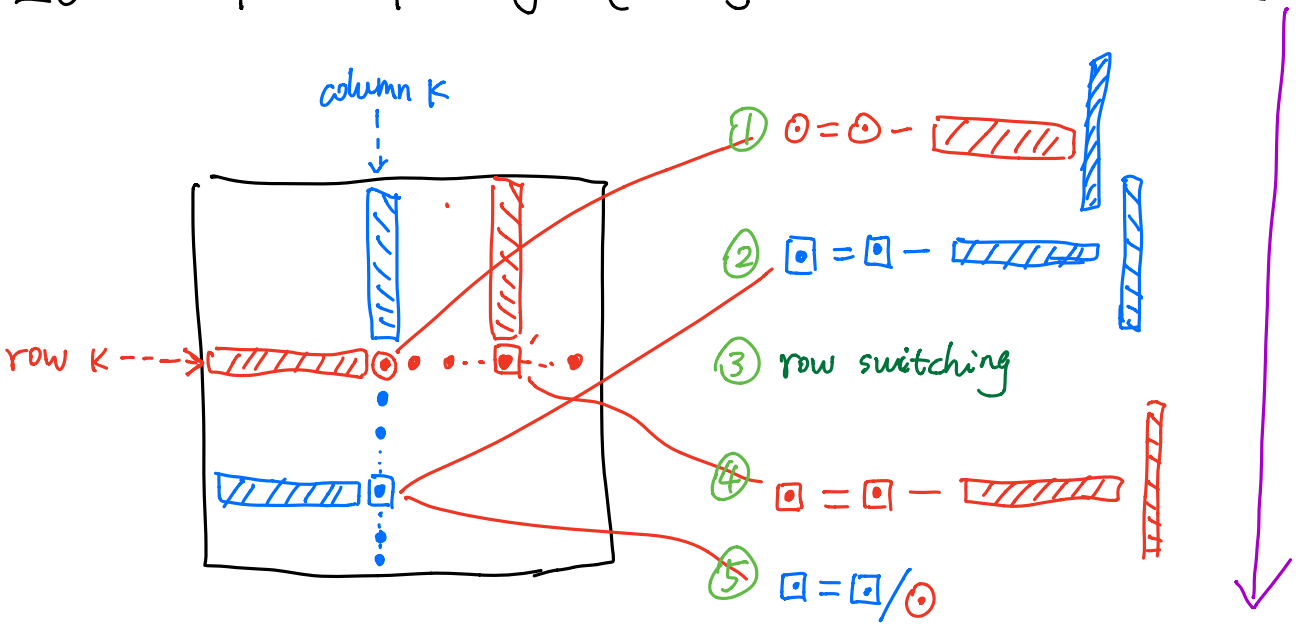
In the original LU:



Original LU is the same as:



LU with partial pivoting: (adding row switch before modify the row)



The modification of $A(k, k+1:n)$ has to be done after row switching

$P = I$

for $k = 1, 2, \cdots, n$

① ② $A(k:n, k) = A(k:n, k) - A(k:n, 1:k-1)\, A(1:k-1, k)$

③
$$j_k = \arg\max_{k \le j \le n} \{|A(j,k)|\}$$
$$A(k, 1:n) \longleftrightarrow A(j_k, 1:n)$$
$$P(k, 1:n) \longleftrightarrow P(j_k, 1:n)$$

④ $A(k, k+1:n) = A(k, k+1:n) - A(k, 1:k-1)\, A(1:k-1, k+1:n)$

⑤ $A(k+1:n, k) = A(k+1:n, k) / A(k,k)$

end

We will get $PA = LU$. Then,

$PAx = Pb \iff LUx = Pb \iff Ly = Pb$ and $Ux = y$.

Consequently, the forward substitution will solve $Ly = Pb$

and the back substitution still solves $Ux = y$.

- Example 1: $\begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ on a 3-digits computer

LU decomposition: $\begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 1 \\ 10^{-4} & 1 \end{bmatrix} \to \begin{bmatrix} 1 & 1 \\ 10^{-4} & 1-10^{-4} \end{bmatrix} \to \begin{bmatrix} 1 & 1 \\ 10^{-4} & 1 \end{bmatrix}$

$$\begin{bmatrix}1 & 1\\ 10^{-4} & 1\end{bmatrix} \uparrow \begin{bmatrix}10^{-4} & 1\\ 1 & 1\end{bmatrix} \to \begin{bmatrix}10^{-4} & 1\\ 1 & 1\end{bmatrix} \to \begin{bmatrix}10^{-4} & 1\\ 0 & 1-10^{-4}\end{bmatrix}$$

$$P=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix} \qquad \approx \begin{bmatrix}1 & 1\\ 10^{-4} & 1\end{bmatrix}$$

(keep 3 digits)

So $P=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$ $L=\begin{bmatrix}1 & \\ 10^{-4} & 1\end{bmatrix}$ $U=\begin{bmatrix}1 & 1\\ & 1\end{bmatrix}$ and $PA=LU$.

Forward-substitution:

$$\begin{bmatrix}1 & \\ 10^{-4} & 1\end{bmatrix}\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 2\end{bmatrix}=\begin{bmatrix}2\\ 1\end{bmatrix}$$

$$\Rightarrow \begin{cases} y_1 = 2 \\ y_2 = 1-10^{-4}\cdot y_1 = 1-2\cdot 10^{-4} = 0.9998 \overset{\text{keep 3 digits}}{=\!=} 0.999 \end{cases}$$

Back substitution:

$$\begin{bmatrix}1 & 1\\ & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}=\begin{bmatrix}2\\ 0.999\end{bmatrix}$$

$$\Rightarrow \begin{cases} x_2 = 0.999 \\ x_1 = 2-x_2 = 2-0.999 = 1.001 \overset{\text{keep 3 digits}}{=\!=} 1.000 \end{cases}$$

So the computed solution is $x_{computed}=\begin{pmatrix}1.000\\ 0.999\end{pmatrix}$, and the relative error is $$\frac{\|x_{computed}-x_{true}\|_2}{\|x_{true}\|_2}=\frac{\left\|\begin{pmatrix}1.000\\ 0.999\end{pmatrix}-\begin{pmatrix}1\\ 0.999\end{pmatrix}\right\|_2}{\left\|\begin{pmatrix}1\\ 0.999\end{pmatrix}\right\|_2}=0\%.$$

- Example 2: Find the partially pivoted LU decomposition of

$$A=\begin{bmatrix}2 & 1 & -1\\ 4 & 5 & -3\\ -2 & 5 & -2\end{bmatrix}$$

(Ignore the round-off errors)

$$\begin{bmatrix}2 & 1 & -1\\ \textcircled{4} & 5 & -3\\ -2 & 5 & -2\end{bmatrix} \to \uparrow \begin{bmatrix}4 & 5 & -3\\ 2 & 1 & -1\\ -2 & 5 & -2\end{bmatrix} \to \begin{bmatrix}4 & 5 & -3\\ \frac{1}{2} & 1 & -1\\ -\frac{1}{2} & 5 & -2\end{bmatrix} \to$$

$$P=\begin{bmatrix}0&1&0\\1&0&0\\0&0&1\end{bmatrix}$$

$$\begin{bmatrix}4&5&-3\\ \frac{1}{2}&-\frac{3}{2}&-1\\ -\frac{1}{2}&\frac{15}{2}&-2\end{bmatrix}\longrightarrow\begin{bmatrix}4&5&-3\\ -\frac{1}{2}&\frac{15}{2}&-2\\ \frac{1}{2}&-\frac{3}{2}&-1\end{bmatrix}\rightarrow\begin{bmatrix}4&5&-3\\ -\frac{1}{2}&\frac{15}{2}&-\frac{7}{2}\\ \frac{1}{2}&-\frac{1}{5}&-1\end{bmatrix}\rightarrow\begin{bmatrix}4&5&-3\\ -\frac{1}{2}&\frac{15}{2}&-\frac{7}{2}\\ \frac{1}{2}&-\frac{1}{5}&-\frac{1}{5}\end{bmatrix}$$

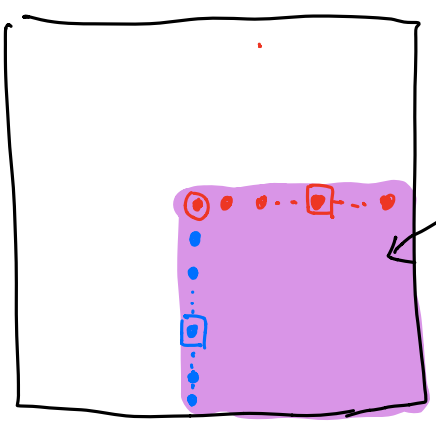
$$P=\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}$$

Therefore,

$$\begin{bmatrix}0&1&0\\0&0&1\\1&0&0\end{bmatrix}\begin{bmatrix}2&1&-1\\4&5&-3\\-2&5&-2\end{bmatrix}=\begin{bmatrix}1&&\\-\frac{1}{2}&1&\\\frac{1}{2}&-\frac{1}{5}&1\end{bmatrix}\begin{bmatrix}4&5&-3\\&\frac{15}{2}&-\frac{7}{2}\\&&-\frac{1}{5}\end{bmatrix}$$

$$P\quad A\quad =\quad L\quad U$$

- In partially pivoting, the strict lower triangular part of L has an absolute value smaller than 1, because the numerator is always smaller than the denominator (i.e., the pivot).
- There are other pivoting strategies, e.g., fully pivoting, where permutation on both equations and unknowns are imposed.



Full pivoting: Choose "pivot" from by switching both rows and columns.

Then, we have $PAQ=LU$, where P, Q are both permutations.

- Matrix decomposition/factorization is the main theme of matrix computation. We will learn many other matrix factorizations such as QR, eigenvalue decomposition, singular value decomposition.

- there are other decompositions that can be used to solve $Ax=b$. For example, we can use QR decomposition to solve $Ax=b$.

Exploiting Structures of A

Both memory and computation can be saved by exploiting the structure of A in solving $Ax=b$.

Symmetric Positive Definite (SPD) Matrices

Definition: A matrix $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite (SPD) if

① $A = A^T$ (A is symmetric) and

② $x^TAx > 0$ for any $x \in \mathbb{R}^n$ and $x \neq 0$ (A is positive definite)

If the ">0" in ② is replaced by "≥ 0", then A is called symmetric positive semi-definite (SPSD).

Facts: (a) $A = CC^T$ for some $C \in \mathbb{R}^{n\times m}$ is SPSD. In particular, if $C \in \mathbb{R}^{n\times n}$ is non-singular, then $A = CC^T$ is SPD.

(b) If A is SPD, then A is nonsingular.

(c) If A is SPD (SPSD respectively), then $A(i_1:i_2, i_1:i_2)$ is SPD (SPSD respectively) for $i_2 \geq i_1$. In particular, all diagonal entries a_{ii} of A are positive (non-negative respectively) if A is SPD (SPSD respectively).

Examples of SPSD/SPD matrices:

(a) The Hessian of a convex function is SPSD. In particular, the Hessian of a strictly convex function is SPD.

(b) The covariance matrix is SPSD.

(c) $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ is SPD.

Cholesky decomposition:

If $A \in \mathbb{R}^{n\times n}$ is SPD, then there exists a decomposition

$$A = LL^T,$$

where $L \in \mathbb{R}^{n\times n}$ is lower triangular (NOT necessarily unit lower triangular).

Algorithms for Cholesky decomposition.

Compute $L \in \mathbb{R}^{n\times n}$ column by column.

Sub-step 1: Partition L as $L = \begin{bmatrix} L(1,1) & 0 \\ L(2:n,1) & L(2:n,2:n) \end{bmatrix}$ — entries to be computed.

Then $LL^T = \begin{bmatrix} (L(1,1))^2 & * \\ L(2:n,1)L(1,1) & * \end{bmatrix}$

Compare it with

$$A = \begin{bmatrix} A(1,1) & * \\ A(2:n,1) & * \end{bmatrix}$$

we get

$$L(1,1)^2 = A(1,1) \Rightarrow L(1,1) = (A(1,1))^{1/2}$$

$$L(2:n,1)L(1,1) = A(2:n,1) \Rightarrow L(2:n,1) = A(2:n,1)/L(1,1)$$

Remarks:
- SPD-ness of A guarantees $A(1,1) > 0$.
- We may also choose $L(1,1) = -A(1,1)^{1/2}$. Nevertheless, in the standard Cholesky decomposition, we choose $L(1,1) = (A(1,1))^{1/2}$.

⋮

Sub-step K: Partition L as

$$L = \begin{bmatrix} L(1:k-1,\,1:k-1) & 0 & 0 \\ L(k,\,1:k-1) & L(k,k) & 0 \\ L(k+1:n,\,1:k-1) & L(k+1:n,k) & L(k+1:n,k+1:n) \end{bmatrix}$$

— entries to be computed

— entries already computed.

Then

$$LL^T = \begin{bmatrix} * & * & * \\ * & L(k,1:k-1)\left(L(k,1:k-1)\right)^T + \left(L(k,k)\right)^2 & * \\ * & L(k+1:n,1:k-1)\left(L(k,1:k-1)\right)^T + L(k+1:n,k)L(k,k) & * \end{bmatrix}$$

Compare it with

$$A = \begin{bmatrix} * & * & * \\ * & A(k,k) & * \\ * & A(k+1:n,k) & * \end{bmatrix}$$

We obtain;

$$L(k,1:k-1)\left(L(k,1:k-1)\right)^T + \left(L(k,k)\right)^2 = A(k,k)$$

$$\Rightarrow L(k,k) = \left(A(k,k) - L(k,1:k-1)\left(L(k,1:k-1)\right)^T\right)^{1/2}$$

↳ The SPD-ness of A ensures this number is positive

$$L(k+1:n,1:k-1)\left(L(k,1:k-1)\right)^T + L(k+1:n,k)L(k,k) = A(k+1:n,k)$$

$$\Rightarrow L(k+1:n,k) = \left(A(k+1:n,k) - L(k+1:n,1:k-1)\left(L(k,1:k-1)\right)^T\right)/L(k,k)$$

$$\vdots$$

Repeat until n-substeps.

So, the full Cholesky decomposition algorithm is

```
for k=1:n
    L(k,k) = (A(k,k) - L(k,1:k-1)*(L(k,1:k-1))')^(1/2)
    L(k+1:n,k) = (A(k+1:n,k) - L(k+1:n,1:k-1)*(L(k,1:k-1))')/L(k,k)
end
```

To save memory, note that $\begin{matrix} A(k,k) \\ A(k+1:n,k) \end{matrix}$ are used only for $\begin{matrix} L(k,k) \\ L(k+1:n,k) \end{matrix}$ at sub-step k, so $\begin{matrix} A(k,k) \\ A(k+1:n,k) \end{matrix}$ can be overwritten by $\begin{matrix} L(k,k) \\ L(k+1:n,k) \end{matrix}$.

Thus, a memory effcient version is

for $k=1:n$

$$A(k,k) = \left(A(k,k) - A(k,1:k-1)\left(A(k,1:k-1)\right)^T\right)^{1/2}$$

$$A(k+1:n,k) = \left(A(k+1:n,k) - A(k+1:n,1:k-1)\left(A(k,1:k-1)\right)^T\right)/A(k,k)$$

end

Remark: 1. The memory is $\frac{n(n+1)}{2}$, which is about half of standard LU.

2. The computational cost is $\frac{1}{3}n^3 +$ lower order terms, which is about half of the standard LU decomposition.

3. The SPD-ness of A implies that $A(k,k) - L(k,1:k-1)\left(L(k,1:k-1)\right)^T$ is positive. Therefore, no pivoting is not necessary.

Example:

$$A = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & * & * \\ -1 & 2 & * \\ 0 & -1 & 2 \end{bmatrix} \longrightarrow \begin{bmatrix} \sqrt{2} & * & * \\ -\frac{1}{\sqrt{2}} & 2 & * \\ 0 & -1 & 2 \end{bmatrix} \longrightarrow \begin{bmatrix} \sqrt{2} & * & * \\ -\frac{1}{\sqrt{2}} & \frac{3}{2} & * \\ 0 & -\frac{2}{3} & 2 \end{bmatrix}$$

$k=1$ $\qquad$ $k=2$

$\left(2-\left(-\frac{1}{\sqrt{2}}\right)^2\right)^{1/2} = \sqrt{\frac{3}{2}}$ $\qquad$ $\left((-1) - 0\cdot\left(-\frac{1}{\sqrt{2}}\right)\right)/\sqrt{\frac{3}{2}}$

$\left(2-\left(-\frac{2}{3}\right)^2\right)^{1/2} = \sqrt{\frac{4}{3}}$

Therefore

$$L = \begin{bmatrix} \sqrt{2} & 0 & 0 \\ -\frac{1}{\sqrt{2}} & \frac{3}{2} & 0 \\ 0 & -\frac{2}{3} & \frac{4}{3} \end{bmatrix} \quad \text{and} \quad A = LL^T.$$

Tridiagonal matrices.

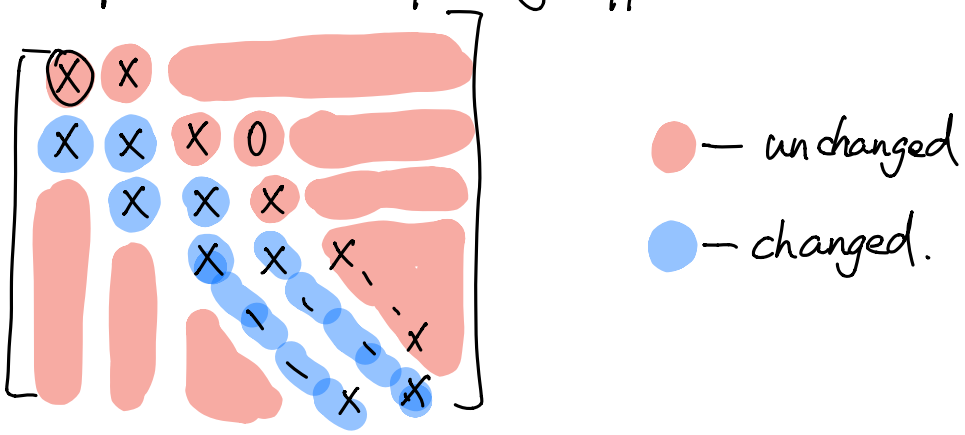
$A \in \mathbb{R}^{n\times n}$ is a tridiagonal matrix if

$$a_{ij} = 0 \quad \text{for} \quad |i-j| > 1.$$

i.e.,

$$A = \begin{bmatrix} x & x & & & \\ x & x & x & & 0 \\ & x & x & x & \\ & & \ddots & \ddots & \ddots \\ & 0 & & x & x \\ & & & x & x \end{bmatrix}$$

The LU decomposition without pivoting applied to A:



We see that:

(a). Only $2(n-1)$ entries are modified. So, the LU decomposition for tridiagonal matrix A can be simplified to:

```
for k=1:n
    A(k,k) = A(k,k) - A(k,k-1) A(k-1,k)
    A(k+1,k) = A(k+1,k) / A(k,k)
end
```

Obvious, the computational cost is $O(n)$.

(b). The resulting matrix is still tridiagonal, so that

$$L = \begin{bmatrix} 1 & & & \\ x & 1 & & \\ & x & \ddots & \\ & & x & 1 \end{bmatrix} \qquad U = \begin{bmatrix} x & x & & \\ & x & x & \\ & & \ddots & x \\ & & & x \end{bmatrix}$$

(L and U are also bi-diagonal.)

Consequently, the forward substitution (solving $Ly=b$) and back substitution (solving $Ux=y$) can be done in $O(n)$ as well.

(c). Altogether, solving $Ax=b$ with tridiagonal A can be done in $O(n)$ computations. As a comparison, general dense A needs

$O(n^3)$ (in particular $\frac{2}{3}n^3$) computations.