

## Décompositions matricielles

Vincent Nozick



## Décomposition LU

Qu'est-ce que c'est ?

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

## Décomposition LU

Qu'est-ce que c'est ?



Décomposition de la matrice M :

$$M = LU$$

avec :

- M, L et U : 3 matrices  $n \times n$
- L : matrice unitaire triangulaire inférieure
- U : matrice triangulaire supérieure

## Décomposition LU

Méthode :

On fait comme un pivot de Gauss, mais on effectue les opérations de pivot sous forme de produits de matrices.

## Décomposition LU

Exemple :

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & 6 & 1 \\ -2 & 11 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & 6 & 1 \\ -2 & 11 & 8 \end{bmatrix} \begin{matrix} L'_2 = L_2 - 2L_1 \\ L'_3 = L_3 + L_1 \\ \rightarrow \end{matrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix}$$

## Décomposition LU

Exemple :

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & 6 & 1 \\ -2 & 11 & 8 \end{bmatrix} \begin{matrix} L'_2 = L_2 - 2L_1 \\ L'_3 = L_3 + L_1 \\ \rightarrow \\ L_2 = L'_2 + 2L_1 \\ L_3 = L'_3 - L_1 \end{matrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & 6 & 1 \\ -2 & 11 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix}$$

## Décomposition LU

Exemple :

$$\begin{bmatrix} 2 & 1 & -1 \\ 4 & 6 & 1 \\ -2 & 11 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix} \begin{matrix} L'_3 = L_3 - 3L_2 \\ \rightarrow \end{matrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 0 & -2 \end{bmatrix}$$

## Décomposition LU

Exemple :

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix} \begin{matrix} L'_3 = L_3 - 3L_2 \\ \rightarrow \\ L_3 = L'_3 + 3L_2 \end{matrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 0 & -2 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 0 & -2 \end{bmatrix}$$

## Décomposition LU

Exemple :

$$\begin{aligned}
 M &= \begin{bmatrix} 2 & 1 & -1 \\ 4 & 6 & 1 \\ -2 & 11 & 8 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 12 & 7 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 0 & -2 \end{bmatrix} \\
 &= \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 3 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 0 & 4 & 3 \\ 0 & 0 & -2 \end{bmatrix} = LU
 \end{aligned}$$

## Méthode

---

**Algorithm 1:** Décomposition LU

---

**input:** une matrice  $M_{n \times n}$  à décomposer

```

1 L = Id
2 foreach pivot i de 1 à n - 1 do
3   L̂ = Id
4   foreach ligne j de i + 1 à n do
5     L̂j,i = -Mj,i/Mi,i
6   M = L̂M
7   L = LL̂-1      (L̂-1 facile à calculer)
8 U = M
9 return L et U

```

---

## Décomposition LU

Propriétés :

- Décomposition de M existe ssi  $\det(M) \neq 0$
- $\det(M) = \det(L) \times \det(U) = \det(U)$
- la décomposition  $M = LU$  est unique

## Application

Résolution de système linéaire :

$$Mx = b$$

- $LUx = b$
- on résout :  $Ly = b$  (élimination)
- on résout :  $Ux = y$  (élimination)

## Application

### Inverser une matrice :

$$MM^{-1} = \text{Id}$$

- décomposition :  $M = LU$
- on résoud  $n$  systèmes :  $LU M_i^{-1} = \mathbf{e}_i$  ( $A_i$  :  $i$ -ème colonne de  $A$ )

→ inverse utilisée par matlab par défaut.

## Application

### Calcul du déterminant :

$$\underbrace{\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}}_M = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix}}_L \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}}_U$$

$$\det(M) = \pm \det(L) \cdot \det(U) = \pm \prod_i u_{i,i}$$

Le signe du déterminant dépend du nombre de permutations de lignes effectuées durant la décomposition.

## Application

### Réduction d'erreur :

Soit le système :  $A\mathbf{x} = \mathbf{b}$

pour lequel on trouve une solution  $\mathbf{x}_0$  telle que :

$$A\mathbf{x}_0 = \mathbf{b}_0 \simeq \mathbf{b} \quad (\text{erreurs numériques})$$

On a

$$A(\mathbf{x}_0 + \delta\mathbf{x}) = \mathbf{b} \Leftrightarrow A\mathbf{x}_0 + A\delta\mathbf{x} = \mathbf{b}$$

soit

$$A\delta\mathbf{x} = \mathbf{b} - A\mathbf{x}_0$$

On résoud le système et on obtient :  $\mathbf{x} = \mathbf{x}_0 + \delta\mathbf{x}$

La décomposition de  $A$  est utilisée pour résoudre les 2 systèmes.

## Décomposition QR

### Qu'est-ce que c'est ?

Décomposition de la matrice  $M$  :  $M = QR$

avec :

- $M$ ,  $Q$  et  $R$  : 3 matrices  $n \times n$
- $Q$  : matrice de rotation ( $Q^T Q = \text{Id}$ )
- $R$  : matrice triangulaire supérieure

## Décomposition QR

### Qu'est-ce que c'est ?

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

## Décomposition QR

$$M = QR$$

### Applications :

- résoudre un système linéaire :  $R\mathbf{x} = Q^T \mathbf{b}$
- inverse de matrice : comme pour la décomposition LU.
- calcul du déterminant :  $\det(M) = \pm \det(R) = \pm \prod_i R_{i,i}$
- trouver les racines d'un polynôme.

## Méthode

### Algorithm 2: Décompositon QR

**input:** une matrice  $M_{n \times n}$  à décomposer

```

1 Q = Id
2 foreach block-colonne m = Mii(n - i + 1) do
3   α = ||m||2
4   if m1 < 0 then α = -α
5   u = mi - αei
6   v =  $\frac{u}{\|u\|_2}$ 
7   Q̂ = Idn
8   Q̂ii(n - 1 + i × n - i + 1) = 2vvT
9   M = Q̂M
10  Q = QQ̂
11 R = M
12 return Q et R

```

## Cholesky

### Introduction :

Permet de décomposer une matrice symétrique définie positive A en  $A = LL^T$  où L est une matrice triangulaire inférieure.

### Définie positive :

A est définie positive si :  $\forall \mathbf{x} \neq \mathbf{0} \quad \mathbf{x}^T A \mathbf{x} > 0$

## Calcul

Colonne par colonne :

$$L_{ii} = \sqrt{A_{ii} - \sum_{k=0}^{k < i} L_{ik}^2} \quad \text{pour } i = j$$

$$L_{ij} = \frac{1}{L_{jj}} \left( A_{ij} - \sum_{k=0}^{k < j} L_{ik} L_{jk} \right) \quad \text{pour } i > j$$

## Diagonalisation

$$A = PMP^{-1}$$

Propriétés :

- les éléments de M sont les valeurs propres de A.
- rang de A : nombre de valeurs propres  $\neq 0$ .
- les colonnes de P sont les vecteurs propres de A associés aux valeurs propres correspondantes de M.

## Diagonalisation

Introduction :

$$A = PMP^{-1}$$

avec :

- A : matrice  $n \times n$ .
- P : matrice  $n \times n$ .
- M : matrice  $n \times n$  diagonale.

## Diagonalisation

Application :

- inverse (peu utilisée)
- Analyse en Composantes Principales (*PCA* en anglais)

# Décomposition en Valeurs Singulières

**Introduction :** (SVD = *Singular Value Decomposition*)

$$A = UDV^T$$

avec :

- $A$  : matrice  $m \times n$ ,  $m \geq n$ .
- $U$  : matrice  $m \times m$ , orthogonale si  $m = n$ .
- $D$  : matrice  $n \times n$  diagonale.
- $V^T$  : matrice  $n \times n$  orthogonale.

# SVD

**Propriétés :**

- les éléments de  $D$  sont les valeurs singulières de  $A$ .
- rang de  $A$  : nombre de valeurs singulières  $\neq 0$ .
- en annulant les plus petites valeurs singulières, on obtient des matrices de rang inférieur à  $A$  les plus proches de  $A$ .

# SVD

**Calcul de  $A^{-1}$  :**

$$A = UDV^T$$

$$\begin{aligned} A^{-1} &= (UDV^T)^{-1} \\ &= V^{-T} D^{-1} U^{-1} \\ &= V D^{-1} U^T \end{aligned}$$

- $V^{-T} = V$  car  $V$  est orthogonale.
- $U^{-1} = U^T$  car  $U$  est orthogonale.
- $D$  est diagonale donc facile à inverser.

# Moindres carrés

**Propriété :**

Si  $A$  n'est pas carrée (lignes  $>$  colonnes), l'inverse calculé à partir de la SVD génère la pseudo inverse  $A^+$  de  $A$  :

$$A^+ = V D^{-1} U^T$$

$\Leftrightarrow$  permet de résoudre des systèmes **surdéterminés** au sens des moindres carrés.

## Null space

### Propriété :

La SVD permet aussi de résoudre des systèmes d'équations du genre :

$$Ax = \mathbf{0}$$

- Pivot de Gauss  $\rightarrow$  solution triviale  $\mathbf{x} = \mathbf{0}$
- SVD  $\rightarrow$  vecteur  $\mathbf{x} \neq \mathbf{0}$  tel que :
  - $\|\mathbf{x}\|_2 = 1$ .
  - $\|A\mathbf{x}\|_2$  soit minimal.
  - solution : il s'agit du vecteur propre (colonne de  $V$ ) associé à la plus petite valeur singulière de  $D$ .

## Décompositions

### Bilan :

décomposition	conditions matrice	rapidité	fiabilité précision	rang
LU pivot partiel	matrice régulière	++	+	•
LU pivot total	•	–	+++	oui
QR Householder	•	++	+	•
QR pivot partiel	•	+	++	oui
QR pivot total	•	–	+++	oui
Cholesky	symétrique déf. $> 0$	++++	+	•
SVD	•	–	++++	oui