# Analyse Numérique I

## Sup'Galilée, Ingénieurs MACS 1ère année & L3 MIM

François Cuvelier

Laboratoire d'Analyse Géométrie et Applications
Institut Galilée
Université Paris XIII.

2021/10/27

# Chapitre IV

## Résolution de systèmes linéaires

# Plan

1. Conditionnement
2. Méthodes directes
   - Matrices particulières
     - Matrices diagonales
     - Matrices triangulaires inférieures
     - Matrices triangulaires supérieures
   - Exercices et résultats préliminaires
   - Méthode de Gauss-Jordan
     - Ecriture algébrique
   - Factorisation $\mathbb{LU}$
     - Résultats théoriques
     - Utilisation pratique
   - Factorisation $\mathbb{LDL}^*$
   - Factorisation de Cholesky
     - Résultats théoriques
     - Résolution d'un système linéaire
     - Algorithme : Factorisation positive de Cholesky
   - Factorisation $\mathbb{QR}$
     - La tranformation de Householder

Soient $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ une matrice inversible et $\boldsymbol{b} \in \mathbb{K}^n$.

Résoudre

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$$

Le calcul de la matrice inverse $\mathbb{A}^{-1}$ revient à résoudre $n$ systèmes linéaires.

⚠ Pour résoudre un système linéaire, on ne calcule pas la matrice inverse associée.

- **Méthodes directes :** On cherche $\mathbb{M}$ inversible tel que $\mathbb{M}\mathbb{A}$ *facilement* inversible

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \iff \mathbb{M}\mathbb{A}\boldsymbol{x} = \mathbb{M}\boldsymbol{b}.$$

- **Méthodes itératives :** On cherche $\mathbb{B}$ et $\boldsymbol{c}$,

$$\boldsymbol{x}^{[k+1]} = \mathbb{B}\boldsymbol{x}^{[k]} + \boldsymbol{c}, \quad k \geqslant 0, \ \boldsymbol{x}^{[0]} \text{ donné}$$

en espérant $\lim_{k\to+\infty} \boldsymbol{x}^{[k]} = \boldsymbol{x}$.

# Conditionnement

Soient $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ inversible et $\boldsymbol{b} \in \mathbb{K}^n$.

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$$

De petites erreurs sur les données engendrent-elles de petites erreurs sur la solution?

# Exemple de R.S. Wilson

Soient

$$\mathbb{A} = \begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix}, \quad \Delta\mathbb{A} = \begin{pmatrix} 0 & 0 & \frac{1}{10} & \frac{1}{5} \\ \frac{2}{25} & \frac{1}{25} & 0 & 0 \\ 0 & -\frac{1}{50} & -\frac{11}{100} & 0 \\ -\frac{1}{100} & -\frac{1}{100} & 0 & -\frac{1}{50} \end{pmatrix}$$

et $\boldsymbol{b}^{t} = (32, 23, 33, 31)$, $(\boldsymbol{\Delta b})^{t} = \left(\frac{1}{100}, -\frac{1}{100}, \frac{1}{100}, -\frac{1}{100}\right)$. Des calculs exacts donnent

$$\begin{aligned}
\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \quad &\Longleftrightarrow \quad \boldsymbol{x}^{t} = (1, 1, 1, 1) \\
\mathbb{A}\boldsymbol{u} = (\boldsymbol{b} + \boldsymbol{\Delta b}) \quad &\Longleftrightarrow \quad \boldsymbol{u}^{t} = \left(\frac{91}{50}, -\frac{9}{25}, \frac{27}{20}, \frac{79}{100}\right) \\
&\qquad\qquad\quad \approx (1.8, -0.36, 1.3, 0.79) \\
(\mathbb{A} + \Delta\mathbb{A})\boldsymbol{v} = \boldsymbol{b} \quad &\Longleftrightarrow \quad \boldsymbol{v}^{t} = (-81, 137, -34, 22) \\
(\mathbb{A} + \Delta\mathbb{A})\boldsymbol{y} = (\boldsymbol{b} + \boldsymbol{\Delta b}) \quad &\Longleftrightarrow \quad \boldsymbol{y}^{t} = \left(-\frac{18283543}{461600}, \frac{31504261}{461600}, -\frac{3741501}{230800}, \frac{5235241}{461600}\right) \\
&\qquad\qquad\quad \approx (-39.61, 68.25, -16.21, 11.34)
\end{aligned}$$

Soient $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ inversible et $\boldsymbol{b} \in \mathbb{K}^n$.

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$$

De petites erreurs sur les données engendrent-elles de petites erreurs sur la solution?

Non, pas forcément!

Le système linéaire prédédent est **mal conditionné**.
On dit qu'un système linéaire est **bien conditionné** ou qu'il a un **bon conditionnement** si de petites perturbations des données n'entrainent qu'une variation *raisonnable* de la solution.

Est-il possible de "mesurer" le **conditionnement** d'une matrice?

### Definition 1.1

Soit $\|.\|$ une norme matricielle subordonnée, le conditionnement d'une matrice régulière $\mathbb{A}$, associé à cette norme, est le nombre

$$\mathrm{cond}(\mathbb{A}) = \|\mathbb{A}\| \left\|\mathbb{A}^{-1}\right\|.$$

Nous noterons $\mathrm{cond}_p(\mathbb{A}) = \|\mathbb{A}\|_p \left\|\mathbb{A}^{-1}\right\|_p$.

**Proposition:**

Soit $\mathbb{A}$ une matrice régulière. On a les propriétés suivantes

1. $\forall \alpha \in \mathbb{K}^*, \operatorname{cond}(\alpha\mathbb{A}) = \operatorname{cond}(\mathbb{A})$.
2. $\operatorname{cond}_p(\mathbb{A}) \geqslant 1, \forall p \in [\![1, +\infty]\!]$.
3. $\operatorname{cond}_2(\mathbb{A}) = 1$ si et seulement si $\mathbb{A} = \alpha\mathbb{Q}$ avec $\alpha \in \mathbb{K}^*$ et $\mathbb{Q}$ matrice unitaire

**Théorème 2:**

Soit $\mathbb{A}$ une matrice inversible. Soient $\boldsymbol{x}$ et $\boldsymbol{x}+\boldsymbol{\Delta x}$ les solutions respectives de

$$\mathbb{A}\boldsymbol{x}=\boldsymbol{b} \quad \text{et} \quad \mathbb{A}\left(\boldsymbol{x}+\boldsymbol{\Delta x}\right)=\boldsymbol{b}+\boldsymbol{\Delta b}.$$

Supposons $\boldsymbol{b}\neq\mathbf{0}$, alors l'inégalité

$$\frac{\|\boldsymbol{\Delta x}\|}{\|\boldsymbol{x}\|}\leqslant \mathrm{cond}(\mathbb{A})\frac{\|\boldsymbol{\Delta b}\|}{\|\boldsymbol{b}\|}$$

est satisfaite, et c'est la meilleure possible : pour une matrice $\mathbb{A}$ donnée, on peut trouver des vecteurs $\boldsymbol{b}\neq\mathbf{0}$ et $\boldsymbol{\Delta b}\neq\mathbf{0}$ tels qu'elle devienne une égalité.

**Théorème:**

Soient $\mathbb{A}$ et $\mathbb{A} + \Delta\mathbb{A}$ deux matrices inversibles. Soient $\boldsymbol{x}$ et $\boldsymbol{x} + \boldsymbol{\Delta x}$ les solutions respectives de

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \text{ et } (\mathbb{A} + \Delta\mathbb{A})(\boldsymbol{x} + \boldsymbol{\Delta x}) = \boldsymbol{b}.$$

Supposons $\boldsymbol{b} \neq \mathbf{0}$, alors on a

$$\frac{\|\boldsymbol{\Delta x}\|}{\|\boldsymbol{x} + \boldsymbol{\Delta x}\|} \leqslant \operatorname{cond}(\mathbb{A}) \frac{\|\Delta\mathbb{A}\|}{\|\mathbb{A}\|}.$$

Remarque 2.1

Une matrice est donc **bien conditionnée** si son conditionnement est proche de 1.

# Plan

1 Conditionnement
2 Méthodes directes
- Matrices particulières
  - Matrices diagonales
  - Matrices triangulaires inférieures
  - Matrices triangulaires supérieures
- Exercices et résultats préliminaires
- Méthode de Gauss-Jordan
  - Ecriture algébrique
- Factorisation $\mathbb{LU}$
  - Résultats théoriques
  - Utilisation pratique
- Factorisation $\mathbb{LDL}^*$
- Factorisation de Cholesky
  - Résultats théoriques
  - Résolution d'un système linéaire
  - Algorithme : Factorisation positive de Cholesky
- Factorisation $\mathbb{QR}$
  - La tranformation de Householder

# Système diagonal

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ diagonale inversible et $\boldsymbol{b} \in \mathbb{K}^n$.

$$x_i = b_i / A_{i,i}, \quad \forall i \in [\![1, n]\!]. \tag{1}$$

---

**Algorithme 1** Fonction RSLMatDiag permettant de résoudre le système linéaire à matrice diagonale inversible

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}.$$

---

**Données :** $\mathbb{A}$ : matrice diagonale de $\mathcal{M}_n(\mathbb{R})$ inversible.
$\boldsymbol{b}$ : vecteur de $\mathbb{R}^n$.
**Résultat :** $\boldsymbol{x}$ : vecteur de $\mathbb{R}^n$.

1: **Fonction** $\boldsymbol{x} \leftarrow$ RSLMatDiag ( $\mathbb{A}, \boldsymbol{b}$ )
2: **Pour** $i \leftarrow 1$ **à** $n$ **faire**
3: $x(i) \leftarrow b(i)/A(i,i)$
4: **Fin Pour**
5: **Fin Fonction**

---

# Système triangulaire inférieur

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ triangulaire inférieure inversible ($A_{i,j} = 0$ si $i < j$)

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \quad \Longleftrightarrow \quad \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ A_{n,1} & \dots & \dots & A_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

$\mathbb{A}$ inversible $\Longleftrightarrow$

# Système triangulaire inférieur

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ triangulaire inférieure inversible ($A_{i,j} = 0$ si $i < j$)

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \quad \Longleftrightarrow \quad \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ A_{n,1} & \dots & \dots & A_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

$$\mathbb{A} \text{ inversible} \Longleftrightarrow A_{i,i} \neq 0, \forall i \in [\![1, n]\!]$$

# Système triangulaire inférieur

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ triangulaire inférieure inversible ($A_{i,j} = 0$ si $i < j$)

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \iff \begin{pmatrix} A_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ A_{n,1} & \dots & \dots & A_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

$$\mathbb{A} \text{ inversible} \iff A_{i,i} \neq 0, \forall i \in [\![1, n]\!]$$

Soit $i \in [\![1, n]\!]$, $\quad (\mathbb{A}\boldsymbol{x})_i = b_i, \iff \displaystyle\sum_{j=1}^{n} A_{i,j} x_j = b_i.$

$$b_i = \sum_{j=1}^{i-1} A_{i,j} x_j + A_{i,i} x_i + \sum_{j=i+1}^{n} \underbrace{A_{i,j}}_{=0} x_j = \sum_{j=1}^{i-1} A_{i,j} x_j + A_{i,i} x_i$$

$$x_i = \frac{1}{A_{i,i}} \left( b_i - \sum_{j=1}^{i-1} A_{i,j} x_j \right), \ \forall i \in [\![1, n]\!]. \tag{2}$$

$$x_i = \frac{1}{A_{i,i}}\left(b_i - \sum_{j=1}^{i-1} A_{i,j}x_j\right), \ \forall i \in [\![1, n]\!].$$

**Algorithme 2** $\boxed{\mathcal{R}_0}$

1: Résoudre $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$ en calculant successivement $x_1, x_2, \ldots, x_n$.

**Algorithme 2** $\boxed{\mathcal{R}_1}$

1: **Pour** $i \leftarrow 1$ **à** $n$ **faire**

2: $\quad x_i \leftarrow \frac{1}{A_{i,i}}\left(b_i - \sum_{j=1}^{i-1} A_{i,j}x_j\right)$

3: **Fin Pour**

$$x_i = \frac{1}{A_{i,i}} \left( b_i - \sum_{j=1}^{i-1} A_{i,j} x_j \right), \ \forall i \in [\![1, n]\!].$$

**Algorithme 2** $\mathcal{R}_1$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$$x_i \leftarrow \frac{1}{A_{i,i}} \left( b_i - \sum_{j=1}^{i-1} A_{i,j} x_j \right)$$

**Fin Pour**

**Algorithme 2** $\mathcal{R}_2$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$$\left\{ \begin{array}{l} S \leftarrow \displaystyle\sum_{j=1}^{i-1} A_{i,j} x_j \\ x_i \leftarrow (b_i - S)/A_{i,i} \end{array} \right.$$

**Fin Pour**

$$x_i = \frac{1}{A_{i,i}} \left( b_i - \sum_{j=1}^{i-1} A_{i,j} x_j \right), \ \forall i \in [\![1, n]\!].$$

**Algorithme 2** $\mathcal{R}_3$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
    $\boxed{S \leftarrow \sum_{j=1}^{i-1} A_{i,j} x_j}$
    $x_i \leftarrow (b_i - S)/A_{i,i}$
**Fin Pour**

**Algorithme 2** $\mathcal{R}_4$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
    $S \leftarrow 0$
    **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
        $S \leftarrow S + A(i,j) * x(j)$
    **Fin Pour**
    $x_i \leftarrow (b_i - S)/A_{i,i}$
**Fin Pour**

$$x_i = \frac{1}{A_{i,i}}\left(b_i - \sum_{j=1}^{i-1} A_{i,j}x_j\right),\ \forall i \in [\![1, n]\!].$$

**Algorithme 2** Fonction RSLTriInf permettant de résoudre le système linéaire triangulaire inférieur inversible

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}.$$

**Données :** $\mathbb{A}$ : matrice triangulaire de $\mathcal{M}_n(\mathbb{K})$ inférieure inversible.
$\boldsymbol{b}$ : vecteur de $\mathbb{K}^n$.
**Résultat :** $\boldsymbol{x}$ : vecteur de $\mathbb{K}^n$.

**Fonction** $\boldsymbol{x} \leftarrow$ RSLTriInf ( $\mathbb{A}, \boldsymbol{b}$ )
  **Pour** $i \leftarrow 1$ **à** $n$ **faire**
    $S \leftarrow 0$
    **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
      $S \leftarrow S + A(i,j) * x(j)$
    **Fin Pour**
    $x(i) \leftarrow (b(i) - S)/A(i,i)$
  **Fin Pour**
**Fin Fonction**

# Système triangulaire supérieur

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ triangulaire supérieure inversible ($A_{i,j} = 0$ si $i > j$)

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \quad \Longleftrightarrow \quad \begin{pmatrix} A_{1,1} & \dots & \dots & A_{1,n} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & A_{n,n} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \vdots \\ \vdots \\ b_n \end{pmatrix}$$

**Exercice**

Ecrire la fonction RSLTriSup permettant de résoudre le système triangulaire supérieure $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$.

# Plan


1. Conditionnement
2. Méthodes directes
   - Matrices particulières
     - Matrices diagonales
     - Matrices triangulaires inférieures
     - Matrices triangulaires supérieures
   - Exercices et résultats préliminaires
   - Méthode de Gauss-Jordan
     - Ecriture algébrique
   - Factorisation $\mathbb{LU}$
     - Résultats théoriques
     - Utilisation pratique
   - Factorisation $\mathbb{LDL}^*$
   - Factorisation de Cholesky
     - Résultats théoriques
     - Résolution d'un système linéaire
     - Algorithme : Factorisation positive de Cholesky
   - Factorisation $\mathbb{QR}$
     - La tranformation de Householder

**Lemme 3.1:**

Soit $(i,j) \in [\![1,n]\!]^2$. On note $\mathbb{P}_n^{[i,j]} \in \mathcal{M}_n(\mathbb{R})$ la matrice identitée dont on a permuté les lignes $i$ et $j$. Alors la matrice $\mathbb{P}_n^{[i,j]}$ est symétrique et orthogonale. Pour toute matrice $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$,

1. la matrice $\mathbb{P}_n^{[i,j]}\mathbb{A}$ est matrice $\mathbb{A}$ dont on a permuté les **lignes** $i$ et $j$,
2. la matrice $\mathbb{A}\mathbb{P}_n^{[i,j]}$ est matrice $\mathbb{A}$ dont on a permuté les **colonnes** $i$ et $j$,

**Lemme 3.2:**

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ avec $A_{1,1} \neq 0$. Il existe une matrice $\mathbb{E} \in \mathcal{M}_n(\mathbb{C})$ triangulaire inférieure à diagonale unité telle que

$$\mathbb{E}\mathbb{A}\boldsymbol{e}_1 = A_{1,1}\boldsymbol{e}_1 \tag{3}$$

où $\boldsymbol{e}_1$ est le premier vecteur de la base canonique de $\mathbb{C}^n$.

### Théorème 4: Décomposition de Schur ★★★★☆

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$. Il existe une matrice unitaire $\mathbb{U}$ et une matrice triangulaire supérieure $\mathbb{T}$ telles que

$$\mathbb{A} = \mathbb{U}\mathbb{T}\mathbb{U}^* \tag{4}$$

# Plan


1. Conditionnement
2. Méthodes directes
   - Matrices particulières
     - Matrices diagonales
     - Matrices triangulaires inférieures
     - Matrices triangulaires supérieures
   - Exercices et résultats préliminaires
   - Méthode de Gauss-Jordan
     - Ecriture algébrique
   - Factorisation $\mathbb{L}\mathbb{U}$
     - Résultats théoriques
     - Utilisation pratique
   - Factorisation $\mathbb{L}\mathbb{D}\mathbb{L}^*$
   - Factorisation de Cholesky
     - Résultats théoriques
     - Résolution d'un système linéaire
     - Algorithme : Factorisation positive de Cholesky
   - Factorisation $\mathbb{Q}\mathbb{R}$
     - La tranformation de Householder

# Algorithme de Gauss-Jordan

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \iff \mathbb{U}\boldsymbol{x} = \boldsymbol{f}$$

où $\mathbb{U}$ matrice triangulaire supérieure.
Opérations élémentaires sur les matrices :

- $\mathcal{L}_i \leftrightarrow \mathcal{L}_j$ permutation lignes $i$ et $j$
- $\mathcal{L}_i \leftarrow \mathcal{L}_i + \alpha\mathcal{L}_j$ combinaison linéaire

A l'aide d'opérations élémentaires, on va transformer successivement en $n-1$ étapes le système. A l'étape $j$, on va *s'arranger* pour annuler les termes sous-diagonaux de la colonne $j$ de la matrice sans modifier les $j-1$ premières colonnes.

$$
\begin{array}{c}
\overset{j-1}{\longleftrightarrow} \\
\left(\begin{array}{cccc|ccc}
\bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \bullet \\
0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
\vdots & \ddots & \ddots & \bullet & \vdots & & \vdots \\
0 & \dots & 0 & \bullet & \bullet & \cdots & \bullet \\
\hline
0 & \cdots & \cdots & 0 & \color{red}{\bullet} & \cdots & \bullet \\
\vdots & & & \vdots & \color{red}{\vdots} & & \vdots \\
0 & \dots & \dots & 0 & \color{red}{\bullet} & \cdots & \bullet
\end{array}\right)
\end{array}
\boldsymbol{x} =
\left(\begin{array}{c}
\bullet \\ \vdots \\ \bullet \\ \hline \bullet \\ \vdots \\ \bullet
\end{array}\right)
$$

**Etape $j$** $\qquad \Longleftrightarrow$

$$
\begin{array}{c}
\overset{j}{\longleftrightarrow} \\
\left(\begin{array}{cccc|ccc}
\bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \bullet \\
0 & \ddots & \ddots & \vdots & \vdots & & \vdots \\
\vdots & \ddots & \ddots & \bullet & \vdots & & \vdots \\
0 & \dots & 0 & \bullet & \bullet & \cdots & \bullet \\
\hline
0 & \cdots & \cdots & \color{red}{0} & \bullet & \cdots & \bullet \\
\vdots & & & \color{red}{\vdots} & \vdots & & \vdots \\
0 & \dots & \dots & \color{red}{0} & \bullet & \cdots & \bullet
\end{array}\right)
\end{array}
\boldsymbol{x} =
\left(\begin{array}{c}
\bullet \\ \vdots \\ \bullet \\ \hline \bullet \\ \vdots \\ \bullet
\end{array}\right)
$$

**Algorithme 3** Algorithme de Gauss-Jordan formel pour la résolution de $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$

**Pour** $j \leftarrow 1$ **à** $n-1$ **faire**
    Rechercher l'indice $k$ de la ligne du pivot (sur la colonne $j$, $k \in [\![j, n]\!]$)
    Permuter les lignes $j$ $(\mathcal{L}_j)$ et $k$ $(\mathcal{L}_k)$ du système si besoin.
    **Pour** $i \leftarrow j+1$ **à** $n$ **faire**
        Eliminer en effectuant $\mathcal{L}_i \leftarrow \mathcal{L}_i - \frac{A_{i,j}}{A_{j,j}}\mathcal{L}_j$
    **Fin Pour**
**Fin Pour**
Résoudre le système triangulaire supérieur par la méthode de la remontée.

**Algorithme 4** Algorithme de Gauss-Jordan avec fonctions pour la résolution de $\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$

**Données :** $\mathbb{A}$ : matrice de $\mathcal{M}_n(\mathbb{K})$ inversible.
$\boldsymbol{b}$ : vecteur de $\mathbb{K}^n$.

**Résultat :** $\boldsymbol{x}$ : vecteur de $\mathbb{R}^n$.

**Fonction** $\boldsymbol{x} \leftarrow$ RSLGauss $(\mathbb{A}, \boldsymbol{b})$
  **Pour** $j \leftarrow 1$ **à** $n-1$ **faire**
    $k \leftarrow \boxed{\text{ChercheIndPivot}}(\mathbb{A}, j)$ ▷ à écrire
    $[\mathbb{A}, \boldsymbol{b}] \leftarrow \boxed{\text{PermLignesSys}}(\mathbb{A}, \boldsymbol{b}, j, k)$ ▷ à écrire
    **Pour** $i \leftarrow j+1$ **à** $n$ **faire**
      $[\mathbb{A}, \boldsymbol{b}] \leftarrow \boxed{\text{CombLignesSys}}(\mathbb{A}, \boldsymbol{b}, j, i, -A(i,j)/A(j,j))$ ▷ à écrire
    **Fin Pour**
  **Fin Pour**
  $\boldsymbol{x} \leftarrow$ RSLTriSup$(\mathbb{A}, \boldsymbol{b})$ ▷ déjà écrite
**Fin Fonction**

**Algorithme 5** Recherche d'un pivot pour l'algorithme de Gauss-Jordan.

| | | | |
|---|---|---|---|
| **Données :** | $\mathbb{A}$ | : | matrice de $\mathcal{M}_n(\mathbb{K})$. |
| | $j$ | : | entier, $1 \leqslant j \leqslant n$. |
| **Résultat :** | $k$ | : | entier, indice ligne pivot |

**Fonction** $k \leftarrow$ ChercheIndPivot $(\mathbb{A}, j)$
    $k \leftarrow j,\ \mathrm{pivot} \leftarrow |\mathbb{A}(j,j)|$
    **Pour** $i \leftarrow j+1$ **à** $n$ **faire**
        **Si** $|\mathbb{A}(i,j)| > \mathrm{pivot}$ **alors**
            $k \leftarrow i$
            $\mathrm{pivot} \leftarrow |\mathbb{A}(i,j)|$
        **Fin Si**
    **Fin Pour**
**Fin Fonction**

**Algorithme 6** Permutte deux lignes d'une matrice et d'un vecteur.

| | | | |
|---|---|---|---|
| **Données :** | $\mathbb{A}$ | : | matrice de $\mathcal{M}_n(\mathbb{K})$. |
| | $\boldsymbol{b}$ | : | vecteur de $\mathbb{K}^n$. |
| | $j, k$ | : | entiers, $1 \leqslant j, k \leqslant n$. |
| **Résultat :** | $\mathbb{A}$ et $\boldsymbol{b}$ modifiés. | | |

**Fonction** $[\mathbb{A}, \boldsymbol{b}] \leftarrow$ PermLignesSys $(\mathbb{A}, \boldsymbol{b}, j, k)$
    **Pour** $l \leftarrow 1$ **à** $n$ **faire**
        $t \leftarrow \mathbb{A}(j,l)$
        $\mathbb{A}(j,l) \leftarrow \mathbb{A}(k,l)$
        $\mathbb{A}(k,l) \leftarrow t$
    **Fin Pour**
    $t \leftarrow \boldsymbol{b}(j),\ \boldsymbol{b}(j) \leftarrow \boldsymbol{b}(k),\ \boldsymbol{b}(k) \leftarrow t$
**Fin Fonction**

**Algorithme 7** Combinaison linéaire $\mathcal{L}_i \leftarrow \mathcal{L}_i + \alpha \mathcal{L}_j$ appliqué à une matrice et à un vecteur.

| | | | |
|---|---|---|---|
| **Données :** | $\mathbb{A}$ | : | matrice de $\mathcal{M}_n(\mathbb{K})$. |
| | $\boldsymbol{b}$ | : | vecteur de $\mathbb{K}^n$. |
| | $j, i$ | : | entiers, $1 \leqslant j, i \leqslant n$. |
| | alpha | : | scalaire de $\mathbb{K}$ |
| **Résultat :** | $\mathbb{A}$ et $\boldsymbol{b}$ modifiés. | | |

**Fonction** $[\mathbb{A}, \boldsymbol{b}] \leftarrow$ CombLignesSys $(\mathbb{A}, \boldsymbol{b}, j, i, \alpha)$
    **Pour** $k \leftarrow 1$ **à** $n$ **faire**
        $\mathbb{A}(i,k) \leftarrow \mathbb{A}(i,k) + \alpha * \mathbb{A}(j,k)$
    **Fin Pour**
    $\boldsymbol{b}(i) \leftarrow \boldsymbol{b}(i) + \alpha \boldsymbol{b}(j)$
**Fin Fonction**

## Exercice 4.1: Méthode de Gauss, écriture algébrique

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ inversible.

### Q. 1

*Montrer qu'il existe une matrice* $\mathbb{G} \in \mathcal{M}_n(\mathbb{C})$ *telle que* $|\det(\mathbb{G})| = 1$ *et* $\mathbb{G}\mathbb{A}\boldsymbol{e}_1 = \alpha \boldsymbol{e}_1$ *avec* $\alpha \neq 0$ *et* $\boldsymbol{e}_1$ *premier vecteur de la base canonique de* $C^n$.

### Q. 2

1. *Montrer par récurrence sur l'ordre des matrices que pour toute matrice* $\mathbb{A}_n \in \mathcal{M}_n(\mathbb{C})$ *inversible, il existe une matrice* $\mathbb{S}_n \in \mathcal{M}_n(\mathbb{C})$ *telle que* $|\det \mathbb{S}_n| = 1$ *et* $\mathbb{S}_n\mathbb{A}_n = \mathbb{U}_n$ *avec* $\mathbb{U}_n$ *matrice triangulaire supérieure inversible.*
2. *Soit* $\boldsymbol{b} \in \mathbb{C}^n$. *En supposant connue la décompostion précédente* $\mathbb{S}_n\mathbb{A}_n = \mathbb{U}_n$, *expliquer comment résoudre le système* $\mathbb{A}_n\boldsymbol{x} = \boldsymbol{b}$.

### Q. 3

*Que peut-on dire si* $\mathbb{A}$ *est non inversible?*

**Indication :** utiliser les Lemmes 3.1 et 3.2.

On a donc démontré le théorème suivant

**Théorème 5**

Soit $\mathbb{A}$ une matrice carrée, inversible ou non. Il existe (au moins) une matrice inversible $\mathbb{G}$ telle que $\mathbb{G}\mathbb{A}$ soit triangulaire supérieure.

# Plan


1. Conditionnement
2. Méthodes directes
   - Matrices particulières
     - Matrices diagonales
     - Matrices triangulaires inférieures
     - Matrices triangulaires supérieures
   - Exercices et résultats préliminaires
   - Méthode de Gauss-Jordan
     - Ecriture algébrique
   - Factorisation $\mathbb{LU}$
     - Résultats théoriques
     - Utilisation pratique
   - Factorisation $\mathbb{LDL}^*$
   - Factorisation de Cholesky
     - Résultats théoriques
     - Résolution d'un système linéaire
     - Algorithme : Factorisation positive de Cholesky
   - Factorisation $\mathbb{QR}$
     - La tranformation de Householder

## Exercice: Factorisation $\mathbb{L}\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ une matrice dont les sous-matrices principales d'ordre $i$, notées $\Delta_i$, $i \in [\![1, n]\!]$ (voir Definition B.48, page 188) sont inversibles. Montrer qu'il existe des matrices $\mathbb{E}^{[k]} \in \mathcal{M}_n(\mathbb{C})$, $k \in [\![1, n-1]\!]$, triangulaires inférieures à diagonale unité telles que la matrice $\mathbb{U}$ définie par

$$\mathbb{U} = \mathbb{E}^{[n-1]} \dots \mathbb{E}^{[1]} \mathbb{A}$$

soit triangulaire supérieure avec $U_{i,i} = \det \Delta_i / (U_{1,1} \times \dots \times U_{i-1,i-1})$, $\forall i \in [\![1, n]\!]$.

## Théorème 6: Factorisation $\mathbb{L}\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ une matrice dont les sous-matrices principales sont inversibles alors il existe une unique matrice $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$ triangulaire inférieure (*lower triangular* en anglais) à diagonale unité et une unique matrice $\mathbb{U} \in \mathcal{M}_n(\mathbb{C})$ triangulaire supérieure (*upper triangular* en anglais) inversible telles ques

$$\mathbb{A} = \mathbb{L}\mathbb{U}.$$

**preuve :**

- **Existence :** exercice précédant $\mathbb{U} = \mathbb{E}^{[n-1]} \ldots \mathbb{E}^{[1]}\mathbb{A}$

$$\mathbb{L} = \left(\mathbb{E}^{[n-1]} \ldots \mathbb{E}^{[1]}\right)^{-1}$$

- **Unicité :** $\mathbb{A} = \mathbb{L}_1\mathbb{U}_1 = \mathbb{L}_2\mathbb{U}_2 \ldots$

**Corollaire 6.1:**

Si $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ est une matrice hermitienne définie positive alors elle admet une unique factorisation $\mathbb{L}\mathbb{U}$.

**preuve :** $\mathbb{A}$ hermitienne définie positive alors toutes ses sous-matrices principales sont définies positives et donc inversibles.

### Remarque 6.2

Si la matrice $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ est inversible mais que ses sous-matrices principales ne sont pas toutes inversibles, il est possible par des permutations préalables de lignes de la matrice de se ramener à une matrice telle que ses sous-matrices principales soient inversibles.

### Théorème 7: Factorisation $\mathbb{L}\mathbb{U}$ avec permutations

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ une matrice inversible. Il existe une matrice $\mathbb{P}$, produit de matrices de permutation, une matrice $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$ triangulaire inférieure à diagonale unité et une matrice $\mathbb{U} \in \mathcal{M}_n(\mathbb{C})$ triangulaire supérieure telles ques

$$\mathbb{P}\mathbb{A} = \mathbb{L}\mathbb{U}. \tag{5}$$

# Utilisation pratique de la factorisation $\mathbb{L}\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{L}\mathbb{U}$

Trouver $\boldsymbol{x} \in \mathbb{K}^n$ tel que

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \iff \mathbb{L}\mathbb{U}\boldsymbol{x} = \boldsymbol{b} \tag{6}$$

est équivalent à

# Utilisation pratique de la factorisation $\mathbb{L}\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{L}\mathbb{U}$

Trouver $\boldsymbol{x} \in \mathbb{K}^n$ tel que

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \iff \mathbb{L}\mathbb{U}\boldsymbol{x} = \boldsymbol{b} \tag{6}$$

est équivalent à

Trouver $\boldsymbol{x} \in \mathbb{K}^n$ solution de

$$\mathbb{U}\boldsymbol{x} = \boldsymbol{y} \tag{7}$$

avec $\boldsymbol{y} \in \mathbb{K}^n$ solution de

$$\mathbb{L}\boldsymbol{y} = \boldsymbol{b}. \tag{8}$$

# Algorithme de résolution de systèmes linéaire par $\mathbb{L}\mathbb{U}$

**Algorithme 8** Fonction RSLFactLU permettant de résoudre, par une factorisation $\mathbb{L}\mathbb{U}$, le système linéaire

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$$

où $\mathbb{A}$ une matrice de $\mathcal{M}_n(\mathbb{R})$ définie positive et $\boldsymbol{b} \in \mathbb{R}^n$.

**Données :** $\mathbb{A}$ : matrice de $\mathcal{M}_n(\mathbb{R})$ dont les sous-matrices principales sont inversibles définie positive,
$\boldsymbol{b}$ : vecteur de $\mathbb{R}^n$.

**Résultat :** $\boldsymbol{x}$ : vecteur de $\mathbb{R}^n$.

1: **Fonction** $\boldsymbol{x} \leftarrow$ RSLFactLU $(\mathbb{A}, \boldsymbol{b})$
2: $[\mathbb{L}, \mathbb{U}] \leftarrow$ FactLU$(\mathbb{A})$ ▷ Factorisation $\mathbb{L}\mathbb{U}$
3: $\boldsymbol{y} \leftarrow$ RSLTriInf$(\mathbb{L}, \boldsymbol{b})$ ▷ Résolution du système $\mathbb{L}\boldsymbol{y} = \boldsymbol{b}$
4: $\boldsymbol{x} \leftarrow$ RSLTriSup$(\mathbb{U}, \boldsymbol{y})$ ▷ Résolution du système $\mathbb{U}\boldsymbol{x} = \boldsymbol{y}$
5: **Fin Fonction**

Il nous faut donc écrire la fonction FactLU

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{L}\mathbb{U}$.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ L_{2,1} & 1 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n,1} & L_{n,2} & \dots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & \dots & U_{1,n} \\ 0 & U_{2,2} & \dots & \dots & U_{2,n} \\ 0 & 0 & U_{3,3} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & U_{n,n} \end{pmatrix}$$

On connait $\mathbb{A}$, on cherche $\mathbb{L}$ et $\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{LU}$.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ L_{2,1} & 1 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n,1} & L_{n,2} & \dots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & \dots & U_{1,n} \\ 0 & U_{2,2} & \dots & \dots & U_{2,n} \\ 0 & 0 & U_{3,3} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & U_{n,n} \end{pmatrix}$$

- **Etape** 1 :
  - On connait la **première ligne** de $\mathbb{L} \Longrightarrow$ on peut calculer la **première ligne** de $\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{L}\mathbb{U}$.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ L_{2,1} & 1 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n,1} & L_{n,2} & \dots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & \dots & U_{1,n} \\ 0 & U_{2,2} & \dots & \dots & U_{2,n} \\ 0 & 0 & U_{3,3} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & U_{n,n} \end{pmatrix}$$

- **Etape** 1 :
  - On connait la **première ligne** de $\mathbb{L}$ $\Longrightarrow$ on peut calculer la **première ligne** de $\mathbb{U}$
  - On connait la **première colonne** de $\mathbb{U}$ $\Longrightarrow$ on peut calculer la **première colonne** de $\mathbb{L}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{LU}$.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ L_{2,1} & 1 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n,1} & L_{n,2} & \dots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & \dots & U_{1,n} \\ 0 & U_{2,2} & \dots & \dots & U_{2,n} \\ 0 & 0 & U_{3,3} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & U_{n,n} \end{pmatrix}$$

- **Etape** 1 :
  - On connait la **première ligne** de $\mathbb{L}$ $\Longrightarrow$ on peut calculer la **première ligne** de $\mathbb{U}$
  - On connait la **première colonne** de $\mathbb{U}$ $\Longrightarrow$ on peut calculer la **première colonne** de $\mathbb{L}$
- **Etape** 2 :
  - On connait la **deuxième ligne** de $\mathbb{L}$ $\Longrightarrow$ on peut calculer la **deuxième ligne** de $\mathbb{U}$ car on connait la première ligne de $\mathbb{U}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{L}\mathbb{U}$.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ L_{2,1} & 1 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n,1} & L_{n,2} & \dots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & \dots & U_{1,n} \\ 0 & U_{2,2} & \dots & \dots & U_{2,n} \\ 0 & 0 & U_{3,3} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & U_{n,n} \end{pmatrix}$$

- **Etape** 1 :
  - On connait la **première ligne** de $\mathbb{L} \Longrightarrow$ on peut calculer la **première ligne** de $\mathbb{U}$
  - On connait la **première colonne** de $\mathbb{U} \Longrightarrow$ on peut calculer la **première colonne** de $\mathbb{L}$
- **Etape** 2 :
  - On connait la **deuxième ligne** de $\mathbb{L} \Longrightarrow$ on peut calculer la **deuxième ligne** de $\mathbb{U}$ car on connait la première ligne de $\mathbb{U}$
  - On connait la **deuxième colonne** de $\mathbb{U} \Longrightarrow$ on peut calculer la **deuxième colonne** de $\mathbb{L}$ car on connait la première colonne de $\mathbb{L}$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{K})$ admettant une factorisation $\mathbb{L}\mathbb{U}$.

$$\begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,n} \\ A_{2,1} & A_{2,2} & \dots & A_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ A_{n,1} & A_{n,2} & \dots & A_{n,n} \end{pmatrix} = \begin{pmatrix} 1 & 0 & \dots & \dots & 0 \\ L_{2,1} & 1 & 0 & \dots & 0 \\ L_{3,1} & L_{3,2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ L_{n,1} & L_{n,2} & \dots & L_{n,n-1} & 1 \end{pmatrix} \begin{pmatrix} U_{1,1} & U_{1,2} & \dots & \dots & U_{1,n} \\ 0 & U_{2,2} & \dots & \dots & U_{2,n} \\ 0 & 0 & U_{3,3} & & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & U_{n,n} \end{pmatrix}$$

- **Etape** 1 :
  - On connait la **première ligne** de $\mathbb{L} \Longrightarrow$ on peut calculer la **première ligne** de $\mathbb{U}$
  - On connait la **première colonne** de $\mathbb{U} \Longrightarrow$ on peut calculer la **première colonne** de $\mathbb{L}$
- **Etape** 2 :
  - On connait la **deuxième ligne** de $\mathbb{L} \Longrightarrow$ on peut calculer la **deuxième ligne** de $\mathbb{U}$ car on connait la première ligne de $\mathbb{U}$
  - On connait la **deuxième colonne** de $\mathbb{U} \Longrightarrow$ on peut calculer la **deuxième colonne** de $\mathbb{L}$ car on connait la première colonne de $\mathbb{L}$
- ...

$$\mathbb{A} = \overset{\mathbb{L}}{\begin{pmatrix} \bullet & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \bullet & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \bullet & \cdots & \bullet & \bullet & 0 & \cdots & \cdots & 0 \\ \bullet & \cdots & \cdots & \bullet & L_{i,i} & 0 & \cdots & 0 \\ \vdots & & & \vdots & \bullet & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ \bullet & \cdots & \cdots & \bullet & \bullet & \cdots & \bullet & L_{n,n} \end{pmatrix}} \underset{\mathbb{U}}{\begin{pmatrix} \bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ 0 & \cdots & 0 & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \cdots & \cdots & 0 & U_{i,i} & \bullet & \cdots & \bullet \\ \vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & \bullet \\ 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & U_{n,n} \end{pmatrix}}$$

$i-1$ Connus (colonnes de $\mathbb{L}$) — Connus, $i-1$ (lignes de $\mathbb{U}$)

Par récurrence, on suppose connues les $i-1$ premières colonnes de $\mathbb{L}$ et les $i-1$ premières lignes de $\mathbb{U}$.

Peut-on calculer la colonne $i$ de $\mathbb{L}$ et la ligne $i$ de $\mathbb{U}$?

$\mathbb{L}$

$i-1$

Connus

$$\mathbb{A} = \left(\begin{array}{cccc:cccc}
\bullet & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\
\bullet & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\bullet & \cdots & \bullet & \bullet & 0 & \cdots & \cdots & 0 \\
\bullet & \cdots & \cdots & \bullet & L_{i,i} & 0 & \cdots & 0 \\
\vdots & & & \vdots & \bullet & \ddots & \ddots & \vdots \\
\vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\
\bullet & \cdots & \cdots & \bullet & \bullet & \cdots & \bullet & L_{n,n}
\end{array}\right)
\left(\begin{array}{cccc:cccc}
\bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \cdots & \bullet \\
0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
0 & \cdots & 0 & \bullet & \bullet & \cdots & \cdots & \bullet \\
\hdashline
0 & \cdots & \cdots & 0 & U_{i,i} & \bullet & \cdots & \bullet \\
\vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & & & \vdots & \vdots & \ddots & \ddots & \bullet \\
0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & U_{n,n}
\end{array}\right)$$

$i-1$

Connus

$L_{i,i} = 1$

$\mathbb{U}$

$\Rightarrow$ ligne $i$ de $\mathbb{L}$ connue

$i-1$ $\mathbb{L}$

Connus

$$\mathbb{A} = \begin{pmatrix} \bullet & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \bullet & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \bullet & \cdots & \bullet & \bullet & 0 & \cdots & \cdots & 0 \\ \bullet & \cdots & \cdots & \bullet & L_{i,i} & 0 & \cdots & 0 \\ \vdots & & & \vdots & \bullet & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ \bullet & \cdots & \cdots & \bullet & \bullet & \cdots & \bullet & L_{n,n} \end{pmatrix} \begin{pmatrix} \bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ 0 & \cdots & 0 & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \cdots & \cdots & 0 & U_{i,i} & \bullet & \cdots & \bullet \\ \vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & \bullet \\ 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & U_{n,n} \end{pmatrix}$$

$i-1$

Connus

$L_{i,i}=1$

$\Rightarrow$ ligne $i$ de $\mathbb{L}$ connue

$\mathbb{U}$

On peut calculer ligne $i$ de $\mathbb{U}$

On cherche $U_{i,j}$ $\forall j \in [\![i, n]\!]$.

$$A_{i,j} \stackrel{\text{def}}{=} \sum_{k=1}^{n} L_{i,k} U_{k,j} = \sum_{k=1}^{i-1} \overbrace{L_{i,k} U_{k,j}}^{\text{connus}} + \overbrace{L_{i,i}}^{=1} U_{i,j} + \sum_{k=1}^{i+1} \overbrace{L_{i,k}}^{=0} U_{k,j}$$

$$U_{i,j} = A_{i,j} - \sum_{k=1}^{i-1} L_{i,k} U_{k,j}, \quad \forall j \in [\![i, n]\!].$$

$i-1$ $\mathbb{L}$

Connus

$$\mathbb{A} = \begin{pmatrix} \bullet & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\ \bullet & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \bullet & \cdots & \bullet & \bullet & 0 & \cdots & \cdots & 0 \\ \bullet & \cdots & \cdots & \bullet & L_{i,i} & 0 & \cdots & 0 \\ \vdots & & & \vdots & \bullet & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ \bullet & \cdots & \cdots & \bullet & \bullet & \cdots & \bullet & L_{n,n} \end{pmatrix} \begin{pmatrix} \bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ 0 & \cdots & 0 & \bullet & \bullet & \cdots & \cdots & \bullet \\ 0 & \cdots & \cdots & 0 & U_{i,i} & \bullet & \cdots & \bullet \\ \vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & \bullet \\ 0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & U_{n,n} \end{pmatrix} \quad i-1$$

Connus

$\mathbb{U}$

On connait la colonne $i$ de $\mathbb{U}$

$\mathbb{L}$ $\quad$ $i-1$ $\quad$ Connus $\quad$ $i-1$

$$\mathbb{A} = \left(\begin{array}{cccc:cccc}
\bullet & 0 & \cdots & 0 & 0 & \cdots & \cdots & 0 \\
\bullet & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\bullet & \cdots & \bullet & \bullet & 0 & \cdots & \cdots & 0 \\
\bullet & \cdots & \cdots & \bullet & L_{i,i} & 0 & \cdots & 0 \\
\vdots & & & \vdots & \bullet & \ddots & \ddots & \vdots \\
\vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\
\bullet & \cdots & \cdots & \bullet & \bullet & \cdots & \bullet & L_{n,n}
\end{array}\right)
\left(\begin{array}{cccc:cccc}
\bullet & \bullet & \cdots & \bullet & \bullet & \cdots & \cdots & \bullet \\
0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
\vdots & \ddots & \ddots & \vdots & \vdots & & & \vdots \\
0 & \cdots & 0 & \bullet & \bullet & \cdots & \cdots & \bullet \\ \hdashline
0 & \cdots & \cdots & 0 & U_{i,i} & \bullet & \cdots & \bullet \\
\vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\
\vdots & & & \vdots & \vdots & \ddots & \ddots & \bullet \\
0 & \cdots & \cdots & 0 & 0 & \cdots & 0 & U_{n,n}
\end{array}\right)$$

Connus $\quad$ $\mathbb{U}$

On peut calculer la colonne $i$ de $\mathbb{L}$ $\Leftarrow$ On connait la colonne $i$ de $\mathbb{U}$

On cherche $L_{j,i}$ $\forall j \in [\![i+1, n]\!]$, $(L_{i,i} = 1)$

$$A_{j,i} \stackrel{\text{def}}{=} \sum_{k=1}^{n} L_{j,k} U_{k,i} = \sum_{k=1}^{i-1} \overbrace{L_{j,k} U_{k,i}}^{\text{connus}} + L_{j,i} \overbrace{U_{i,i}}^{\text{connu}} + \sum_{k=1}^{i+1} L_{j,k} \overbrace{U_{k,i}}^{=0}$$

$$L_{j,i} = \left( A_{j,i} - \sum_{k=1}^{i-1} L_{j,k} U_{k,i} \right) / U_{i,i}, \quad \forall j \in [\![i+1, n]\!].$$

**Algorithme 9** $\boxed{\mathcal{R}_0}$

1: $\boxed{\text{Calculer les matrices } \mathbb{L} \text{ et } \mathbb{U}}$

**Algorithme 9** $\boxed{\mathcal{R}_1}$

1: **Pour** $i \leftarrow 1$ **à** $n$ **faire**
2: Calculer la ligne $i$ de $\mathbb{U}$.
3: Calculer la colonne $i$ de $\mathbb{L}$.
4: **Fin Pour**

**Algorithme 9** $\mathcal{R}_1$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
    Calculer la ligne i de $\mathbb{U}$.
    Calculer la colonne i de $\mathbb{L}$.
**Fin Pour**

**Algorithme 9** $\mathcal{R}_2$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
    **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
        $U(i,j) \leftarrow 0$
    **Fin Pour**
    **Pour** $j \leftarrow i$ **à** $n$ **faire**
        $$U_{i,j} \leftarrow A_{i,j} - \sum_{k=1}^{i-1} L_{i,k} U_{k,j}$$
    **Fin Pour**
    **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
        $L_{j,i} \leftarrow 0$
    **Fin Pour**
    $L_{i,i} \leftarrow 1$
    **Pour** $j \leftarrow i+1$ **à** $n$ **faire**
        $$L_{j,i} \leftarrow \frac{1}{U_{i,i}} \left( A_{j,i} - \sum_{k=1}^{i-1} L_{j,k} U_{k,i} \right)$$
    **Fin Pour**
**Fin Pour**

**Algorithme 9** $\boxed{\mathcal{R}_2}$

**Pour** $i \leftarrow 1$ à $n$ **faire**
    **Pour** $j \leftarrow 1$ à $i-1$ **faire**
        $U(i,j) \leftarrow 0$
    **Fin Pour**
    **Pour** $j \leftarrow i$ à $n$ **faire**
        $\boxed{U_{i,j} \leftarrow A_{i,j} - \sum_{k=1}^{i-1} L_{i,k}U_{k,j}}$
    **Fin Pour**
    **Pour** $j \leftarrow 1$ à $i-1$ **faire**
        $L_{j,i} \leftarrow 0$
    **Fin Pour**
    $L_{i,i} \leftarrow 1$
    **Pour** $j \leftarrow i+1$ à $n$ **faire**
        $\boxed{L_{j,i} \leftarrow \frac{1}{U_{i,i}}\left(A_{j,i} - \sum_{k=1}^{i-1} L_{j,k}U_{k,i}\right)}$
    **Fin Pour**
**Fin Pour**

**Algorithme 9** $\boxed{\mathcal{R}_3}$

**Pour** $i \leftarrow 1$ à $n$ **faire**
    **Pour** $j \leftarrow 1$ à $i-1$ **faire**
        $U(i,j) \leftarrow 0$
    **Fin Pour**
    **Pour** $j \leftarrow i$ à $n$ **faire**
        $S_1 \leftarrow \sum_{k=1}^{i-1} L_{i,k}U_{k,j}$
        $U_{i,j} \leftarrow A_{i,j} - S_1$
    **Fin Pour**
    **Pour** $j \leftarrow 1$ à $i-1$ **faire**
        $L_{j,i} \leftarrow 0$
    **Fin Pour**
    $L_{i,i} \leftarrow 1$
    **Pour** $j \leftarrow i+1$ à $n$ **faire**
        $S_2 \leftarrow \sum_{k=1}^{i-1} L_{j,k}U_{k,i}$
        $L_{j,i} \leftarrow \frac{1}{U_{i,i}}(A_{j,i} - S_2)$.
    **Fin Pour**
**Fin Pour**

**Algorithme 9** $\boxed{\mathcal{R}_3}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
    $U(i,j) \leftarrow 0$
  **Fin Pour**
  **Pour** $j \leftarrow i$ **à** $n$ **faire**
    $\boxed{S_1 \leftarrow \sum_{k=1}^{i-1} L_{i,k} U_{k,j}}$
    $U_{i,j} \leftarrow A_{i,j} - S_1$
  **Fin Pour**
  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
    $L_{j,i} \leftarrow 0$
  **Fin Pour**
  $L_{i,i} \leftarrow 1$
  **Pour** $j \leftarrow i+1$ **à** $n$ **faire**
    $\boxed{S_2 \leftarrow \sum_{k=1}^{i-1} L_{j,k} U_{k,i}}$
    $L_{j,i} \leftarrow \frac{1}{U_{i,i}} (A_{j,i} - S_2)$.
  **Fin Pour**
**Fin Pour**

**Algorithme 9** $\boxed{\mathcal{R}_4}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
    $U(i,j) \leftarrow 0$
  **Fin Pour**
  **Pour** $j \leftarrow i$ **à** $n$ **faire**
    $S_1 \leftarrow 0$
    **Pour** $k \leftarrow 1$ **à** $i-1$ **faire**
      $S_1 \leftarrow S_1 + L_{i,k} * U_{k,j}$
    **Fin Pour**
    $U_{i,j} \leftarrow A_{i,j} - S_1$
  **Fin Pour**
  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
    $L_{j,i} \leftarrow 0$
  **Fin Pour**
  $L_{i,i} \leftarrow 1$
  **Pour** $j \leftarrow i+1$ **à** $n$ **faire**
    $S_2 \leftarrow 0$
    **Pour** $k \leftarrow 1$ **à** $i-1$ **faire**
      $S_2 \leftarrow S_2 + L_{j,k} * U_{k,i}$
    **Fin Pour**
    $L_{j,i} \leftarrow \frac{1}{U_{i,i}} (A_{j,i} - S_2)$.
  **Fin Pour**
**Fin Pour**

**Algorithme 9** Fonction FactLU permet de calculer les matrices $\mathbb{L}$ et $\mathbb{U}$ dites matrice de factorisation $\mathbb{L}\mathbb{U}$ associée à la matrice $\mathbb{A}$, telle que

$$\mathbb{A} = \mathbb{L}\mathbb{U}$$

---

**Données :** $\mathbb{A}$ : matrice de $\mathcal{M}_n(\mathbb{K})$ dont les sous-matrices principales sont inversibles.

**Résultat :** $\mathbb{L}$ : matrice de $\mathcal{M}_n(\mathbb{K})$ triangulaire inférieure avec $L_{i,i} = 1, \ \forall i \in [\![1, n]\!]$

$\mathbb{U}$ : matrice de $\mathcal{M}_n(\mathbb{K})$ triangulaire supérieure.

**Fonction** $[\mathbb{L}, \mathbb{U}] \leftarrow$ FactLU ( $\mathbb{A}$ )<br>
 $\mathbb{U} \leftarrow \mathbb{O}_n$    ▷ $\mathbb{O}_n$ matrice nulle $n \times n$<br>
 $\mathbb{L} \leftarrow \mathbb{I}_n$    ▷ $\mathbb{I}_n$ matrice identitée $n \times n$<br>
 **Pour** $i \leftarrow 1$ **à** $n$ **faire**<br>
  **Pour** $j \leftarrow i$ **à** $n$ **faire**    ▷ Calcul de la ligne $i$ de $\mathbb{U}$<br>
   $S_1 \leftarrow 0$<br>
   **Pour** $k \leftarrow 1$ **à** $i-1$ **faire**<br>
    $S_1 \leftarrow S_1 + L(i,k) * U(k,j)$<br>
   **Fin Pour**<br>
   $U(i,j) \leftarrow A(i,j) - S_1$<br>
  **Fin Pour**<br>
  **Pour** $j \leftarrow i+1$ **à** $n$ **faire**    ▷ Calcul de la colonne $i$ de $\mathbb{L}$<br>
   $S_2 \leftarrow 0$<br>
   **Pour** $k \leftarrow 1$ **à** $i-1$ **faire**<br>
    $S_2 \leftarrow S_2 + L(j,k) * U(k,i)$<br>
   **Fin Pour**<br>
   $L(j,i) \leftarrow (A_{j,i} - S_2)/U(i,i).$<br>
  **Fin Pour**<br>
 **Fin Pour**<br>
**Fin Fonction**

# Plan


1. Conditionnement
2. Méthodes directes
   - Matrices particulières
     - Matrices diagonales
     - Matrices triangulaires inférieures
     - Matrices triangulaires supérieures
   - Exercices et résultats préliminaires
   - Méthode de Gauss-Jordan
     - Ecriture algébrique
   - Factorisation $\mathbb{LU}$
     - Résultats théoriques
     - Utilisation pratique
   - Factorisation $\mathbb{LDL}^*$
   - Factorisation de Cholesky
     - Résultats théoriques
     - Résolution d'un système linéaire
     - Algorithme : Factorisation positive de Cholesky
   - Factorisation $\mathbb{QR}$
     - La tranformation de Householder

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ **hermitienne inversible** admettant une factorisation $\mathbb{L}\mathbb{U}$. On pose

$$\mathbb{D} = \operatorname{diag} \mathbb{U} \text{ et } \mathbb{R} = \mathbb{D}^{-1}\mathbb{U}.$$

$\mathbb{R}$ est alors triangulaire supérieure à diagonale unité. On a alors

$$\mathbb{A} = \mathbb{L}\mathbb{U} = \mathbb{L}\mathbb{D}\mathbb{D}^{-1}\mathbb{U} = \mathbb{L}\mathbb{D}\mathbb{R}.$$

$$\mathbb{A} \text{ hermitienne } \mathbb{A}^* = \mathbb{A} \implies \mathbb{A} = \mathbb{R}^*(\mathbb{D}^*\mathbb{L}^*) = \mathbb{L}(\mathbb{D}\mathbb{R})$$

Par unicité de la factorisation $\mathbb{L}\mathbb{U}$ :

$$\mathbb{R}^* = \mathbb{L} \text{ et} \mathbb{D}^*\mathbb{L}^* = \mathbb{D}\mathbb{R} \implies \mathbb{R}^* = \mathbb{L} \text{ et } \mathbb{D}^* = \mathbb{D}$$

### Théorème 8: Factorisation $\mathbb{L}\mathbb{D}\mathbb{L}^*$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ une matrice hermitienne inversible admettant une factorisation $\mathbb{L}\mathbb{U}$. Alors $\mathbb{A}$ s'écrit sous la forme

$$\mathbb{A} = \mathbb{L}\mathbb{D}\mathbb{L}^* \tag{9}$$

où $\mathbb{D} = \operatorname{diag} \mathbb{U}$ est une matrice à coefficients réels.

### Corollaire 8.1:

Une matrice $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ admet une factorisation $\mathbb{L}\mathbb{D}\mathbb{L}^*$ avec $\mathbb{L} \in \mathcal{M}_n(\mathbb{C})$ matrice triangulaire inférieure à diagonale unité et $\mathbb{D} \in \mathcal{M}_n(\mathbb{R})$ matrice diagonale à coefficients diagonaux strictement positifs **si et seulement si** la matrice $\mathbb{A}$ est hermitienne définie positive.

# Plan

1 Conditionnement
2 Méthodes directes
- Matrices particulières
  - Matrices diagonales
  - Matrices triangulaires inférieures
  - Matrices triangulaires supérieures
- Exercices et résultats préliminaires
- Méthode de Gauss-Jordan
  - Ecriture algébrique
- Factorisation $\mathbb{LU}$
  - Résultats théoriques
  - Utilisation pratique
- Factorisation $\mathbb{LDL}^*$
- Factorisation de Cholesky
  - Résultats théoriques
  - Résolution d'un système linéaire
  - Algorithme : Factorisation positive de Cholesky
- Factorisation $\mathbb{QR}$
  - La tranformation de Householder

### Definition

Une **factorisation régulière de Cholesky** d'une matrice $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ est une factorisation $\mathbb{A} = \mathbb{B}\mathbb{B}^*$ où $\mathbb{B}$ est une matrice triangulaire inférieure inversible.
Si les coefficients diagonaux de $\mathbb{B}$ sont positifs, on parle alors d'une **factorisation positive de Cholesky**.

### Théorème: Factorisation de Cholesky

La matrice $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ admet une factorisation régulière de Cholesky **si et seulement si** la matrice $\mathbb{A}$ est hermitienne définie positive. Dans ce cas, elle admet une unique factorisation positive.

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ hermitienne définie positive et $\boldsymbol{b} \in \mathbb{C}^n$. On note $\mathbb{B}$ la matrice de factorisation positive de Cholesky de $\mathbb{A}$.

Trouver $\boldsymbol{x} \in \mathbb{C}^n$ tel que

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b} \quad (\iff \mathbb{B}\mathbb{B}^*\boldsymbol{x} = \boldsymbol{b}) \tag{10}$$

est équivalent à

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ hermitienne définie positive et $\boldsymbol{b} \in \mathbb{C}^n$. On note $\mathbb{B}$ la matrice de factorisation positive de Cholesky de $\mathbb{A}$.

$$\boxed{\begin{array}{l}\text{Trouver } \boldsymbol{x} \in \mathbb{C}^n \text{ tel que} \\ \qquad \mathbb{A}\boldsymbol{x} = \boldsymbol{b} \quad (\Longleftrightarrow \quad \mathbb{B}\mathbb{B}^*\boldsymbol{x} = \boldsymbol{b}) \qquad (10)\end{array}}$$

est équivalent à

$$\boxed{\begin{array}{l}\text{Trouver } \boldsymbol{x} \in \mathbb{C}^n \text{ solution de} \\ \qquad \mathbb{B}^*\boldsymbol{x} = \boldsymbol{y} \qquad (11) \\ \text{avec } \boldsymbol{y} \in \mathbb{C}^n \text{ solution de} \\ \qquad \mathbb{B}\boldsymbol{y} = \boldsymbol{b}. \qquad (12)\end{array}}$$

# Algorithme de résolution de systèmes linéaire par Cholesky

**Algorithme 10** Fonction RSLCholesky permettant de résoudre, par une factorisation de Cholesky positive, le système linéaire

$$\mathbb{A}\boldsymbol{x} = \boldsymbol{b}$$

où $\mathbb{A}$ une matrice hermitienne de $\mathcal{M}_n(\mathbb{C})$ définie positive et $\boldsymbol{b} \in \mathbb{C}^n$.

**Données :** $\mathbb{A}$ : matrice de $\mathcal{M}_n(\mathbb{C})$ hermitienne définie positive,
$\boldsymbol{b}$ : vecteur de $\mathbb{C}^n$.

**Résultat :** $\boldsymbol{x}$ : vecteur de $\mathbb{C}^n$.

```
1: Fonction x ← RSLCholesky ( A, b )
2:     B ← Cholesky(A)           ▷ Factorisation positive de Cholesky
3:     y ← RSLTriInf(B, b)       ▷ Résolution du système By = b
4:     U ← MatAdjointe(B)        ▷ Calcul de la matrice adjointe de B
5:     x ← RSLTriSup(U, y)       ▷ Résolution du système B*x = y
6: Fin Fonction
```

Il nous faut donc écrire la fonction Cholesky

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ hermitienne définie positive. il existe une unique matrice $\mathbb{B} \in \mathcal{M}_n(\mathbb{C})$ triangulaire inférieure avec $\mathrm{B}_{i,i} \in \mathbb{R}^{+*}$, $\forall i \in [\![1, n]\!]$, telle que

$$\mathbb{A} = \mathbb{B}\mathbb{B}^*$$

c'est à dire

$$\begin{pmatrix} \mathrm{A}_{1,1} & \dots & \mathrm{A}_{1,n} \\ \vdots & \ddots & \dots \\ \mathrm{A}_{n,1} & \dots & \mathrm{A}_{n,n} \end{pmatrix} = \begin{pmatrix} \mathrm{B}_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \mathrm{B}_{n,1} & \dots & \dots & \mathrm{B}_{n,n} \end{pmatrix} \begin{pmatrix} \overline{\mathrm{B}_{1,1}} & \dots & \dots & \overline{\mathrm{B}_{n,1}} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \overline{\mathrm{B}_{n,n}} \end{pmatrix}.$$

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ hermitienne définie positive. il existe une unique matrice $\mathbb{B} \in \mathcal{M}_n(\mathbb{C})$ triangulaire inférieure avec $\mathrm{B}_{i,i} \in \mathbb{R}^{+*}$, $\forall i \in [\![1, n]\!]$, telle que

$$\mathbb{A} = \mathbb{B}\mathbb{B}^*$$

c'est à dire

$$\begin{pmatrix} \mathrm{A}_{1,1} & \dots & \mathrm{A}_{1,n} \\ \vdots & \ddots & \dots \\ \mathrm{A}_{n,1} & \dots & \mathrm{A}_{n,n} \end{pmatrix} = \begin{pmatrix} \mathrm{B}_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \mathrm{B}_{n,1} & \dots & \dots & \mathrm{B}_{n,n} \end{pmatrix} \begin{pmatrix} \overline{\mathrm{B}_{1,1}} & \dots & \dots & \overline{\mathrm{B}_{n,1}} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \overline{\mathrm{B}_{n,n}} \end{pmatrix}.$$

- Calcul de $\mathrm{B}_{1,1}$ (la 1ère ligne de $\mathbb{B}$ est donc déterminée) $\Longrightarrow$ calcul 1ère colonne de $\mathbb{B}$.

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ hermitienne définie positive. il existe une unique matrice $\mathbb{B} \in \mathcal{M}_n(\mathbb{C})$ triangulaire inférieure avec $\mathrm{B}_{i,i} \in \mathbb{R}^{+*}$, $\forall i \in [\![1, n]\!]$, telle que

$$\mathbb{A} = \mathbb{B}\mathbb{B}^*$$

c'est à dire

$$\begin{pmatrix} \mathrm{A}_{1,1} & \dots & \mathrm{A}_{1,n} \\ \vdots & \ddots & \dots \\ \mathrm{A}_{n,1} & \dots & \mathrm{A}_{n,n} \end{pmatrix} = \begin{pmatrix} \mathrm{B}_{1,1} & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 0 \\ \mathrm{B}_{n,1} & \dots & \dots & \mathrm{B}_{n,n} \end{pmatrix} \begin{pmatrix} \overline{\mathrm{B}_{1,1}} & \dots & \dots & \overline{\mathrm{B}_{n,1}} \\ 0 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \overline{\mathrm{B}_{n,n}} \end{pmatrix}.$$

- Calcul de $\mathrm{B}_{1,1}$ (la 1ère ligne de $\mathbb{B}$ est donc déterminée) $\Longrightarrow$ calcul 1ère colonne de $\mathbb{B}$.
- Puis calcul de $\mathrm{B}_{2,2}$ (la 2ème ligne de $\mathbb{B}$ est donc déterminée) $\Longrightarrow$ calcul 2ème colonne de $\mathbb{B}$.
- Etc...

Soit $i \in [\![1, n]\!]$. On suppose connues les $i-1$ premières colonnes de $\mathbb{B}$.

Peut-on calculer la colonne $i$ de $\mathbb{B}$?

$$\mathbb{A} = \mathbb{B}\mathbb{B}^* \implies \mathrm{A}_{i,i} = \sum_{k=1}^{n} \mathrm{B}_{i,k} (\mathbb{B}^*)_{k,i} = \sum_{k=1}^{n} \mathrm{B}_{i,k} \overline{\mathrm{B}_{i,k}}$$

Or $\mathbb{B}$ triangulaire inférieure (i.e. $\mathrm{B}_{i,j} = 0$ si $j > i$)

$$\mathrm{A}_{i,i} = \sum_{k=1}^{i-1} |\mathrm{B}_{i,k}|^2 + |\mathrm{B}_{i,i}|^2$$

et donc

$$\mathrm{B}_{i,i} = \left( \mathrm{A}_{i,i} - \sum_{j=1}^{i-1} |\mathrm{B}_{i,j}|^2 \right)^{1/2}.$$

Il reste à déterminer $\mathrm{B}_{j,i},\ \forall j \in [\![i+1, n]\!]$.

$$\mathrm{A}_{j,i} = \sum_{k=1}^{n} \mathrm{B}_{j,k}(\mathbb{B}^*)_{k,i} = \sum_{k=1}^{n} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}},\ \forall j \in [\![i+1, n]\!]$$

Comme $\mathbb{L}$ est triangulaire inférieure on obtient

$$\mathrm{A}_{j,i} = \sum_{k=1}^{i} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}} = \sum_{k=1}^{i-1} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}} + \mathrm{B}_{j,i}\overline{\mathrm{B}_{i,i}},\ \forall j \in [\![i+1, n]\!]$$

Or $\mathrm{B}_{i,i} > 0$ connu et les $i-1$ premières colonnes de $\mathbb{B}$ aussi.

$$\begin{aligned}
\mathrm{B}_{j,i} &= \frac{1}{\mathrm{B}_{i,i}}\left(\mathrm{A}_{j,i} - \sum_{k=1}^{i-1} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}}\right),\ \forall j \in [\![i+1, n]\!] \\
\mathrm{B}_{j,i} &= 0,\ \forall j \in [\![1, i-1]\!].
\end{aligned}$$

**Algorithme 11** $\boxed{\mathcal{R}_0}$

1: $\boxed{\text{Calculer la matrice } \mathbb{B}}$

**Algorithme 11** $\boxed{\mathcal{R}_1}$

1: **Pour** $i \leftarrow 1$ **à** $n$ **faire**
2: Calculer $\mathrm{B}_{i,i}$, connaissant les $i-1$ premières colonnes de $\mathbb{B}$.
3: Calculer la $i^{\text{ème}}$ colonne de $\mathbb{B}$.
4: **Fin Pour**

**Algorithme 11** $\boxed{\mathcal{R}_1}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**
  $\boxed{\text{Calculer } \mathrm{B}_{i,i}\text{, connaissant les } i-1 \text{ premières colonnes de } \mathbb{B}.}$
  $\boxed{\text{Calculer la } i^{\text{ème}} \text{ colonne de } \mathbb{B}.}$
**Fin Pour**

**Algorithme 11** $\boxed{\mathcal{R}_2}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$$\mathrm{B}_{i,i} \leftarrow \left( \mathrm{A}_{i,i} - \sum_{j=1}^{i-1} |\mathrm{B}_{i,j}|^2 \right)^{1/2}$$

  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
    $\mathrm{B}_{j,i} \leftarrow 0$
  **Fin Pour**
  **Pour** $j \leftarrow i+1$ **à** $n$ **faire**

$$\mathrm{B}_{j,i} \leftarrow \frac{1}{\mathrm{B}_{i,i}} \left( \mathrm{A}_{j,i} - \sum_{k=1}^{i-1} \mathrm{B}_{j,k} \overline{\mathrm{B}_{i,k}} \right).$$

  **Fin Pour**
**Fin Pour**

**Algorithme 11** $\boxed{\mathcal{R}_2}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$$\boxed{\mathrm{B}_{i,i} \leftarrow \left(\mathrm{A}_{i,i} - \sum_{j=1}^{i-1} |\mathrm{B}_{i,j}|^2\right)^{1/2}}$$

  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**

    $\mathrm{B}_{j,i} \leftarrow 0$

  **Fin Pour**

  **Pour** $j \leftarrow i+1$ **à** $n$ **faire**

$$\boxed{\mathrm{B}_{j,i} \leftarrow \frac{1}{\mathrm{B}_{i,i}}\left(\mathrm{A}_{j,i} - \sum_{k=1}^{i-1} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}}\right).}$$

  **Fin Pour**

**Fin Pour**

**Algorithme 11** $\boxed{\mathcal{R}_3}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$$\left\{\begin{array}{l} S_1 \leftarrow \displaystyle\sum_{j=1}^{i-1} |\mathrm{B}_{i,j}|^2 \\ \mathrm{B}_{i,i} \leftarrow (\mathrm{A}_{i,i} - S_1)^{1/2} \end{array}\right.$$

  **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**

    $\mathrm{B}_{j,i} \leftarrow 0$

  **Fin Pour**

  **Pour** $j \leftarrow i+1$ **à** $n$ **faire**

$$\left\{\begin{array}{l} S_2 \leftarrow \displaystyle\sum_{k=1}^{i-1} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}} \\ \mathrm{B}_{j,i} \leftarrow \dfrac{1}{\mathrm{B}_{i,i}}(\mathrm{A}_{j,i} - S_2). \end{array}\right.$$

  **Fin Pour**

**Fin Pour**

**Algorithme 11** $\boxed{\mathcal{R}_3}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$$\boxed{S_1 \leftarrow \sum_{j=1}^{i-1} |\mathrm{B}_{i,j}|^2}$$

$\quad \mathrm{B}_{i,i} \leftarrow (\mathrm{A}_{i,i} - S_1)^{1/2}$

$\quad$ **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**

$\quad\quad \mathrm{B}_{j,i} \leftarrow 0$

$\quad$ **Fin Pour**

$\quad$ **Pour** $j \leftarrow i+1$ **à** $n$ **faire**

$$\boxed{S_2 \leftarrow \sum_{k=1}^{i-1} \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}}}$$

$$\mathrm{B}_{j,i} \leftarrow \frac{1}{\mathrm{B}_{i,i}} (\mathrm{A}_{j,i} - S_2).$$

$\quad$ **Fin Pour**

**Fin Pour**

**Algorithme 11** $\boxed{\mathcal{R}_4}$

**Pour** $i \leftarrow 1$ **à** $n$ **faire**

$\quad S_1 \leftarrow 0$

$\quad$ **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**

$\quad\quad S_1 \leftarrow S_1 + |\mathrm{B}_{i,j}|^2$

$\quad$ **Fin Pour**

$\quad \mathrm{B}_{i,i} \leftarrow (\mathrm{A}_{i,i} - S_1)^{1/2}$

$\quad$ **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**

$\quad\quad \mathrm{B}_{j,i} \leftarrow 0$

$\quad$ **Fin Pour**

$\quad$ **Pour** $j \leftarrow i+1$ **à** $n$ **faire**

$\quad\quad S_2 \leftarrow 0$

$\quad\quad$ **Pour** $k \leftarrow 1$ **à** $i-1$ **faire**

$\quad\quad\quad S_2 \leftarrow S_2 + \mathrm{B}_{j,k}\overline{\mathrm{B}_{i,k}}$

$\quad\quad$ **Fin Pour**

$$\mathrm{B}_{j,i} \leftarrow \frac{1}{\mathrm{B}_{i,i}} (\mathrm{A}_{j,i} - S_2).$$

$\quad$ **Fin Pour**

**Fin Pour**

**Algorithme 11** Fonction CHOLESKY permettant de calculer la matrice $\mathbb{B}$, dites matrice de factorisation positive de Cholesky associée à la matrice $\mathbb{A}$, telle que

$$\mathbb{A} = \mathbb{B}\mathbb{B}^*.$$

**Données :** $\mathrm{A}$ : matrice de $\mathcal{M}_n(\mathbb{C})$ hermitienne définie positive.
**Résultat :** $\mathrm{B}$ : matrice de $\mathcal{M}_n(\mathbb{C})$ triangulaire inférieure avec $\mathrm{B}(i,i) > 0, \; \forall i \in [\![1, n]\!]$

**Fonction** $\mathrm{B} \leftarrow$ CHOLESKY ( $\mathrm{A}$ )
  **Pour** $i \leftarrow 1$ **à** $n$ **faire**
    $S_1 \leftarrow 0$
    **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
      $S_1 \leftarrow S_1 + |\mathrm{B}(i,j)|^2$
    **Fin Pour**
    $\mathrm{B}(i,i) \leftarrow \mathrm{SQRT}(\mathrm{A}(i,i) - S_1)$
    **Pour** $j \leftarrow 1$ **à** $i-1$ **faire**
      $\mathrm{B}(j,i) \leftarrow 0$
    **Fin Pour**
    **Pour** $j \leftarrow i+1$ **à** $n$ **faire**
      $S_2 \leftarrow 0$
      **Pour** $k \leftarrow 1$ **à** $i-1$ **faire**
        $S_2 \leftarrow S_2 + \mathrm{B}(j,k) * \overline{\mathrm{B}(i,k)}$
      **Fin Pour**
      $\mathrm{B}(j,i) \leftarrow (\mathrm{A}(j,i) - S_2)/\mathrm{B}(i,i).$
    **Fin Pour**
  **Fin Pour**
**Fin Fonction**

### Exercice 8.1

Proposer une méthode permettant de tester la fonction CHOLESKY .

# Plan

1. Conditionnement
2. Méthodes directes
   - Matrices particulières
     - Matrices diagonales
     - Matrices triangulaires inférieures
     - Matrices triangulaires supérieures
   - Exercices et résultats préliminaires
   - Méthode de Gauss-Jordan
     - Ecriture algébrique
   - Factorisation $\mathbb{LU}$
     - Résultats théoriques
     - Utilisation pratique
   - Factorisation $\mathbb{LDL}^*$
   - Factorisation de Cholesky
     - Résultats théoriques
     - Résolution d'un système linéaire
     - Algorithme : Factorisation positive de Cholesky
   - Factorisation $\mathbb{QR}$
     - La tranformation de Householder

**Definition: Matrice élémentaire de Householder**

Soit $\boldsymbol{u} \in \mathbb{C}^n$ tel que $\|\boldsymbol{u}\|_2 = 1$. On appelle **matrice élémentaire de Householder** la matrice $\mathbb{H}(\boldsymbol{u}) \in \mathcal{M}_n(\mathbb{C})$ définie par

$$\mathbb{H}(\boldsymbol{u}) = \mathbb{I} - 2\boldsymbol{u}\boldsymbol{u}^*. \tag{13}$$

**Propriété:**

Toute matrice élémentaire de Householder est hermitienne et unitaire.

**Propriété:**

Soient $\boldsymbol{x} \in \mathbb{K}^n$ et $\boldsymbol{u} \in \mathbb{K}^n$, $\|\boldsymbol{u}\|_2 = 1$. On note $\boldsymbol{x}_{\|} = \mathrm{proj}_{\boldsymbol{u}}(\boldsymbol{x}) \stackrel{\text{def}}{=} \langle \boldsymbol{u}, \boldsymbol{x} \rangle \boldsymbol{u}$ et $\boldsymbol{x}_{\perp} = \boldsymbol{x} - \boldsymbol{x}_{\|}$. On a alors

$$\mathbb{H}(\boldsymbol{u})(\boldsymbol{x}_{\perp} + \boldsymbol{x}_{\|}) = \boldsymbol{x}_{\perp} - \boldsymbol{x}_{\|}. \tag{14}$$

et

$$\mathbb{H}(\boldsymbol{u})\boldsymbol{x} = \boldsymbol{x}, \quad \text{si } \langle \boldsymbol{x}, \boldsymbol{u} \rangle = 0. \tag{15}$$

**Théorème:**

Soient $\boldsymbol{a}$, $\boldsymbol{b}$ deux vecteurs non colinéaires de $\mathbb{C}^n$ avec $\|\boldsymbol{b}\|_2 = 1$. Soit $\alpha \in \mathbb{C}$ tel que $|\alpha| = \|\boldsymbol{a}\|_2$ et $\arg \alpha = -\arg \langle \boldsymbol{a}, \boldsymbol{b} \rangle \ [\pi]$. On a alors

$$\mathbb{H}\left( \frac{\boldsymbol{a} - \alpha \boldsymbol{b}}{\|\boldsymbol{a} - \alpha \boldsymbol{b}\|_2} \right) \boldsymbol{a} = \alpha \boldsymbol{b}. \tag{16}$$

## Exercice:

Soient $\boldsymbol{a}$ et $\boldsymbol{b}$ deux vecteurs non nuls et non colinéaires de $\mathbb{C}^n$ avec $\|\boldsymbol{b}\|_2 = 1$.

**Q. 1**

*Ecrire la fonction algorithmique* HOUSEHOLDER *permettant de retourner une matrice de Householder* $\mathbb{H}$ *et* $\alpha \in \mathbb{C}$ *tels que* $\mathbb{H}(\boldsymbol{u})\boldsymbol{a} = \alpha\boldsymbol{b}$. *Le choix du* $\alpha$ *est fait par le paramètre* $\delta$ *(0 ou 1) de telle sorte que* $\arg\alpha = -\arg(\langle \boldsymbol{a}, \boldsymbol{b}\rangle) + \delta\pi$ *avec* $|\alpha| = \|\boldsymbol{a}\|_2$ *.*
*Des fonctions comme* DOT$(\boldsymbol{a}, \boldsymbol{b})$ *(produit scalaire de deux vecteurs),* NORM$(\boldsymbol{a})$ *(norme 2 d'un vecteur),* ARG$(z)$ *(argument d'un nombre complexe),* MATPROD$(\mathbb{A}, \mathbb{B})$ *(produit de deux matrices),* CTRANSPOSE$(\mathbb{A})$ *(adjoint d'une matrice), ... pourront être utilisées*

**Q. 2**

*Proposer un programme permettant de tester cette fonction. On pourra utiliser la fonction* VECRAND$(n)$ *retournant un vecteur aléatoire de* $\mathbb{C}^n$*, les parties réelles et imaginaires de chacune de ses composantes étant dans* $]0, 1[$ *(loi uniforme).*

**Q. 3**

*Proposer un programme permettant de vérifier que* $\delta = 1$ *est le "meilleur" choix.*

## Corollaire 8.2:

Soit $\boldsymbol{a} \in \mathbb{C}^n$ avec $a_1 \neq 0$ et $\exists j \in [\![2, n]\!]$ tel que $a_j \neq 0$. Soient $\theta = \arg a_1$ et

$$\boldsymbol{u}_{\pm} = \frac{\boldsymbol{a} \pm \|\boldsymbol{a}\|_2 e^{\imath\theta}\boldsymbol{e}_1}{\left\|\boldsymbol{a} \pm \|\boldsymbol{a}\|_2 e^{\imath\theta}\boldsymbol{e}_1\right\|}$$

Alors

$$\mathbb{H}(\boldsymbol{u}_{\pm})\boldsymbol{a} = \mp \|\boldsymbol{a}\|_2 e^{\imath\theta}\boldsymbol{e}_1 \tag{17}$$

où $\boldsymbol{e}_1$ désigne le premier vecteur de la base canonique de $\mathbb{C}^n$.

## Théorème 9:

Soit $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$ une matrice. Il existe une matrice unitaire $\mathbb{Q} \in \mathcal{M}_n(\mathbb{C})$ produit d'au plus $n-1$ matrices de Householder et une matrice triangulaire supérieure $\mathbb{R} \in \mathcal{M}_n(\mathbb{C})$ telles que

$$\mathbb{A} = \mathbb{Q}\mathbb{R}. \tag{18}$$

Si $\mathbb{A}$ est réelle alors $\mathbb{Q}$ et $\mathbb{R}$ sont aussi réelles et l'on peut choisir $\mathbb{Q}$ de telle sorte que les coefficients diagonaux de $\mathbb{R}$ soient positifs. De plus, si $\mathbb{A}$ est inversible alors la factorisation est unique.

## Exercice 9.1: Algorithmique

**Q. 1**

*Ecrire une fonction* FactQR *permettant de calculer la factorisation* $\mathbb{QR}$ *d'une matrice* $\mathbb{A} \in \mathcal{M}_n(\mathbb{C})$.
*On pourra utiliser la fonction* Householder *(voir Exercice 55, page 79)*.

**Q. 2**

*Ecrire un programme permettant de tester cette fonction.*