

# Méthodes Numériques I

## Algorithmique numérique

François Cuvelier

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

6 avril 2016

# Plan

- 1 Polynômes d'interpolation de Lagrange
- 2 Dérivation numérique
- 3 Intégration numérique
- 4 Résolution de systèmes linéaires
  - Matrices particulières
    - Matrices diagonales
    - Matrices triangulaires inférieures
    - Matrices triangulaires supérieures
  - Méthode de Gauss-Jordan

# Plan

1 Polynômes d'interpolation de Lagrange

**2 Dérivation numérique**

3 Intégration numérique

4 Résolution de systèmes linéaires

- Matrices particulières
  - Matrices diagonales
  - Matrices triangulaires inférieures
  - Matrices triangulaires supérieures
- Méthode de Gauss-Jordan

# Plan

1 Polynômes d'interpolation de Lagrange

2 Dérivation numérique

**3 Intégration numérique**

4 Résolution de systèmes linéaires

- Matrices particulières
  - Matrices diagonales
  - Matrices triangulaires inférieures
  - Matrices triangulaires supérieures
- Méthode de Gauss-Jordan

# Plan

- 1 Polynômes d'interpolation de Lagrange
- 2 Dérivation numérique
- 3 Intégration numérique
- 4 Résolution de systèmes linéaires**
  - **Matrices particulières**
    - Matrices diagonales
    - Matrices triangulaires inférieures
    - Matrices triangulaires supérieures
  - **Méthode de Gauss-Jordan**

# Système diagonal

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

$$x_i = b_i/A_{i,i}, \quad \forall i \in \llbracket 1, n \rrbracket. \quad (1)$$

---

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

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

---

**Données :**  $\mathbb{A}$  : matrice diagonale de  $\mathcal{M}_n(\mathbb{R})$  inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{R}^n$ .

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

- 1: **Fonction**  $\mathbf{x} \leftarrow \mathbf{RSLMATDIAG}(\mathbb{A}, \mathbf{b})$
- 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  $i < j$ )

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

$\mathbb{A}$  inversible  $\iff$

# Système triangulaire inférieur

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

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

$$\mathbb{A} \text{ inversible} \iff A_{i,i} \neq 0, \forall i \in \llbracket 1, n \rrbracket$$

# Système triangulaire inférieur

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

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

$$A \text{ inversible} \iff A_{i,i} \neq 0, \forall i \in \llbracket 1, n \rrbracket$$

Soit  $i \in \llbracket 1, n \rrbracket$ ,  $(Ax)_i = b_i, \iff \sum_{j=1}^n A_{i,j}x_j = b_i.$

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

$$x_i = \frac{1}{A_{i,i}} \left( b_i - \sum_{j=1}^{i-1} A_{i,j}x_j \right), \forall i \in \llbracket 1, n \rrbracket. \quad (2)$$

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

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

1:

Résoudre  $\Delta \mathbf{x} = \mathbf{b}$  en calculant successivement  $x_1, x_2, \dots, x_n$ .

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

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

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

3: **Fin Pour**

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

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

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

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

3: **Fin Pour**

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

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

2:  $S \leftarrow \sum_{j=1}^{i-1} A_{i,j} x_j$

3:  $x_i \leftarrow (b_i - S) / A_{i,i}$

4: **Fin Pour**

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

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

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

2:  $S \leftarrow \sum_{j=1}^{i-1} A_{i,j} x_j$

3:  $x_i \leftarrow (b_i - S) / A_{i,i}$

4: **Fin Pour**

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

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

2:  $S \leftarrow 0$

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

4:  $S \leftarrow S + A(i, j) * x(j)$

5: **Fin Pour**

6:  $x_i \leftarrow (b_i - S) / A_{i,i}$

7: **Fin Pour**

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

---

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

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

---

**Données :**  $\mathbb{A}$  : matrice triangulaire de  $\mathcal{M}_n(\mathbb{K})$  inférieure inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

**Résultat :**  $\mathbf{x}$  : vecteur de  $\mathbb{K}^n$ .

- 1: **Fonction**  $\mathbf{x} \leftarrow \mathbf{RSLTriINF}(\mathbb{A}, \mathbf{b})$
- 2:   **Pour**  $i \leftarrow 1$  à  $n$  **faire**
- 3:      $S \leftarrow 0$
- 4:     **Pour**  $j \leftarrow 1$  à  $i - 1$  **faire**
- 5:        $S \leftarrow S + A(i, j) * x(j)$
- 6:     **Fin Pour**
- 7:      $x(i) \leftarrow (b(i) - S) / A(i, i)$
- 8:   **Fin Pour**
- 9: **Fin Fonction**

# Système triangulaire supérieur

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

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



## Exercice 1

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

# Algorithme de Gauss-Jordan

$$\mathbf{Ax} = \mathbf{b} \iff \mathbf{Ux} = \mathbf{f}$$

où  $\mathbf{U}$  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  $k$ , on va *s'arranger* pour annuler les termes sous-diagonaux de la colonne  $k$  de la matrice sans modifier les  $k - 1$  premières colonnes.

$$\begin{array}{c}
 \xrightarrow{k-1} \\
 \left( \begin{array}{cccc|cccc}
 \bullet & \bullet & \dots & \bullet & \bullet & \dots & \bullet \\
 0 & \dots & \dots & \vdots & \vdots & \dots & \vdots \\
 \vdots & \dots & \dots & \bullet & \vdots & \dots & \vdots \\
 0 & \dots & 0 & \bullet & \bullet & \dots & \bullet \\
 \hline
 0 & \dots & \dots & 0 & \bullet & \dots & \bullet \\
 \vdots & & & \vdots & \vdots & & \vdots \\
 0 & \dots & \dots & 0 & \bullet & \dots & \bullet
 \end{array} \right) \mathbf{x} = \begin{pmatrix} \bullet \\ \vdots \\ \bullet \\ \vdots \\ \bullet \end{pmatrix}
 \end{array}$$

Etape  $k$



$$\begin{array}{c}
 \xrightarrow{k} \\
 \left( \begin{array}{cccc|cccc}
 \bullet & \bullet & \dots & \bullet & \bullet & \dots & \bullet \\
 0 & \dots & \dots & \vdots & \vdots & \dots & \vdots \\
 \vdots & \dots & \dots & \bullet & \vdots & \dots & \vdots \\
 0 & \dots & 0 & \bullet & \bullet & \dots & \bullet \\
 \hline
 0 & \dots & \dots & 0 & \bullet & \dots & \bullet \\
 \vdots & & & \vdots & \vdots & & \vdots \\
 0 & \dots & \dots & 0 & \bullet & \dots & \bullet
 \end{array} \right) \mathbf{x} = \begin{pmatrix} \bullet \\ \vdots \\ \bullet \\ \vdots \\ \bullet \end{pmatrix}
 \end{array}$$

---

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

---

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

---

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

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$  inversible.

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

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

- 1: **Fonction**  $\mathbf{x} \leftarrow \text{RSLGAUSS}(\mathbb{A}, \mathbf{b})$
  - 2: **Pour**  $j \leftarrow 1$  à  $n - 1$  **faire**
  - 3:  $k \leftarrow \text{CHERCHEINDPIVOT}(\mathbb{A}, j)$  ▷ à écrire
  - 4:  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{PERMLIGNESYS}(\mathbb{A}, \mathbf{b}, j, k)$  ▷ à écrire
  - 5: **Pour**  $i \leftarrow j + 1$  à  $n$  **faire**
  - 6:  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{COMBLIGNESYS}(\mathbb{A}, \mathbf{b}, j, i, -A(i, j)/A(j, j))$  ▷ à écrire
  - 7: **Fin Pour**
  - 8: **Fin Pour**
  - 9:  $\mathbf{x} \leftarrow \text{RSLTRISUP}(\mathbb{A}, \mathbf{b})$  ▷ déjà écrite
  - 10: **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,  $1 \leq j \leq n$ .

**Résultat :**  $k$  : entier, indice ligne pivot

```
1: Fonction  $k \leftarrow \text{CERCHEINDPIVOT}(\mathbb{A}, j)$ 
2:  $k \leftarrow j$ , pivot  $\leftarrow |\mathbb{A}(j, j)|$ 
3: Pour  $i \leftarrow j + 1$  à  $n$  faire
4:   Si  $|\mathbb{A}(i, j)| >$  pivot alors
5:      $k \leftarrow i$ 
6:     pivot  $\leftarrow |\mathbb{A}(i, j)|$ 
7:   Fin Si
8: Fin Pour
9: Fin Fonction
```

---

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

---

**Données :**  $\mathbb{A}$  : matrice de  $\mathcal{M}_n(\mathbb{K})$ .

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

$j, k$  : entiers,  $1 \leq j, k \leq n$ .

**Résultat :**  $\mathbb{A}$  et  $\mathbf{b}$  modifiés.

```
1: Fonction  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{PERMLIGNESYS}(\mathbb{A}, \mathbf{b}, j, k)$ 
2:   Pour  $l \leftarrow 1$  à  $n$  faire
3:      $t \leftarrow \mathbb{A}(j, l)$ 
4:      $\mathbb{A}(j, l) \leftarrow \mathbb{A}(k, l)$ 
5:      $\mathbb{A}(k, l) \leftarrow t$ 
6:   Fin Pour
7:    $t \leftarrow \mathbf{b}(j)$ ,  $\mathbf{b}(j) \leftarrow \mathbf{b}(k)$ ,  $\mathbf{b}(k) \leftarrow t$ 
8: 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  $\mathcal{M}_n(\mathbb{K})$ .

$\mathbf{b}$  : vecteur de  $\mathbb{K}^n$ .

$j, i$  : entiers,  $1 \leq j, i \leq n$ .

alpha : scalaire de  $\mathbb{K}$

**Résultat :**  $\mathbb{A}$  et  $\mathbf{b}$  modifiés.

```
1: Fonction  $[\mathbb{A}, \mathbf{b}] \leftarrow \text{COMBLIGNESSYS}(\mathbb{A}, \mathbf{b}, j, i, \alpha)$ 
2:   Pour  $k \leftarrow 1$  à  $n$  faire
3:      $\mathbb{A}(i, k) \leftarrow \mathbb{A}(i, k) + \alpha * \mathbb{A}(j, k)$ 
4:   Fin Pour
5:    $\mathbf{b}(i) \leftarrow \mathbf{b}(i) + \alpha \mathbf{b}(j)$ 
6: Fin Fonction
```

