Calcul Scientifique Parallèle

Cours AMS301 — Automne 2021 — Cours 6

Résolution de systèmes linéaires : Méthodes directes

Axel Modave

Résolution de systèmes linéaires — Rappels

On cherche $\mathbf{x} \in \mathbb{R}^n$ tel que $\boxed{\mathbf{A}\mathbf{x} = \mathbf{b}}$ avec $\mathbf{A} \in \mathbb{R}^{n\times n}$ et $\mathbf{b} \in \mathbb{R}^n$.

Procédures de résolution

- **Méthodes directes :** Factorisation de $\mathbf{A}$ en matrices triangulaires et/ou diagonales (ex. $\mathbf{A} = \mathbf{L}\mathbf{U}$) et résolution de problèmes simples.

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{L}\mathbf{U}\mathbf{x} = \mathbf{b} \quad \Leftrightarrow \quad \left| \begin{array}{l} \mathbf{L}\mathbf{y} = \mathbf{b} \\ \mathbf{U}\mathbf{x} = \mathbf{y} \end{array} \right.$$

  Avantages : solution exacte obtenue après un nombre fini d'opérations
  Difficultés : coût important *(calcul/mémoire)*, parallélisation difficile

- **Méthodes itératives :** Procédure itérative permettant de minimiser une norme de l'erreur $\|\mathbf{x}^{(k)} - \mathbf{x}_{\text{ref}}\|$ et/ou d'une norme du résidu $\|\mathbf{A}\mathbf{x}^{(k)} - \mathbf{b}\|$.

$$\left| \begin{array}{l} \mathbf{x}^{(0)} = \mathrm{Iter}^{(0)}(\mathbf{A},\mathbf{b}) \\ \mathbf{x}^{(k+1)} = \mathrm{Iter}^{(k+1)}\left(\mathbf{x}^{(k)},\mathbf{x}^{(k-1)},\dots,\mathbf{A},\mathbf{b}\right), \quad \text{pour } k \geq 0 \end{array} \right.$$

  Avantages : coût limité par itération *(calcul/mémoire)*, parallélisation facile/efficace
  Difficultés : solution approchée, contrôle de la convergence du processus

Méthodes directes pour les systèmes linéaires

Systèmes triangulaires — Résolution par points [1/2]

Définitions

- Matrice triangulaire inférieure : $\mathbf{L} \in \mathbb{R}^{n\times n}$ avec $L_{ij} = 0$ si $i < j$ ($\mathbf{L}$ pour *'lower'*)
- Matrice triangulaire supérieure : $\mathbf{U} \in \mathbb{R}^{n\times n}$ avec $U_{ij} = 0$ si $i > j$ ($\mathbf{U}$ pour *'upper'*)

$$\mathbf{L} = \begin{bmatrix} L_{11} & 0 & 0 \\ L_{21} & L_{22} & 0 \\ L_{31} & L_{32} & L_{33} \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} U_{11} & U_{12} & U_{13} \\ 0 & U_{22} & U_{23} \\ 0 & 0 & U_{33} \end{bmatrix}$$

Propriétés

- Le déterminant d'une matrice triangulaire est le produit des éléments diagonaux :

$$\det(\mathbf{L}) = \prod_{i=1}^{n} L_{ii} \qquad \det(\mathbf{U}) = \prod_{i=1}^{n} U_{ii}$$

- Une matrice triangulaire est inversible
  si, et seulement si, ses éléments diagonaux sont non-nuls.

Systèmes triangulaires — Résolution par points [2/2]

Résolution de $\mathbf{Lx} = \mathbf{b}$ *(Algorithme de la descente)*

Données : $\mathbf{L}$ et $\mathbf{b}$
Initialisation : $\mathbf{x} \leftarrow 0$
for $i = 1, \dots, n$ **do**
  $x_i \leftarrow \left[b_i - \sum_{j=1}^{i-1} L_{ij} x_j\right] / L_{ii}$
end

Résolution de $\mathbf{Ux} = \mathbf{b}$ *(Algorithme de la remontée)*

Données : $\mathbf{U}$ et $\mathbf{b}$
Initialisation : $\mathbf{x} \leftarrow 0$
for $i = n, \dots, 1$ **do**
  $x_i \leftarrow \left[b_i - \sum_{j=i+1}^{n} U_{ij} x_j\right] / U_{ii}$
end

Aspects algorithmiques

- Coût : n^2 opérations
- Parallélisation faible *(calcul des sommes en parallèle)*

Systèmes triangulaires — Résolution par blocs [1/2]

Définitions

- Matrice triangulaire inférieure par blocs : $\mathbf{L} \in \mathbb{R}^{n \times n}$ avec $\mathbf{L}_{IJ} = 0$ si $I < J$
- Matrice triangulaire supérieure par blocs : $\mathbf{U} \in \mathbb{R}^{n \times n}$ avec $\mathbf{U}_{IJ} = 0$ si $I > J$

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_{11} & 0 & 0 \\ \mathbf{L}_{21} & \mathbf{L}_{22} & 0 \\ \mathbf{L}_{31} & \mathbf{L}_{32} & \mathbf{L}_{33} \end{bmatrix} \qquad \mathbf{U} = \begin{bmatrix} \mathbf{U}_{11} & \mathbf{U}_{12} & \mathbf{U}_{13} \\ 0 & \mathbf{U}_{22} & \mathbf{U}_{23} \\ 0 & 0 & \mathbf{U}_{33} \end{bmatrix}$$

Les blocs diagonaux sont nécessairement carrés. Les autres ne le sont pas forcément.

Propriété

- Le déterminant d'une matrice triangulaire par blocs est le produit des déterminants des blocs diagonaux :

$$\det(\mathbf{L}) = \prod_{I=1}^{N} \det(\mathbf{L}_{II}) \qquad \det(\mathbf{U}) = \prod_{I=1}^{N} \det(\mathbf{U}_{II})$$

- Une matrice diagonale ou triangulaire par blocs est inversible si, et seulement si, ses blocs diagonaux sont inversibles.

Systèmes triangulaires — Résolution par blocs [2/2]

Résolution de $\mathbf{Lx} = \mathbf{b}$ *(Algorithme de la descente)*

Données : $\mathbf{L}$ et $\mathbf{b}$
Initialisation : $\mathbf{x} \leftarrow 0$
for $I = 1, \dots, N$ **do**
  $\mathbf{x}_I \leftarrow \mathbf{L}_{II}^{-1} \left[\mathbf{b}_I - \sum_{J=1}^{I-1} \mathbf{L}_{IJ} \mathbf{x}_J\right]$
end

Résolution de $\mathbf{Ux} = \mathbf{b}$ *(Algorithme de la remontée)*

Données : $\mathbf{U}$ et $\mathbf{b}$
Initialisation : $\mathbf{x} \leftarrow 0$
for $I = N, \dots, 1$ **do**
  $\mathbf{x}_I \leftarrow \mathbf{U}_{II}^{-1} \left[\mathbf{b}_I - \sum_{J=I+1}^{N} \mathbf{U}_{IJ} \mathbf{x}_J\right]$
end

Aspects algorithmiques

- Coût : N petits systèmes à résoudre et $N(N-1)/2$ produits matrice-vecteur
- Meilleure parallélisation *(calcul des produits matrice-vecteur en parallèle)*

Systèmes généraux — Élimination de Gauss [1/5]

Objectif : Transformer $\mathbf{Ax} = \mathbf{b}$ en un système équivalent $\mathbf{Ux} = \mathbf{b}'$.

Procédure :

- Initialisation :

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & A_{1n}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} & \cdots & A_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ A_{n1}^{(1)} & A_{n2}^{(1)} & \cdots & A_{nn}^{(1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{bmatrix}$$

- Itération 1 :

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & A_{1n}^{(1)} \\ 0 & A_{22}^{(2)} & \cdots & A_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & A_{n2}^{(2)} & \cdots & A_{nn}^{(2)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{bmatrix}$$

avec

$$\alpha_{i1} = A_{i1}^{(1)} / A_{11}^{(1)} \qquad i = 2, \ldots, n \qquad \text{(multiplicateurs)}$$

$$A_{ij}^{(2)} = A_{ij}^{(1)} - \alpha_{i1} A_{1j}^{(1)} \qquad i, j = 2, \ldots, n$$

$$b_i^{(2)} = b_i^{(1)} - \alpha_{i1} b_1^{(1)} \qquad i = 2, \ldots, n$$

Systèmes généraux — Élimination de Gauss [2/5]

Objectif : Transformer $\mathbf{Ax} = \mathbf{b}$ en un système équivalent $\mathbf{Ux} = \mathbf{b}'$.

Procédure :

- Itération k :

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & & \cdots & & \cdots & A_{1n}^{(1)} \\ 0 & A_{22}^{(2)} & & & & & & A_{2n}^{(2)} \\ \vdots & & \ddots & & & & & \vdots \\ 0 & \cdots & 0 & A_{k+1,k+1}^{(k)} & & \cdots & & A_{k+1,n}^{(k)} \\ \vdots & & \vdots & \vdots & & & & \vdots \\ 0 & \cdots & 0 & A_{n,k+1}^{(k)} & & \cdots & & A_{n,n}^{(k)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{k+1} \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_{k+1}^{(k)} \\ \vdots \\ b_n^{(k)} \end{bmatrix}$$

- Itération $n-1$:

$$\begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & \cdots & A_{1n}^{(1)} \\ 0 & A_{22}^{(2)} & & & A_{2n}^{(2)} \\ \vdots & & \ddots & & \vdots \\ 0 & & & \ddots & \vdots \\ 0 & & & & A_{nn}^{(n)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ \vdots \\ b_n^{(n)} \end{bmatrix}$$

Systèmes généraux — Élimination de Gauss [3/5]

Élimination de Gauss

Données : $\mathbf{A}$ et $\mathbf{b}$

Initialisation : $\mathbf{A}^{(1)} = \mathbf{A}$ et $\mathbf{b}^{(1)} = \mathbf{b}$

for $k = 1, \ldots, n-1$ **do**

  for $i = k+1, \ldots, n$ **do**

    $\alpha_{ik} = A_{ik}^{(k)} / A_{kk}^{(k)}$ Si $A_{kk}^{(k)} \neq 0$!

    $A_{ij}^{(k+1)} = A_{ij}^{(k)} - \alpha_{ik} A_{kj}^{(k)} \quad (j = k+1, \ldots, n)$

    $b_i^{(k+1)} = b_i^{(k)} - \alpha_{ik} b_k^{(k)}$

  end

end

Élimination de Gauss *(Réécriture)*

Données : $\mathbf{A}$ et $\mathbf{b}$

for $k = 1, \ldots, n-1$ **do**

  for $i = k+1, \ldots, n$ **do**

    $\alpha \leftarrow A_{ik} / A_{kk}$ Si $A_{kk} \neq 0$!

    $A_{ij} \leftarrow A_{ij} - \alpha A_{kj} \quad (j = k+1, \ldots, n)$

    $A_{ik} \leftarrow 0$

    $b_i \leftarrow b_i - \alpha b_k$

  end

end

Systèmes généraux — Élimination de Gauss [4/5]

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} A'_{11} & A'_{12} & A'_{13} \\ 0 & A'_{22} & A'_{23} \\ 0 & 0 & A'_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} b'_1 \\ b'_2 \\ b'_3 \end{bmatrix}$$

Coût de résolution

- Élimination de Gauss : $2(n-1)n(n+1)/3 + n(n-1)$ flop
- Résolution par remontée : n^2 flop
- Terme dominant : $2/3\, n^3$ flop

Conditions d'utilisation

- La méthode doit être modifiée si $A_{kk}^{(k)} = 0$ à une étape.
- La méthode est utilisable sans modification si :
  - → $\mathbf{A}$ est à diagonale dominante par ligne ou par colonne
  - → $\mathbf{A}$ est symétrique définie positive
- Pour les autres cas ⇒ Permutation de lignes et/ou colonnes

Systèmes généraux — Élimination de Gauss [5/5]

Élimination de Gauss avec pivot

Données : $\mathbf{A}$ et $\mathbf{b}$
for $k = 1, \dots, n-1$ **do**
  Trouver $r \in [k, \dots, n]$ qui maximise $|A_{rk}|$ (si $\max_r |A_{rk}| = 0$, alors $\mathbf{A}$ pas inversible)
  Échanger les lignes r et k de $\mathbf{A}$ et de $\mathbf{b}$
  for $i = k+1, \dots, n$ **do**
    $\alpha \leftarrow A_{ik} / A_{kk}$
    $A_{ij} \leftarrow A_{ij} - \alpha A_{kj} \quad (j = k+1, \dots, n)$
    $A_{ik} \leftarrow 0$
    $b_i \leftarrow b_i - \alpha b_k$
  end
end

Stratégies de pivotages possibles :

- Trouver $r \in [k, \dots, n]$ qui maximise $|A_{kr}|$ *(pivotage partiel)*
- Trouver $r \in [k, \dots, n]$ et $s \in [k, \dots, n]$ qui maximise $|A_{rs}|$ *(pivotage total)*

On peut toujours trouver $r \in [k, \dots, n]$ tel que $|A_{rk}| > 0$, sinon $\mathbf{A}$ n'est pas inversible.
$\Rightarrow$ L'élimination de Gauss avec pivot est applicable à toute matrice inversible !

Méthodes directes pour les systèmes linéaires

Résolution de systèmes triangulaires
Élimination de Gauss
Factorisation de matrices

Généralités sur la factorisation de matrices [1/2]

Principe

- Factoriser $\mathbf{A}$ en matrices triangulaires :

$$\underbrace{\begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}}_{\mathbf{A}} = \underbrace{\begin{bmatrix} \times & 0 & 0 \\ \times & \times & 0 \\ \times & \times & \times \end{bmatrix}}_{\mathbf{L}} \underbrace{\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \end{bmatrix}}_{\mathbf{U}}$$

- Réécrire le problème avec des sytèmes triangulaires :

$$\mathbf{Ax} = \mathbf{b} \quad \Leftrightarrow \quad \mathbf{LUx} = \mathbf{b} \quad \Leftrightarrow \quad \begin{cases} \mathbf{Ly} = \mathbf{b} \\ \mathbf{Ux} = \mathbf{y} \end{cases}$$

  et résoudre ces systèmes avec une descente/remontée.

Commentaires

- A priori, même coût de résolution que l'élimination de Gauss.
- Si le système doit être résolu pour plusieurs membres de droites, une seule factorisation est nécessaire.

Généralités sur la factorisation de matrices [2/2]

Différentes factorisations

Si $\mathbf{A}$ est inversible et factorisable *(à préciser plus loin)*, on a :

$\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$	(Gauss)	
$\mathbf{A} = \mathbf{L}\tilde{\mathbf{U}}$	(Gauss)	
$\mathbf{A} = \tilde{\mathbf{L}}\mathbf{D}\tilde{\mathbf{U}}$	(Gauss-Jordan)	
$\mathbf{A} = \tilde{\mathbf{L}}\mathbf{D}\tilde{\mathbf{L}}^T$	(Crout)	*Si $\mathbf{A}$ symétrique.*
$\mathbf{A} = \mathbf{L}\mathbf{L}^T$	(Cholesky)	*Si $\mathbf{A}$ symétrique définie positive (SDP).*

avec

$\mathbf{D}$: matrice diagonale
$\mathbf{L}$: matrice triangulaire inférieure
$\tilde{\mathbf{L}}$: matrice triangulaire inférieure à diagonale unité
$\mathbf{U}$: matrice triangulaire supérieure
$\tilde{\mathbf{U}}$: matrice triangulaire supérieure à diagonale unité

Factorisation de Gauss ($\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$) [1/4]

Procédure

- Initialisation :

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & A_{1n}^{(1)} \\ A_{21}^{(1)} & A_{22}^{(1)} & \cdots & A_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ A_{n1}^{(1)} & A_{n2}^{(1)} & \cdots & A_{nn}^{(1)} \end{bmatrix}$$

- Itération 1 :

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ \alpha_{21} & 1 & & & 0 \\ \alpha_{31} & 0 & 1 & & 0 \\ \vdots & \vdots & & \ddots & \vdots \\ \alpha_{n1} & 0 & \cdots & \cdots & 1 \end{bmatrix} \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & A_{1n}^{(1)} \\ 0 & A_{22}^{(2)} & \cdots & A_{2n}^{(2)} \\ 0 & A_{32}^{(2)} & \cdots & A_{3n}^{(2)} \\ \vdots & \vdots & & \vdots \\ 0 & A_{n2}^{(2)} & \cdots & A_{nn}^{(2)} \end{bmatrix}$$

avec

$$\alpha_{i1} = A_{i1}^{(1)} / A_{11}^{(1)} \qquad i = 2, \dots, n$$

$$A_{ij}^{(2)} = A_{ij}^{(1)} - \alpha_{i1} A_{1j}^{(1)} \qquad i, j = 2, \dots, n$$

Factorisation de Gauss ($\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$) [2/4]

Procédure *(suite)*

- Itération k :

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ \alpha_{21} & \ddots & & & & 0 \\ \vdots & \ddots & 1 & & & \vdots \\ \vdots & & \vdots & 1 & & \vdots \\ \vdots & & \vdots & & \ddots & \vdots \\ \alpha_{n1} & \cdots & \alpha_{nk} & 0 & \cdots & 1 \end{bmatrix} \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & \cdots & \cdots & A_{1n}^{(1)} \\ 0 & A_{22}^{(2)} & & & & A_{2n}^{(2)} \\ \vdots & & \ddots & & & \vdots \\ 0 & \cdots & 0 & A_{k+1,k+1}^{(k)} & \cdots & A_{k+1,n}^{(k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & A_{n,k+1}^{(k)} & \cdots & A_{nn}^{(k)} \end{bmatrix}$$

- Itération $n-1$:

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ \alpha_{21} & 1 & & & 0 \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & 1 & \vdots \\ \alpha_{n1} & \cdots & \cdots & \alpha_{n(n-1)} & 1 \end{bmatrix} \begin{bmatrix} A_{11}^{(1)} & A_{12}^{(1)} & \cdots & \cdots & A_{1n}^{(1)} \\ 0 & A_{22}^{(2)} & & & A_{2n}^{(2)} \\ \vdots & & \ddots & & \vdots \\ 0 & & & \ddots & \vdots \\ 0 & & & & A_{nn}^{(n)} \end{bmatrix}$$

Factorisation de Gauss ($\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$) [3/4]

Factorisation de Gauss

Donnée : $\mathbf{A}$
Initialisation : $\tilde{\mathbf{L}} = \mathbf{I}$ et $\mathbf{U} = \mathbf{A}$
for $k = 1, \dots, n-1$ **do**
  for $i = k+1, \dots, n$ **do**
    $\tilde{L}_{ik} \leftarrow U_{ik} / U_{kk}$ Si $U_{kk} \neq 0$!
    $U_{ij} \leftarrow U_{ij} - \tilde{L}_{ik} U_{kj}$ $(j = k+1 \dots n)$
    $U_{ik} \leftarrow 0$
  end
end

Factorisation de Gauss *(Réécriture)*

Donnée : $\mathbf{A}$
for $k = 1, \dots, n-1$ **do**
  for $i = k+1, \dots, n$ **do**
    $A_{ik} \leftarrow A_{ik} / A_{kk}$ Si $A_{kk} \neq 0$!
    $A_{ij} \leftarrow A_{ij} - A_{ik} A_{kj}$ $(j = k+1 \dots n)$
  end
end

Les éléments de $\tilde{\mathbf{L}}$ et $\mathbf{U}$ sont respectivement ceux de la partie triangulaire inférieure *(hors diagonale)* et triangulaire supérieure *(avec diagonale)* de $\mathbf{A}$.

Factorisation de Gauss ($\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$) [4/4]

Théorème – Unicité de la factorisation de Gauss ($\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$)

La factorisation de Gauss, si elle existe, est unique.

Théorème – Existence de la factorisation de Gauss ($\mathbf{A} = \tilde{\mathbf{L}}\mathbf{U}$) *(condition suffisante)*

Toute matrice *symétrique définie positive* est factorisable.

Matrice de permutation

- Une matrice de permutation élémentaire $\mathbf{P}^{(i_1,i_2)}$ est une matrice dont l'application sur une matrice $\mathbf{A}$ permute ses lignes i_1 et i_2.

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

- La matrice de permutation $\mathbf{P}^{(i_1,i_2)}$ est obtenue en permutant les lignes i_1 et i_2 et les colonnes i_1 et i_2 de la matrice identité.
  Propriétés intéressantes : $\mathbf{P}^{(i_1,i_2)}\mathbf{P}^{(i_1,i_2)} = \mathbf{I}_n$ et $(\mathbf{P}^{(i_1,i_2)})^T = \mathbf{P}^{(i_1,i_2)}$

Théorème – Existence de la factorisation de Gauss après permutation ($\mathbf{PA} = \tilde{\mathbf{L}}\mathbf{U}$)

Pour toute une matrice inversible, il existe une suite de permutations élémentaires telles que la matrice permutée admette une factorisation de Gauss.

Résolution avec factorisation

Factorisation de Gauss *(Réécriture)*

Donnée : $\mathbf{A}$
for $k = 1, \ldots, n-1$ **do**
  for $i = k+1, \ldots, n$ **do**
    $A_{ik} \leftarrow A_{ik} / A_{kk}$ Si $A_{kk} \neq 0$!
    $A_{ij} \leftarrow A_{ij} - A_{ik} A_{kj}$ $(j = k+1 \ldots n)$
  end
end

Coût de résolution

- Factorisation de Gauss ou Gauss-Jordan : $2/3\, n^3$ opérations
- Factorisation de Crout ou Cholesky : $1/3\, n^3$ opérations
- Remontée + Descente : $2\, n^2$ opérations
- Les techniques de pivot ajoutent des opérations !

Parallélisation

- Parallélisation difficile pour les problèmes avec matrices denses.
- Il est intéressant d'utiliser une stratégie *"par blocs"*.
- Stratégies possibles pour les problèmes avec matrices creuses.

Summary

- **Triangular systems**
  - Simple computational procedures, but not suited for parallel computing!
  - Computational cost:
    - By points: n^2 scalar operations
    - By blocks: N small systems to solve and $\mathcal{O}(N^2)$ matrix-vector products
  - Parall. comp.: by-block strategy *(dense mat.)* or ad-hoc strategy *(sparse mat.)*
- **General systems**
  - Approaches:
    - Gaussian elimination $\Rightarrow$ Gives the solution for a given vector $\mathbf{b}$
    - LU factorization + Two triangular systems to solve
      $\Rightarrow$ Factorization computed once and used for any vector $\mathbf{b}$
  - For any nonsingular matrix $\mathbf{A}$, $\exists$ permutation matrix $\mathbf{P}$ such that $\mathbf{PA} = \tilde{\mathbf{L}}\mathbf{U}$
  - Different factorizations: $\tilde{\mathbf{L}}\mathbf{U}$, $\tilde{\mathbf{L}}\mathbf{D}\tilde{\mathbf{U}}$ (Gen), $\tilde{\mathbf{L}}\mathbf{D}\tilde{\mathbf{L}}^\top$ (Sym), $\mathbf{L}\mathbf{L}^\top$ (SDP)
  - Computational cost: $\mathcal{O}(n^3)$ operations
  - Parall. comp.: by-block strategy *(dense mat.)* or ad-hoc strategy *(sparse mat.)*

Quelques ressources

- *Méthodes Numériques : Algorithmes, analyse et applications*
  A. Quarteroni, R. Sacco, F. Saleri (2007), Springer
- *Calcul scientifique parallèle*
  F. Magoulès et F.-X. Roux (2017), Dunod
- *Calcul scientifique parallèle*
  P. Ciarlet et E. Jamelot, polycopié de cours