Résolution de systèmes linéaires : Méthodes directes

Polytech'Paris-UPMC

Propriétés mathématiques

Rappels mathématiques

Soit à résoudre le système linéaire

$$Ax = b.$$

$$\begin{aligned} A \in M_n(\mathbb{R}) &\quad \text{: matrice carrée de dimension } n \times n \\ x, b \in \mathbb{R}^n &\quad \text{: vecteurs de dimension } n. \end{aligned}$$

Rappels mathématiques

Soit à résoudre le système linéaire

$$Ax = b.$$

$A \in M_n(\mathbb{R})$: matrice carrée de dimension $n \times n$

$x, b \in \mathbb{R}^n$: vecteurs de dimension n.

CNS d'existence de la solution :

Le système $Ax = b$ a une solution unique si et seulement si son déterminant est non nul.

Rappels mathématiques

Soit à résoudre le système linéaire

$$Ax = b.$$

$A \in M_n(\mathbb{R})$: matrice carrée de dimension $n \times n$
$x, b \in \mathbb{R}^n$: vecteurs de dimension n.

CNS d'existence de la solution :
Le système $Ax = b$ a une solution unique si et seulement si son déterminant est non nul.

Si le déterminant est nul :
$\Rightarrow$ Si $b \in Im(A)$ le système a une infinité de solutions
$\Rightarrow$ Si $b \in \mathbb{R}^n - Im(A)$ le système n'a pas de solution

Exemple 1 :

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

Exemple 1 :

$$\begin{aligned} 2x_1 + \ 3x_2 \quad &= \ 5 \\ 4x_1 - \ 3x_2 \quad &= 1 \end{aligned}$$

Le déterminant vaut $D = -18$, le système a une solution unique $x_1 = 1, x_2 = 1$

Exemples

Exemple 1 :

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

Le déterminant vaut $D = -18$, le système a une solution unique $x_1 = 1, x_2 = 1$

Exemple 2 :

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ 4x_1 + 6x_2 &= 10 \end{aligned}$$

Exemple 1 :

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

Le déterminant vaut $D = -18$, le système a une solution unique $x_1 = 1, x_2 = 1$

Exemple 2 :

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ 4x_1 + 6x_2 &= 10 \end{aligned}$$

Le déterminant vaut $D = 0$, le système a une infinité de solutions : $(1,1) + \lambda \times (3,-2)$, $(\lambda \in \mathbb{R})$.

Exemple 1 :

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

Le déterminant vaut $D = -18$, le système a une solution unique $x_1 = 1, x_2 = 1$

Exemple 2 :

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ 4x_1 + 6x_2 &= 10 \end{aligned}$$

Le déterminant vaut $D = 0$, le système a une infinité de solutions : $(1,1) + \lambda \times (3,-2)$, $(\lambda \in \mathbb{R})$.

Exemple 3 :

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ 4x_1 + 6x_2 &= 9 \end{aligned}$$

Exemples

Exemple 1 :

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

Le déterminant vaut $D = -18$, le système a une solution unique $x_1 = 1, x_2 = 1$

Exemple 2 :

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ 4x_1 + 6x_2 &= 10 \end{aligned}$$

Le déterminant vaut $D = 0$, le système a une infinité de solutions : $(1,1) + \lambda \times (3,-2)$, $(\lambda \in \mathbb{R})$.

Exemple 3 :

$$\begin{aligned} 2x_1 + 3x_2 &= 5 \\ 4x_1 + 6x_2 &= 9 \end{aligned}$$

Le déterminant vaut $D = 0$, le système n'a pas de solution.

On ne change pas la solution d’un système linéaire lorsque :

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,

Propriétés

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,
- on permute deux colonnes,

Propriétés

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,
- on permute deux colonnes,
- on multiplie une ligne par un **réel** non nul,

Propriétés

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,
- on permute deux colonnes,
- on multiplie une ligne par un **réel** non nul,
- on ajoute une ligne à une autre.

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,
- on permute deux colonnes,
- on multiplie une ligne par un **réel** non nul,
- on ajoute une ligne à une autre.

Nous allons donc utiliser ces transformations pour se ramener à un cas simple.

On ne change pas la solution d'un système linéaire lorsque :

- on permute deux lignes,
- on permute deux colonnes,
- on multiplie une ligne par un **réel** non nul,
- on ajoute une ligne à une autre.

Nous allons donc utiliser ces transformations pour se ramener à un cas simple.

! *Ces propriétés sont vraies dans* $\mathbb{R}$ *pas dans* $\mathbb{F}$

Principe général des algorithmes

Les matrices triangulaires

Pour certaines matrices, il est simple de calculer une solution.

Définition *Une matrice $A = (a_{ij})_{1 \leq i,j \leq n}$ est triangulaire supérieure (respectivement inférieure) si*

$\forall i, j$ *t.q.* $j > i$ *(resp.* $j > i$*)*

$$a_{ij} = 0$$

Si A est une matrice triangulaire supérieure, et si aucun élément diagonal n'est nul, la solution du système $Ax = b$ est :

$$\begin{cases} x_n = \frac{b_n}{a_{nn}} \\ x_i = \dfrac{b_i - \sum_{j=i+1}^{n} a_{ij} x_j}{a_{ii}} \end{cases}$$

Si A est une matrice triangulaire inférieure, et si aucun élément diagonal n'est nul, la solution du système $Ax = b$ est :

$$\begin{cases} x_1 = \frac{b_1}{a_{11}} \\ x_i = \dfrac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j}{a_{ii}} \end{cases}$$

Algorithme de remontée

Dans le cas des matrices triangulaires supérieures, l'algorithme est donc le suivant.

Données : $A = (A[i,j])_{1 \leq i,j \leq n}$, $b = (b[i])_{1 \leq i \leq n}$

début

- $x[n] \leftarrow \frac{b[n]}{A[n,n]}$
- **pour** $i = n-1 \ldots 1$ **faire**
  - $sum \leftarrow 0$
  - **pour** $k = i+1 \ldots n$ **faire**
    - $sum \leftarrow sum + A[i,k] \cdot x[k]$
  - $x[i] \leftarrow \frac{b[i] - sum}{A[i,i]}$

fin

Algorithme de remontée

Dans le cas des matrices triangulaires supérieures, l'algorithme est donc le suivant.

Données : $A = (A[i,j])_{1 \leq i,j \leq n}$, $b = (b[i])_{1 \leq i \leq n}$

début

$x[n] \leftarrow \frac{b[n]}{A[n,n]}$

pour $i = n-1 \ldots 1$ **faire**

$sum \leftarrow 0$

pour $k = i+1 \ldots n$ **faire**

$sum \leftarrow sum + A[i,k] \cdot x[k]$

$x[i] \leftarrow \frac{b[i] - sum}{A[i,i]}$

fin

- À partir d'un système d'équations linéaires quelconques,

Algorithme de remontée

Dans le cas des matrices triangulaires supérieures, l'algorithme est donc le suivant.

Données : $A = (A[i,j])_{1\leq i,j\leq n}$, $b = (b[i])_{1\leq i\leq n}$

début

$x[n] \leftarrow \frac{b[n]}{A[n,n]}$

pour $i = n-1 \ldots 1$ **faire**

$sum \leftarrow 0$

pour $k = i+1 \ldots n$ **faire**

$sum \leftarrow sum + A[i,k] \cdot x[k]$

$x[i] \leftarrow \frac{b[i]-sum}{A[i,i]}$

fin

- À partir d'un système d'équations linéaires quelconques,
- on triangularise le système,

Algorithme de remontée

Dans le cas des matrices triangulaires supérieures, l'algorithme est donc le suivant.

Données : $A = (A[i,j])_{1 \leq i,j \leq n}$, $b = (b[i])_{1 \leq i \leq n}$

début

$x[n] \leftarrow \frac{b[n]}{A[n,n]}$

pour $i = n-1 \ldots 1$ **faire**

$sum \leftarrow 0$

pour $k = i+1 \ldots n$ **faire**

$sum \leftarrow sum + A[i,k] \cdot x[k]$

$x[i] \leftarrow \frac{b[i]-sum}{A[i,i]}$

fin

- À partir d'un système d'équations linéaires quelconques,
- on triangularise le système,
- on résout le système triangulaire,

Algorithme de remontée

Dans le cas des matrices triangulaires supérieures, l'algorithme est donc le suivant.

Données : $A = (A[i,j])_{1 \leq i,j \leq n}$, $b = (b[i])_{1 \leq i \leq n}$

début

$x[n] \leftarrow \frac{b[n]}{A[n,n]}$

pour $i = n-1 \ldots 1$ **faire**

$sum \leftarrow 0$

pour $k = i+1 \ldots n$ **faire**

$sum \leftarrow sum + A[i,k] \cdot x[k]$

$x[i] \leftarrow \frac{b[i]-sum}{A[i,i]}$

fin

- À partir d'un système d'équations linéaires quelconques,
- on triangularise le système,
- on résout le système triangulaire,
- pour cela on utilise des permutations de lignes et de colonnes et des combinaisons linéaires de lignes.

Pour résoudre le système $Ax = b$, il faut appliquer les modifications à la fois à la matrice A et au vecteur b.
Il y a deux cas possibles :

Pour résoudre le système $Ax = b$, il faut appliquer les modifications à la fois à la matrice A et au vecteur b.
Il y a deux cas possibles :

- On souhaite résoudre une seule équation $Ax = b$.

Méthodes

Pour résoudre le système $Ax = b$, il faut appliquer les modifications à la fois à la matrice A et au vecteur b.
Il y a deux cas possibles :

- On souhaite résoudre une seule équation $Ax = b$.
  - On travaille sur la matrice $[A\ b]$ qui a n lignes et $n+1$ colonnes
    - ▷ *C'est l'élimination de* GAUSS

Méthodes

Pour résoudre le système $Ax = b$, il faut appliquer les modifications à la fois à la matrice A et au vecteur b.

Il y a deux cas possibles :

- On souhaite résoudre une seule équation $Ax = b$.
  - On travaille sur la matrice $[A\ b]$ qui a n lignes et $n + 1$ colonnes
    - ▷ *C'est l'élimination de* Gauss
  - Pour résoudre le système, il faut
    - Une triangularisation,
    - Une remontée (solution d'un système triangulaire).

Méthodes (suite)

- On doit résoudre plusieurs systèmes avec la même matrice $Ax = b_1 \ldots Ax = b_k$.

Méthodes (suite)

- On doit résoudre plusieurs systèmes avec la même matrice $Ax = b_1 \dots Ax = b_k$.
  - On décompose A en produit de deux matrices triangulaires (U est supérieure et L inférieure) :

$$A \quad = \quad L \cdot U$$

Méthodes (suite)

- On doit résoudre plusieurs systèmes avec la même matrice $Ax = b_1 \ldots Ax = b_k$.
  - On décompose A en produit de deux matrices triangulaires (U est supérieure et L inférieure) :

$$A = L \cdot U$$

  - Une résolution se fait grâce à deux systèmes triangulaires

$$Ax_k = b_k \quad \Leftrightarrow \quad \begin{cases} Ly_k = b_k \\ Ux_k = y_k \end{cases}$$

▷ *C'est la décomposition LU*

Méthodes (suite)

- On doit résoudre plusieurs systèmes avec la même matrice $Ax = b_1 \dots Ax = b_k$.
  - On décompose A en produit de deux matrices triangulaires (U est supérieure et L inférieure) :

$$A = L \cdot U$$

  - Une résolution se fait grâce à deux systèmes triangulaires

$$Ax_k = b_k \quad \Leftrightarrow \quad \left\{ \begin{array}{rcl} Ly_k & = & b_k \\ Ux_k & = & y_k \end{array} \right.$$

▷ *C'est la décomposition LU*

  - Il faut une triangularisation pour « préparer » la matrice et deux remontées par vecteur b_k.

Ce qu'il reste à faire

- Comment triangulariser ?

Ce qu'il reste à faire

- Comment triangulariser ?
- Quelles conditions ?

Ce qu'il reste à faire

- Comment triangulariser ?
- Quelles conditions ?
- Que faire pour les matrices singulières ?

Ce qu'il reste à faire

- Comment triangulariser ?
- Quelles conditions ?
- Que faire pour les matrices singulières ?
- Que faire pour les matrices rectangulaires ?

Ce qu'il reste à faire

- Comment triangulariser ?
- Quelles conditions ?
- Que faire pour les matrices singulières ?
- Que faire pour les matrices rectangulaires ?
- Conditionnement du problème ?

Triangularisation

Triangularisation simple

Contrairement à ce qu'on dit parfois, cette méthode a été rapportée pour la première fois par CHANG TS'ANG au 2[e] siècle avant JC. On l'appelle aussi méthode *fang-cheng*.

La méthode utilise :

- la multiplication par un scalaire
- la somme de deux lignes.

Le but de la méthode est d'annuler progressivement les coefficients qui se trouvent sous la diagonale.

Triangularisation

- On commence avec A une matrice n lignes et m colonnes

- On commence avec A une matrice n lignes et m colonnes
- Il y a n étapes :

- On commence avec A une matrice n lignes et m colonnes
- Il y a n étapes :
- À l'étape k, on annule sous la diagonale les coefficients de la colonne k :

Triangularisation

- On commence avec A une matrice n lignes et m colonnes
- Il y a n étapes :
- À l'étape k, on annule sous la diagonale les coefficients de la colonne k :
  - On appelle k[e] pivot ($p^{(k)}$) le coefficient de la diagonale

$$p^{(k)} = a_{k,k}$$

  - À chaque ligne $i > k$ on soustrait la ligne k multipliée par $\frac{a_{i,k}}{p^{(k)}}$:

$$q = a_{i,k}$$

$$\forall j, k \leq j \leq m \quad \text{on fait}$$

$$a_{i,j} = a_{i,j} - a_{k,j} . \frac{q}{p^{(k)}}$$

Triangularisation

- On commence avec A une matrice n lignes et m colonnes
- Il y a n étapes :
- À l'étape k, on annule sous la diagonale les coefficients de la colonne k :
  - On appelle k^e pivot ($p^{(k)}$) le coefficient de la diagonale
    $$p^{(k)} = a_{k,k}$$
  - À chaque ligne $i > k$ on soustrait la ligne k multipliée par $\frac{a_{i,k}}{p^{(k)}}$:
    $$q = a_{i,k}$$
    $$\forall j, k \leq j \leq m \quad \text{on fait}$$
    $$a_{i,j} = a_{i,j} - a_{k,j}.\frac{q}{p^{(k)}}$$
  - Par définition $\forall i > k$ lorsque $j = k$ on fait l'opération :
    $$\begin{aligned} a_{i,k} &= a_{i,k} - a_{i,k} \\ &= 0 \end{aligned}$$
    en pratique il ne *faut pas* calculer ces coefficients pour éviter les erreurs de calcul.

Triangularisation

- On commence avec A une matrice n lignes et m colonnes
- Il y a n étapes :
- À l'étape k, on annule sous la diagonale les coefficients de la colonne k :
  - On appelle k[e] pivot ($p^{(k)}$) le coefficient de la diagonale
    $$p^{(k)} = a_{k,k}$$
  - À chaque ligne $i > k$ on soustrait la ligne k multipliée par $\frac{a_{i,k}}{p^{(k)}}$:
    $$q = a_{i,k}$$
    $$\forall j, k \leq j \leq m \quad \text{on fait}$$
    $$a_{i,j} = a_{i,j} - a_{k,j} . \frac{q}{p^{(k)}}$$
  - Par définition $\forall i > k$ lorsque $j = k$ on fait l'opération :
    $$\begin{aligned} a_{i,k} &= a_{i,k} - a_{i,k} \\ &= 0 \end{aligned}$$
    en pratique il ne *faut pas* calculer ces coefficients pour éviter les erreurs de calcul.
- L'algorithme ne fonctionne pas si l'un des pivots est nul.

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \dots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \dots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \dots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

$$\begin{pmatrix} A \end{pmatrix}$$

Algorithme de triangularisation

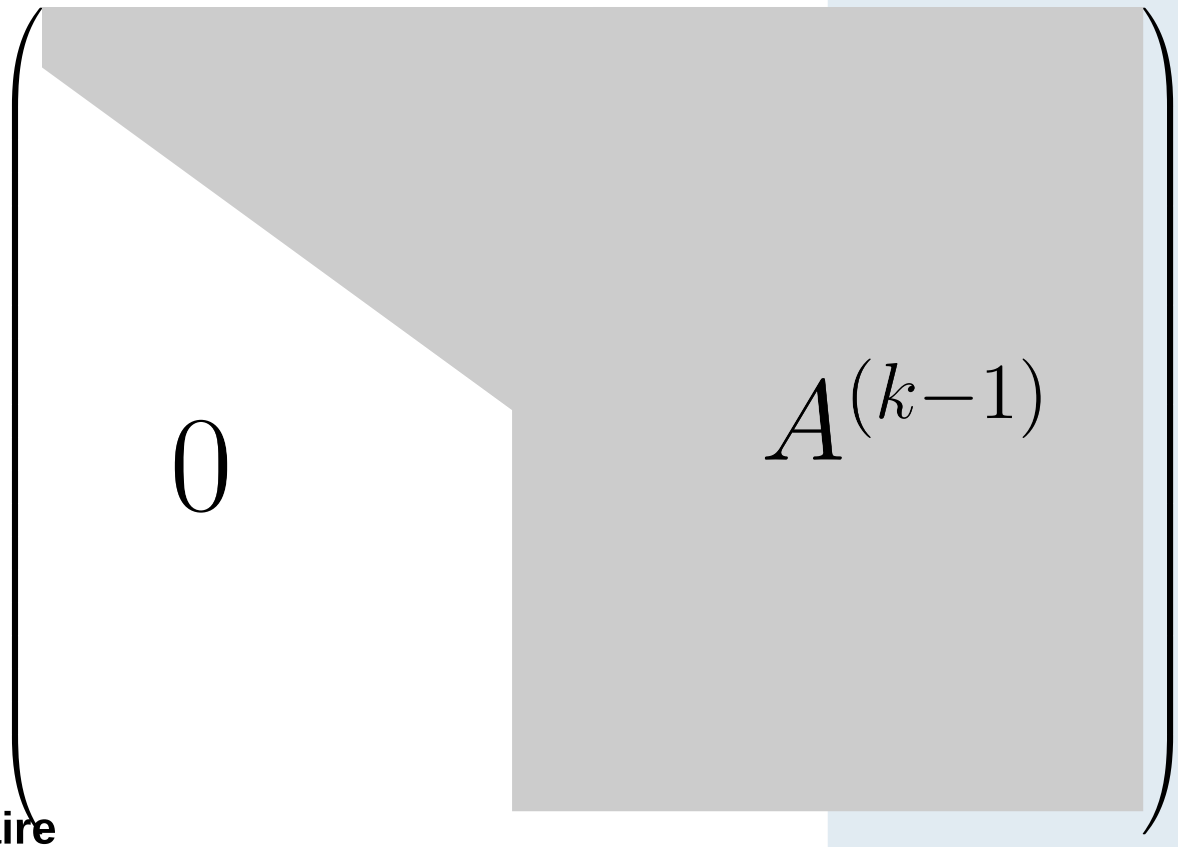
Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \dots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \dots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \dots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*



Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

$$
\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & & & a_{kk} & a_{kk+1} & \cdots & a_{km} \\
 & 0 & & a_{k+1k} & a_{k+1k+1} & \cdots & a_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}
$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes
début

- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin
retourner A *la matrice triangulaire*

$$\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & 0 & & a_{kk} & a_{kk+1} & \cdots & a_{km} \\
 & & & a_{k+1k} & a_{k+1k+1} & \cdots & a_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}$$

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

$$
\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & & & p & a_{kk+1} & \cdots & a_{km} \\
 & 0 & & a_{k+1k} & a_{k+1k+1} & \cdots & a_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}
$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

$$\begin{pmatrix} p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\ & \ddots & & & & & \\ & & p^{(k-1)} & & & & \\ & & & p & a_{kk+1} & \cdots & a_{km} \\ & 0 & & a_{k+1k} & a_{k+1k+1} & \cdots & a_{k+1m} \\ & & & \vdots & \vdots & \ddots & \vdots \\ & & & a_{nk} & a_{nk+1} & \cdots & a_{nm} \end{pmatrix}$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes
début
  pour $k = 1 \dots n$ **faire**
    $p \leftarrow A[k,k]$
    pour $i = k+1 \dots n$ **faire**
      $q \leftarrow A[i,k]$
      $A[i,k] \leftarrow 0$
      pour $j = k+1 \dots m$ **faire**
        $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$
fin
retourner A *la matrice triangulaire*

$$\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & & & p & a_{kk+1} & \cdots & a_{km} \\
 & 0 & & q & a_{k+1k+1} & \cdots & a_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \dots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \dots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \dots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

$$\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & & & p & a_{kk+1} & \cdots & a_{km} \\
 & 0 & & 0 & a_{k+1k+1} & \cdots & a_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes
début

  pour $k = 1 \ldots n$ **faire**
    $p \leftarrow A[k,k]$

    pour $i = k+1 \ldots n$ **faire**
      $q \leftarrow A[i,k]$
      $A[i,k] \leftarrow 0$

      pour $j = k+1 \ldots m$ **faire**
        $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin
retourner A *la matrice triangulaire*

$$\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & & & p & a_{kk+1} & \cdots & a_{km} \\
 & \mathbf{0} & & 0 & a'_{k+1k+1} & \cdots & a_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes
début
  pour $k = 1 \ldots n$ **faire**
    $p \leftarrow A[k,k]$
    pour $i = k+1 \ldots n$ **faire**
      $q \leftarrow A[i,k]$
      $A[i,k] \leftarrow 0$
      pour $j = k+1 \ldots m$ **faire**
        $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$
fin
retourner A *la matrice triangulaire*

$$
\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & p^{(k-1)} & & & & \\
 & 0 & & p & a_{kk+1} & \cdots & a_{km} \\
 & & & 0 & a'_{k+1k+1} & \cdots & a'_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & a_{nk} & a_{nk+1} & \cdots & a_{nm}
\end{pmatrix}
$$

Algorithme de triangularisation

Données : $A = (A[i,j])$,
n le nb de lignes,
m le nb de colonnes

début

- **pour** $k = 1 \dots n$ **faire**
  - $p \leftarrow A[k,k]$
  - **pour** $i = k+1 \dots n$ **faire**
    - $q \leftarrow A[i,k]$
    - $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \dots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

$$\begin{pmatrix}
p^{(1)} & a_{12} & \cdots & a_{1k} & \cdots & & a_{1m} \\
 & \ddots & & & & & \\
 & & \ddots & & & & \\
 & & & p^{(k-1)} & & & \\
 & & & p & a_{kk+1} & \cdots & a_{km} \\
 & \text{0} & & 0 & a'_{k+1k+1} & \cdots & a'_{k+1m} \\
 & & & \vdots & \vdots & \ddots & \vdots \\
 & & & 0 & a'_{nk+1} & \cdots & a'_{nm}
\end{pmatrix}$$

Élimination de GAUSS

- Pour résoudre l'équation $Ax = b$ (n équations, n inconnues).

Élimination de GAUSS

- Pour résoudre l'équation $Ax = b$ (n équations, n inconnues).
- Construire la matrice $[Ab]$ (n colonnes $n+1$ lignes).

Élimination de GAUSS

- Pour résoudre l'équation $Ax = b$ (n équations, n inconnues).
- Construire la matrice $[Ab]$ (n colonnes $n+1$ lignes).
- Triangulariser la matrice.

Élimination de GAUSS

- Pour résoudre l'équation $Ax = b$ (n équations, n inconnues).
- Construire la matrice $[Ab]$ (n colonnes $n+1$ lignes).
- Triangulariser la matrice.
- Appliquer l'algorithme de remontée

Élimination de GAUSS

- Pour résoudre l'équation $Ax = b$ (n équations, n inconnues).
- Construire la matrice $[Ab]$ (n colonnes $n+1$ lignes).
- Triangulariser la matrice.
- Appliquer l'algorithme de remontée
- $2\frac{n^3}{3}$ opérations pour la triangularisation, n^2 pour la remontée.

Élimination de Gauss

- Pour résoudre l'équation $Ax = b$ (n équations, n inconnues).
- Construire la matrice $[Ab]$ (n colonnes $n+1$ lignes).
- Triangulariser la matrice.
- Appliquer l'algorithme de remontée
- $2\frac{n^3}{3}$ opérations pour la triangularisation, n^2 pour la remontée.

Par exemple :

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix} \cdot x = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$$

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & \\ 1 & 1 & 2 & \\ 1 & 1 & 1 & \end{pmatrix}$$

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

Exemple

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Exemple

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

remontée

Exemple

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

remontée

$$\begin{aligned} x_3 &= \frac{1}{-1} \\ &= -1 \end{aligned}$$

Exemple

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

remontée

$$\begin{aligned} x_3 &= \frac{1}{-1} \\ &= -1 \\ x_2 &= \frac{1-(-1\times(-1))}{-1} \\ &= 0 \end{aligned}$$

Exemple

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

remontée

$$\begin{aligned} x_3 &= \frac{1}{-1} \\ &= -1 \\ x_2 &= \frac{1-(-1\times(-1))}{-1} \\ &= 0 \\ x_1 &= \frac{4-(3\times(-1)+2\times 0)}{1} \\ &= 7 \end{aligned}$$

Exemple

triangularisation

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 1 & 2 & 5 \\ 1 & 1 & 1 & 6 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & -1 & -2 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -1 & -1 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

remontée

$$\begin{aligned} x_3 &= \frac{1}{-1} \\ &= -1 \\ x_2 &= \frac{1-(-1\times(-1))}{-1} \\ &= 0 \\ x_1 &= \frac{4-(3\times(-1)+2\times 0)}{1} \\ &= 7 \end{aligned}$$

Donc

$$x = \begin{pmatrix} 7 \\ 0 \\ -1 \end{pmatrix}$$

Forme matricielle de la triangularisation

Décomposition LU

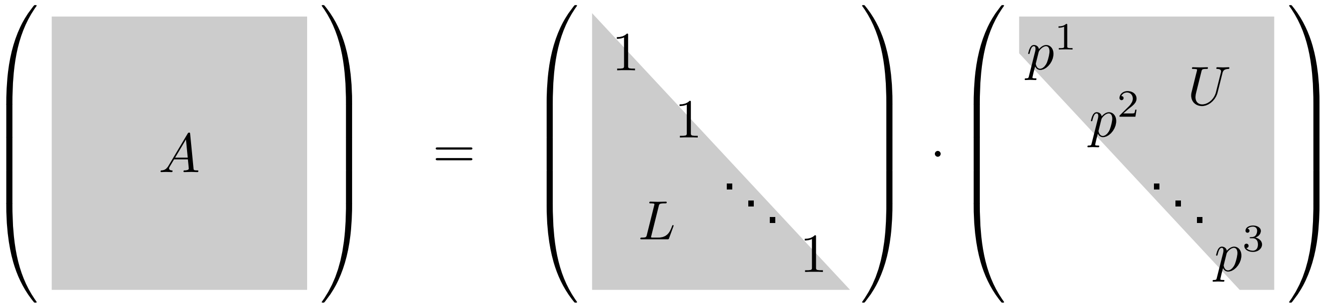
Pour « préparer la matrice » on souhaite la factoriser en deux matrices triangulaires :

$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} = \begin{pmatrix} 1 & & & \\ & 1 & & \\ L & & \ddots & \\ & & & 1 \end{pmatrix} \cdot \begin{pmatrix} p^1 & & U \\ & p^2 & \\ & & \ddots & \\ & & & p^3 \end{pmatrix}$$

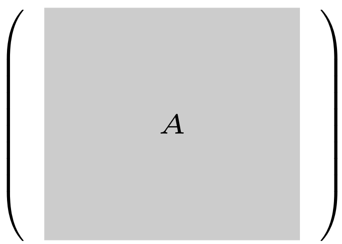
Pour construire L et U on utilise l'élimination de GAUSS en « se souvenant » des opérations faites.

Décomposition LU

Pour « préparer la matrice » on souhaite la factoriser en deux matrices triangulaires :

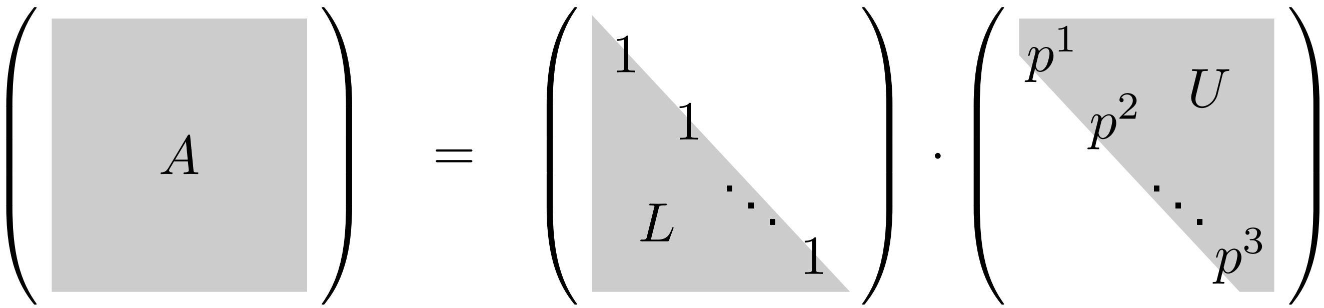


Pour construire L et U on utilise l'élimination de GAUSS en « se souvenant » des opérations faites.
En effet à la fin de la triangularisation, on obtient U :

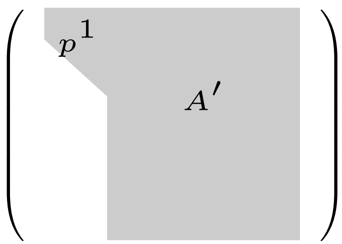


Décomposition LU

Pour « préparer la matrice » on souhaite la factoriser en deux matrices triangulaires :

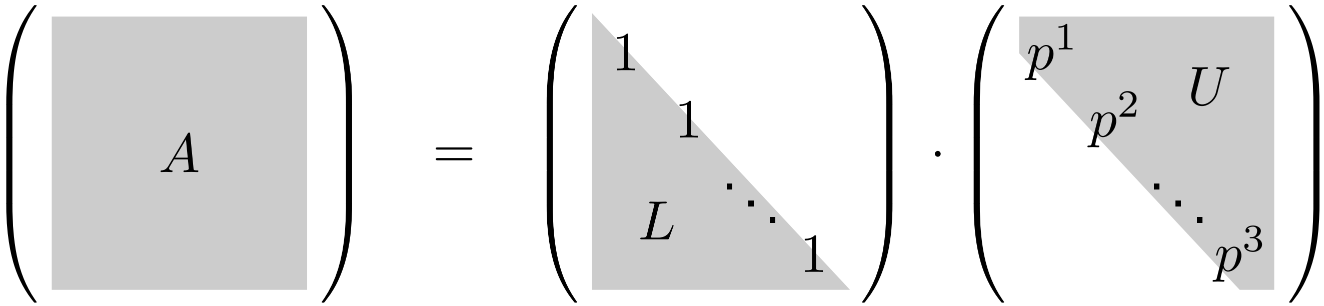


Pour construire L et U on utilise l'élimination de GAUSS en « se souvenant » des opérations faites.
En effet à la fin de la triangularisation, on obtient U :

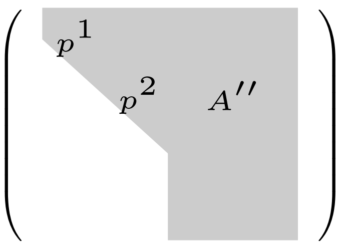


Décomposition LU

Pour « préparer la matrice » on souhaite la factoriser en deux matrices triangulaires :



Pour construire L et U on utilise l'élimination de GAUSS en « se souvenant » des opérations faites.
En effet à la fin de la triangularisation, on obtient U :



Décomposition LU

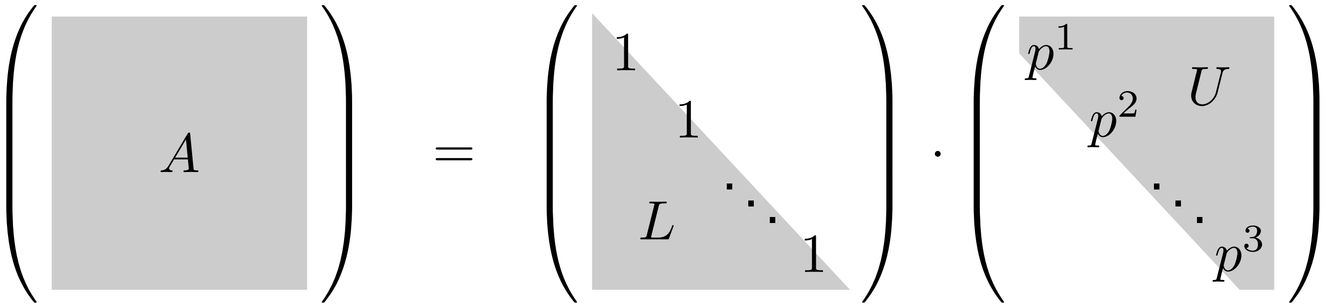
Pour « préparer la matrice » on souhaite la factoriser en deux matrices triangulaires :

$$\begin{pmatrix} & & \\ & A & \\ & & \end{pmatrix} = \begin{pmatrix} 1 & & \\ & 1 & \\ L & & \ddots & \\ & & & 1 \end{pmatrix} \cdot \begin{pmatrix} p^1 & & U \\ & p^2 & \\ & & \ddots & \\ & & & p^3 \end{pmatrix}$$

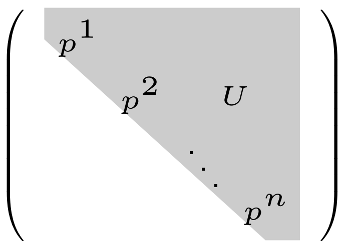
Pour construire L et U on utilise l'élimination de GAUSS en « se souvenant » des opérations faites.
En effet à la fin de la triangularisation, on obtient U :

$$\begin{pmatrix} p^1 & & U \\ & p^2 & \\ & & \ddots & \\ & & & p^n \end{pmatrix}$$

Pour « préparer la matrice » on souhaite la factoriser en deux matrices triangulaires :



Pour construire L et U on utilise l'élimination de GAUSS en « se souvenant » des opérations faites.
En effet à la fin de la triangularisation, on obtient U :



Une étape de l'élimination revient à multiplier A par une matrice $M^{(k)}$ quelle est la forme de cette matrice ?

Soient

$$m_i^k = -\frac{a_{ik}}{a_{kk}} \quad \text{et} \quad M^{(k)} = \begin{array}{c} \hspace{2em} k^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1}^k & & \\ & & \vdots & \ddots & \\ & & m_n^k & & 1 \end{pmatrix} \end{array}$$

Soient

$$m_i^k = -\frac{a_{ik}}{a_{kk}} \quad \text{et} \quad M^{(k)} = \begin{array}{c} k^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1}^k & & \\ & & \vdots & \ddots & \\ & & m_n^k & & 1 \end{pmatrix} \end{array}$$

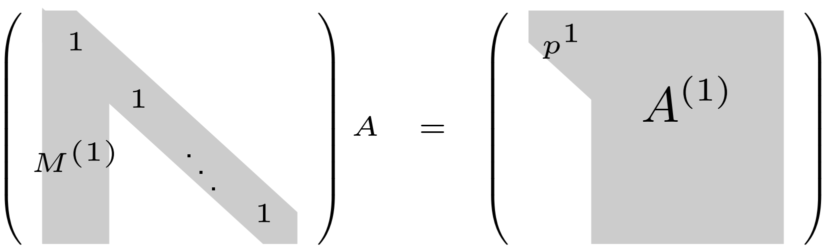
Alors :

$$A = \begin{pmatrix} A \end{pmatrix}$$

Soient

$$m_i^k = -\frac{a_{ik}}{a_{kk}} \qquad \text{et} \qquad M^{(k)} = \begin{array}{c} \quad k^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1}^k & & \\ & & \vdots & \ddots & \\ & & m_n^k & & 1 \end{pmatrix} \end{array}$$

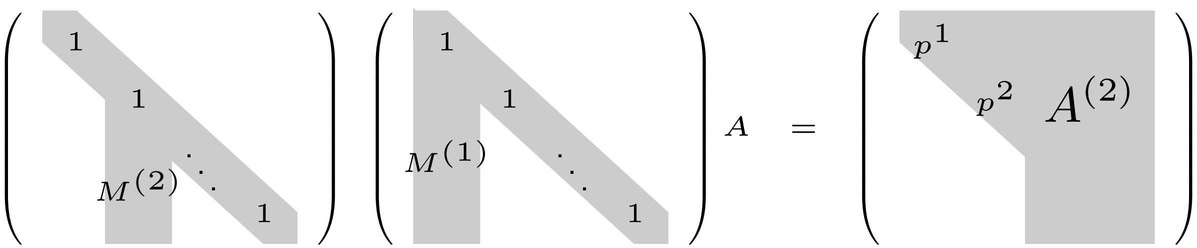
Alors :



Soient

$$m_i^k = -\frac{a_{ik}}{a_{kk}} \qquad \text{et} \qquad M^{(k)} = \begin{array}{c} k^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1}^k & \ddots & \\ & & \vdots & & \\ & & m_n^k & & 1 \end{pmatrix} \end{array}$$

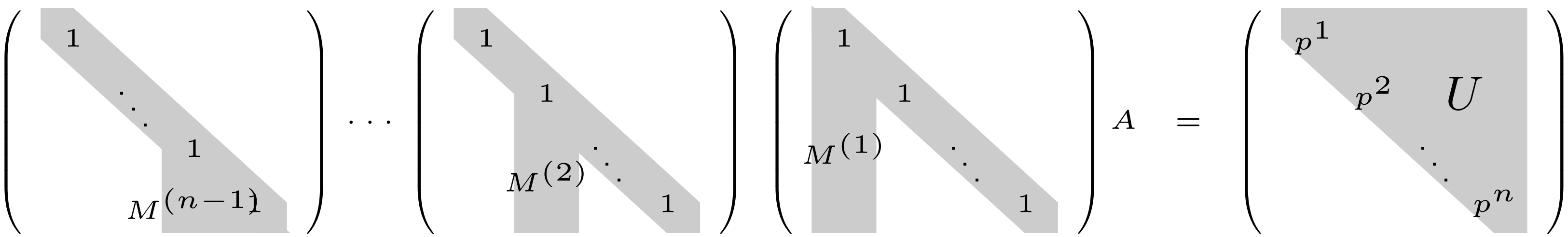
Alors :



Soient

$$m_i^k = -\frac{a_{ik}}{a_{kk}} \qquad \text{et} \qquad M^{(k)} = \begin{array}{c} k^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1}^k & \ddots & \\ & & \vdots & & \\ & & m_n^k & & 1 \end{pmatrix} \end{array}$$

Alors :



Soient

$$m_i^k = -\frac{a_{ik}}{a_{kk}} \qquad \text{et} \qquad M^{(k)} = \begin{array}{c} k^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m^k_{k+1} & \ddots & \\ & & \vdots & & \\ & & m^k_n & & 1 \end{pmatrix} \end{array}$$

Alors :

$$\begin{pmatrix} 1 & & \\ & \ddots & \\ & & 1 \\ & M^{(n-1)} & & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & & \\ & 1 & \\ M^{(2)} & & \ddots \\ & & & 1 \end{pmatrix} \begin{pmatrix} 1 & & \\ & 1 & \\ M^{(1)} & & \ddots \\ & & & 1 \end{pmatrix} A = \begin{pmatrix} p^1 & & \\ & p^2 & U \\ & & \ddots \\ & & & p^n \end{pmatrix}$$

Nous avons donc

$$M \cdot A = U$$

Calcul de la matrice L

A-t-on obtenu les matrices U et L de la décomposition ?

Calcul de la matrice L

A-t-on obtenu les matrices U et L de la décomposition ?

- le produit de deux matrices triangulaires inférieures est triangulaire inférieure,
- L'inverse d'une matrice triangulaire inférieure est une matrice triangulaire inférieure.
- Lorsqu'elle existe la décomposition est unique.

Cela nous prouve que la matrice U obtenue est celle de la décomposition LU et que la matrice M est l'inverse de la matrice L recherchée.

Calcul de la matrice L

A-t-on obtenu les matrices U et L de la décomposition ?

- le produit de deux matrices triangulaires inférieures est triangulaire inférieure,
- L'inverse d'une matrice triangulaire inférieure est une matrice triangulaire inférieure.
- Lorsqu'elle existe la décomposition est unique.

Cela nous prouve que la matrice U obtenue est celle de la décomposition LU et que la matrice M est l'inverse de la matrice L recherchée.

Pour calculer la matrice L il faut :

Calcul de la matrice L

A-t-on obtenu les matrices U et L de la décomposition ?

- le produit de deux matrices triangulaires inférieures est triangulaire inférieure,
- L'inverse d'une matrice triangulaire inférieure est une matrice triangulaire inférieure.
- Lorsqu'elle existe la décomposition est unique.

Cela nous prouve que la matrice U obtenue est celle de la décomposition LU et que la matrice M est l'inverse de la matrice L recherchée.

Pour calculer la matrice L il faut :

- ▷ Inverser les $M^{(i)}$,
- ▷ Calculer le produit :

$$L = {M^{(1)}}^{-1} \cdot {M^{(2)}}^{-1} \cdot {M^{(3)}}^{-1} \cdots {M^{(n-1)}}^{-1}$$

Inverse de la matrice $M^{(i)}$

On peut montrer que :

$$
\begin{array}{c}
\downarrow \text{i}^{\text{e}} \text{ colonne} \\
\begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & X & \ddots & \\ & & & & 1 \end{pmatrix}
\end{array}
\times
\begin{array}{c}
\downarrow \text{j}^{\text{e}} \text{ colonne} \\
\begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & Y & \ddots & \\ & & & & 1 \end{pmatrix}
\end{array}
=
\left\{
\begin{array}{ll}
\begin{array}{c}
\downarrow \text{i}^{\text{e}} \text{ colonne} \\
\begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & X+Y & \ddots & \\ & & & & 1 \end{pmatrix}
\end{array} & \text{si } i = j \\
\begin{array}{c}
\begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & X & \ddots & & & \\ & & & & 1 & & \\ & & & & Y & \ddots & \\ & & & & & & 1 \end{pmatrix} \\
\text{i}^{\text{e}} \text{ colonne} \uparrow \qquad \uparrow \text{j}^{\text{e}} \text{ colonne}
\end{array} & \text{si } i < j
\end{array}
\right.
$$

Calcul de L

Donc $M^{(k)}$ est inversible d'inverse $L^{(k)}$ avec :

$$M^{(k)} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & m_{k+1}^k & & \\ & & & \ddots & \\ & & m_n^k & & 1 \end{pmatrix}$$

Donc $M^{(k)}$ est inversible d'inverse $L^{(k)}$ avec :

$$M^{(k)} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & m_{k+1}^k & & & \\ & & & & \ddots & \\ & & m_n^k & & & 1 \end{pmatrix}$$

$$L^{(k)} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -m_{k+1}^k & & & \\ & & & & \ddots & \\ & & -m_n^k & & & 1 \end{pmatrix}$$

Calcul de L (suite)

La matrice L de la décomposition est :

$$L = L^{(1)} \times L^{(2)} \dots \times L^{(n-1)}$$

$$=$$

Calcul de L (suite)

La matrice L de la décomposition est :

$$
\begin{aligned}
L &= L^{(1)} \times L^{(2)} \cdots \times L^{(n-1)} \\
&= \begin{pmatrix}
1 & & & & & \\
-m_2^1 & 1 & & & & \\
\cdot & \ddots & 1 & & & \\
\cdot & & -m_{k+1}^k & 1 & & \\
\cdot & & \vdots & \ddots & 1 & \\
-m_n^1 & \dots & -m_n^k & \dots & -m_n^{n-1} & 1
\end{pmatrix}
\end{aligned}
$$

Rappel : les m_i^k sont les coefficients de l'élimination de GAUSS

$$
-m_i^k = \frac{a_{ik}}{a_{kk}}
$$

Algorithme

Données : $A = (A[i,j])$, n le nombre de lignes et colonnes

début

pour $k = 1 \dots n$ **faire**

$p \leftarrow A[k,k]$

pour $i = k+1 \dots n$ **faire**

$q \leftarrow A[i,k]$

$A[i,k] \leftarrow 0$

pour $j = k+1 \dots n$ **faire**

$A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner

Algorithme

Données : $A = (A[i,j])$, n le nombre de lignes et colonnes

début

$U \leftarrow A$

pour $k = 1 \dots n$ **faire**

$p \leftarrow U[k,k]$

pour $i = k+1 \dots n$ **faire**

$q \leftarrow U[i,k]$

$U[i,k] \leftarrow 0$

pour $j = k+1 \dots n$ **faire**

$U[i,j] = U[i,j] - U[k,j].\frac{q}{p}$

fin

retourner U *la matrice triangulaire supérieure,*

Algorithme

Données : $A = (A[i,j])$, n le nombre de lignes et colonnes

début

$U \leftarrow A$
$L \leftarrow I$
pour $k = 1 \ldots n$ **faire**
  $p \leftarrow U[k,k]$
  pour $i = k+1 \ldots n$ **faire**
    $q \leftarrow U[i,k]$
    $U[i,k] \leftarrow 0$
    $L[i,k] \leftarrow \frac{q}{p}$
    pour $j = k+1 \ldots n$ **faire**
      $U[i,j] = U[i,j] - U[k,j].\frac{q}{p}$

fin

retourner U *la matrice triangulaire supérieure,*
L *la matrice triangulaire inférieure*

Décomposition de la matrice

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

$$\Big\downarrow$$

Décomposition de la matrice

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

$$U^{(0)} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

$$L^{(0)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Décomposition de la matrice

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

$$U^{(0)} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix} \qquad L^{(0)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$U^{(1)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -1 \\ 0 & -1 & -2 \end{pmatrix} \qquad L^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

Décomposition de la matrice

$$\begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix}$$

$$U^{(0)} = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 1 & 2 \\ 1 & 1 & 1 \end{pmatrix} \qquad L^{(0)} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$U^{(1)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -1 \\ 0 & -1 & -2 \end{pmatrix} \qquad L^{(1)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$$

$$U^{(2)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -1 \\ 0 & 0 & -1 \end{pmatrix} \qquad L^{(2)} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Conditions

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1 \le i,j \le n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k \quad = \quad (a_{ij})_{1 \le i,j \le \mathbf{k}} \qquad \text{sont inversibles}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1 \le i,j \le n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k = (a_{ij})_{1 \le i,j \le \mathbf{k}} \qquad \textit{sont inversibles}$$

$$\begin{pmatrix} \boxed{A_1} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1\leq i,j\leq n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k \quad = \quad (a_{ij})_{1\leq i,j\leq \mathbf{k}} \qquad \text{sont inversibles}$$

$$\begin{pmatrix} \boxed{\begin{matrix} a_{11} & a_{12} \\ & A_2 \\ a_{21} & a_{22} \end{matrix}} & a_{13} & \cdots & a_{1n} \\ & a_{23} & \cdots & a_{2n} \\ a_{31} \quad a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots \quad \vdots & \vdots & \cdots & \vdots \\ a_{n1} \quad a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

Conditions

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1 \leq i,j \leq n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k = (a_{ij})_{1 \leq i,j \leq \mathbf{k}} \quad \textit{sont inversibles}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

A_3

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1 \le i,j \le n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k = (a_{ij})_{1 \le i,j \le k} \qquad \textit{sont inversibles}$$

$$\begin{pmatrix} \boxed{\begin{matrix} a_{11} & a_{12} & a_{13} & \cdots \\ a_{21} & a_{22} & a_{23} & \cdots \\ a_{31} & a_{32} & a_{33} & \cdots \\ \vdots & \vdots & \vdots & \cdots \end{matrix}}_{A_4} & \begin{matrix} a_{1n} \\ a_{2n} \\ a_{3n} \\ \vdots \end{matrix} \\ \begin{matrix} a_{n1} & a_{n2} & a_{n3} & \cdots \end{matrix} & a_{nn} \end{pmatrix}$$

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1 \leq i,j \leq n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k \quad = \quad (a_{ij})_{1 \leq i,j \leq \mathbf{k}} \qquad \textit{sont inversibles}$$

$$A_n = A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

Cet algorithme est applicable sur A *ssi tous les pivots* $p^{(k)}$ *sont non nuls*.

Comment être sur que ces pivots ne seront pas nuls ?

Théorème *L'élimination de* GAUSS *fonctionne sur une matrice* $A = (a_{ij})_{1 \leq i,j \leq n}$ *si et seulement si toutes ses matrices principales (« en coin »)*

$$A_k = (a_{ij})_{1 \leq i,j \leq k} \qquad \text{sont inversibles}$$

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{pmatrix}$$

Cela ne donne pas de moyen à priori pour savoir si la méthode fonctionne.

Pivots nuls

Le fait qu'une matrice principale A_k ne soit pas inversible *ne signifie pas* que la matrice A n'est pas inversible.

Pivots nuls

Le fait qu'une matrice principale A_k ne soit pas inversible *ne signifie pas* que la matrice A n'est pas inversible.

Par exemple :

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

ces matrices sont inversibles et peuvent être triangularisées par une méthode un peu plus complexe.

Pivots nuls

Le fait qu'une matrice principale A_k ne soit pas inversible *ne signifie pas* que la matrice A n'est pas inversible.

Par exemple :

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

ces matrices sont inversibles et peuvent être triangularisées par une méthode un peu plus complexe.

L'avis du mathématicien :

« si un pivot est nul, alors on permute deux lignes »

Pivots nuls

Le fait qu'une matrice principale A_k ne soit pas inversible *ne signifie pas* que la matrice A n'est pas inversible.

Par exemple :

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

ces matrices sont inversibles et peuvent être triangularisées par une méthode un peu plus complexe.

L'avis du mathématicien :

« si un pivot est nul, alors on permute deux lignes »
car :

Théorème *Si tous les pivots possibles sont nuls alors la matrice est singulière*

Pivots nuls

Le fait qu'une matrice principale A_k ne soit pas inversible *ne signifie pas* que la matrice A n'est pas inversible.

Par exemple :

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

ces matrices sont inversibles et peuvent être triangularisées par une méthode un peu plus complexe.

L'avis du mathématicien :

« si un pivot est nul, alors on permute deux lignes »
car :

Théorème *Si tous les pivots possibles sont nuls alors la matrice est singulière*

Pourquoi cela n'est-il pas satisfaisant ?

En modifiant légèrement la matrice précédente :

$$\begin{pmatrix} 1 & 1 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x \;=\; \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

La solution est $(-1, 1, 1)$*.*

Appliquons la méthode de triangularisation avec 4 chiffres décimaux de précision en arrondi au plus près.

Exemple

En modifiant légèrement la matrice précédente :

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x \quad = \quad \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

La solution est $(-1, 1, 1)$*.*

Appliquons la méthode de triangularisation avec 4 chiffres décimaux de précision en arrondi au plus près.

Exemple

En modifiant légèrement la matrice précédente :

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

La solution est $(-1, 1, 1)$.

Appliquons la méthode de triangularisation avec 4 chiffres décimaux de précision en arrondi au plus près. La matrice A devient :

$$A = \begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

En modifiant légèrement la matrice précédente :

$$\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

La solution est $(-1, 1, 1)$.

Appliquons la méthode de triangularisation avec 4 chiffres décimaux de précision en arrondi au plus près. La matrice A devient :

$$A = \begin{pmatrix} 0.3333 & 0.3333 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$

Exemple (suite)

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ -1 & -1 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

Exemple (suite)

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1+0.9999 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1E-4 & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

Exemple (suite)

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1E-4 & 1 \\ 0 & 0 & 1+10000 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2+10000 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1E-4 & 1 \\ 0 & 0 & \mathbf{1E4} \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ \mathbf{1E4} \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1E-4 & 1 \\ 0 & 0 & \mathbf{1E4} \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ \mathbf{1E4} \end{pmatrix}$$

Donc

$$\begin{aligned} x_3 &= \frac{1E4}{1E4} &= 1 \\ x_2 &= \frac{1-1}{1E-4} &= 0 \\ x_1 &= \frac{0}{0.3333} &= 0 \end{aligned}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1E-4 & 1 \\ 0 & 0 & \mathbf{1E4} \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ \mathbf{1E4} \end{pmatrix}$$

Donc

$$\begin{aligned} x_3 &= \frac{1E4}{1E4} &= 1 \\ x_2 &= \frac{1-1}{1E-4} &= 0 \\ x_1 &= \frac{0}{0.3333} &= 0 \end{aligned}$$

Suite aux *erreurs d'arrondis*,
la solution fournie est $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ au lieu de $\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$

Exemple (suite)

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & \mathbf{-1E-4} & 1 \\ 0 & 1 & 1 \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ 2 \end{pmatrix}$$

$$\begin{pmatrix} 0.3333 & 0.3333 & 0 \\ 0 & -1E-4 & 1 \\ 0 & 0 & \mathbf{1E4} \end{pmatrix} \times x = \begin{pmatrix} 0 \\ 1 \\ \mathbf{1E4} \end{pmatrix}$$

Donc

$$\begin{aligned} x_3 &= \frac{1E4}{1E4} &= 1 \\ x_2 &= \frac{1-1}{1E-4} &= 0 \\ x_1 &= \frac{0}{0.3333} &= 0 \end{aligned}$$

Suite aux *erreurs d'arrondis*, la solution fournie est $\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ au lieu de $\begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}$

Pour résoudre ce problème on recherche les pivots maximaux.

Recherche de pivots maximaux

Recherche de pivots maximaux

Nous n'avons pas utilisé les deux propriétés suivantes :

- On ne change pas la solution du système lorsque on permute deux lignes,
- On ne change pas la solution du système lorsque on permute deux colonnes.

Recherche de pivots maximaux

Nous n'avons pas utilisé les deux propriétés suivantes :

- On ne change pas la solution du système lorsque on permute deux lignes,
- On ne change pas la solution du système lorsque on permute deux colonnes.

Or la permutation est une opération qui ne cause *aucune erreur de calcul*.

Recherche de pivots maximaux

Nous n'avons pas utilisé les deux propriétés suivantes :

- On ne change pas la solution du système lorsque on permute deux lignes,
- On ne change pas la solution du système lorsque on permute deux colonnes.

Or la permutation est une opération qui ne cause *aucune erreur de calcul*.

On peut utiliser cette propriété pour *choisir le pivot le plus grand* (en valeur absolue).

- En ne permutant que les lignes, c'est l'algorithme de GAUSS *avec pivot partiel*. ▷*simple*

Recherche de pivots maximaux

Nous n'avons pas utilisé les deux propriétés suivantes :

- On ne change pas la solution du système lorsque on permute deux lignes,
- On ne change pas la solution du système lorsque on permute deux colonnes.

Or la permutation est une opération qui ne cause *aucune erreur de calcul*.

On peut utiliser cette propriété pour *choisir le pivot le plus grand* (en valeur absolue).

- En ne permutant que les lignes, c'est l'algorithme de GAUSS *avec pivot partiel*. ▷*simple*
- En permutant les lignes et les colonnes, c'est l'algorithme de GAUSS *avec pivot total*. ▷*stable*

Pivot partiel

- rechercher le pivot maximal parmi les éléments de la colonne k situés sous la diagonale.
  $p^k = a_{s,k}$ avec $|a_{s,k}| = \max_{i=k\dots n} |a_{i,k}|$
- permuter les lignes s et k de la matrice $A^{(k-1)}$, ce qui revient uniquement à changer l'ordre des équations.
- Si tous les pivots de cette colonne sont nuls, la matrice est singulière.
- « Si tous les pivots de cette colonne sont proches de 0, la matrice est soit singulière mais mal calculée, soit proche d'une matrice singulière donc instable ».

$$\begin{pmatrix}
p^1 & a_{12} & \cdots & a_{1k} & \cdot \quad \cdot \quad \cdot & a_{1m} \\
 & \ddots & & & & \\
 & & p^{k-1} & & & \\
 & 0 & & \boxed{a_{kk}} & a_{kk+1} \quad \cdots & a_{km} \\
 & & & \boxed{\vdots} & \vdots & \vdots \\
 & & & \boxed{a_{nk}} & a_{nk+1} \quad \cdots & a_{nm}
\end{pmatrix}$$

Élimination avec pivot partiel

début

  pour $k = 1 \dots n$ **faire**

    $p \leftarrow A[k, k]$;

    pour $i = k + 1 \dots n$ **faire**

      $q \leftarrow A[i, k]$; $A[i, k] \leftarrow 0$

      pour $j = k + 1 \dots m$ **faire**

        $A[i, j] = A[i, j] - A[k, j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

Élimination avec pivot partiel

début
- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$; $l \leftarrow k$
  - **pour** $i = k \ldots n$ **faire**
    - **si** $|A[i,k]| > p$ **alors**
      - $p \leftarrow A[i,k]$; $l \leftarrow i$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$; $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

Élimination avec pivot partiel

début
- **pour** $k = 1 \ldots n$ **faire**
  - $p \leftarrow A[k,k]$; $l \leftarrow k$
  - **pour** $i = k \ldots n$ **faire**
    - **si** $|A[i,k]| > p$ **alors**
      - $p \leftarrow A[i,k]$; $l \leftarrow i$
  - **si** $l \neq k$ **alors**
    - **pour** $j = k \ldots m$ **faire**
      - $temp \leftarrow A[k,j]$; $A[k,j] \leftarrow A[l,j]$; $A[l,j] \leftarrow temp$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow A[i,k]$; $A[i,k] \leftarrow 0$
    - **pour** $j = k+1 \ldots m$ **faire**
      - $A[i,j] = A[i,j] - A[k,j].\frac{q}{p}$

fin

retourner A *la matrice triangulaire*

Pivot total

- rechercher le pivot maximal parmi les éléments de la sous-matrice $[a_{i,j}] \quad (i \geq k,\ j \geq k)$.
  $p^k = a_{s,t}$ avec $|a_{s,t}| = \max_{i=k\ldots n,\ j=k\ldots n} |a_{i,j}|$
- permuter les lignes s et k
- permuter les colonnes t et k, ce qui modifie l'ordre des inconnues

$$\left(\begin{array}{cccccc}
p^1 & a_{12} & \cdots & a_{1k} & \cdot\ \cdot\ \cdot & a_{1m} \\
 & \ddots & & & & \\
 & & p^{k-1} & & & \\
 & 0 & & \multicolumn{3}{c}{\boxed{\begin{array}{cccc} a_{kk} & a_{kk+1} & \cdots & a_{km} \\ \vdots & \vdots & & \vdots \\ a_{nk} & a_{nk+1} & \cdots & a_{nm} \end{array}}}
\end{array}\right)$$

Résumé

Effet sur la décomposition LU

- Est ce que l'inversion de lignes ou de colonnes est possible lorsqu'on veut la décomposition LU ?

Effet sur la décomposition LU

- Est ce que l'inversion de lignes ou de colonnes est possible lorsqu'on veut la décomposition LU ?

- Matriciellement, à quoi correspond ces permutations ?

Effet sur la décomposition LU

- Est ce que l'inversion de lignes ou de colonnes est possible lorsqu'on veut la décomposition LU ?
- Matriciellement, à quoi correspond ces permutations ?
- Peut-on appliquer les deux formes de recherche de pivot maximaux à l'algorithme de décomposition LU ?

Matrice de permutation

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$
\begin{array}{cc}
 & \begin{array}{ccccc} \phantom{\begin{smallmatrix}1&&\\&\ddots&\\&&1\end{smallmatrix}} & \text{i}^{\text{e}}\text{ colonne}\downarrow & \phantom{\begin{smallmatrix}1&&\\&\ddots&\\&&1\end{smallmatrix}} & \downarrow\text{j}^{\text{e}}\text{ colonne} & \phantom{\begin{smallmatrix}1&&\\&\ddots&\\&&1\end{smallmatrix}} \end{array} \\
P_{ij} = & \begin{pmatrix}
\begin{smallmatrix}1&&\\&\ddots&\\&&1\end{smallmatrix} & & & & \\
 & 0 & & 1 & \\
 & & \begin{smallmatrix}1&&\\&\ddots&\\&&1\end{smallmatrix} & & \\
 & 1 & & 0 & \\
 & & & & \begin{smallmatrix}1&&\\&\ddots&\\&&1\end{smallmatrix}
\end{pmatrix}
\end{array}
$$

Matrice de permutation

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$P_{ij} \quad = \quad \begin{array}{c} \begin{array}{ccccc} \phantom{\begin{matrix}1&&\\&\ddots&\\&&1\end{matrix}} & \text{i}^{\text{e}}\text{ colonne}\downarrow & \phantom{\begin{matrix}1&&\\&\ddots&\\&&1\end{matrix}} & \downarrow\text{j}^{\text{e}}\text{ colonne} & \phantom{\begin{matrix}1&&\\&\ddots&\\&&1\end{matrix}} \end{array} \\ \begin{pmatrix} \begin{matrix}1&&\\&\ddots&\\&&1\end{matrix} & & & & \\ & 0 & & 1 & \\ & & \begin{matrix}1&&\\&\ddots&\\&&1\end{matrix} & & \\ & 1 & & 0 & \\ & & & & \begin{matrix}1&&\\&\ddots&\\&&1\end{matrix} \end{pmatrix} \end{array}$$

Alors, l'élimination de GAUSS avec permutation revient à obtenir :

$$A \quad = \quad A$$

Matrice de permutation

Permuter deux lignes d’une matrice correspond la multiplier à gauche par une matrice de la forme :

$$
P_{ij} = \begin{array}{c}
\begin{array}{ccccc} & \text{i}^{\text{e}}\text{ colonne}\downarrow & & \downarrow\text{j}^{\text{e}}\text{ colonne} & \end{array} \\
\begin{pmatrix}
\begin{smallmatrix}1 & & \\ & \ddots & \\ & & 1\end{smallmatrix} & & & & \\
 & 0 & & 1 & \\
 & & \begin{smallmatrix}1 & & \\ & \ddots & \\ & & 1\end{smallmatrix} & & \\
 & 1 & & 0 & \\
 & & & & \begin{smallmatrix}1 & & \\ & \ddots & \\ & & 1\end{smallmatrix}
\end{pmatrix}
\end{array}
$$

Alors, l’élimination de GAUSS avec permutation revient à obtenir :

$$M^{(1)} P_{1,l_1} \cdot A = A^1$$

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$
P_{ij} \quad = \quad \begin{array}{c}
\qquad\qquad\quad \text{i}^{\text{e}} \text{ colonne} \downarrow \qquad \downarrow \text{j}^{\text{e}} \text{ colonne} \\
\begin{pmatrix}
1 & & & & & & & & \\
& \ddots & & & & & & & \\
& & 1 & & & & & & \\
& & & 0 & & & 1 & & \\
& & & & 1 & & & & \\
& & & & & \ddots & & & \\
& & & & & & 1 & & \\
& & & 1 & & & 0 & & \\
& & & & & & & 1 & \\
& & & & & & & & \ddots \\
& & & & & & & & & 1
\end{pmatrix}
\end{array}
$$

Alors, l'élimination de GAUSS avec permutation revient à obtenir :

$$
M^{(2)} P_{2,l_2} \cdot M^{(1)} P_{1,l_1} \cdot A \quad = \quad A^2
$$

Matrice de permutation

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$
\begin{array}{cc}
 & \begin{array}{ccc} \text{i}^{\text{e}}\text{ colonne} \downarrow & & \downarrow \text{j}^{\text{e}}\text{ colonne} \end{array} \\
P_{ij} = & \begin{pmatrix}
1 & & & & & & & \\
 & \ddots & & & & & & \\
 & & 1 & & & & & \\
 & & & 0 & & 1 & & \\
 & & & & 1 & & & \\
 & & & & & \ddots & & \\
 & & & & & 1 & & \\
 & & & 1 & & 0 & & \\
 & & & & & & 1 & \\
 & & & & & & & \ddots \\
 & & & & & & & & 1
\end{pmatrix}
\end{array}
$$

Alors, l'élimination de Gauss avec permutation revient à obtenir :

$$M^{(n-2)} P_{n-2,l_{n-2}} \cdots M^{(2)} P_{2,l_2} \cdot M^{(1)} P_{1,l_1} \cdot A = A^{n-2}$$

Matrice de permutation

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$P_{ij} = \begin{array}{c} \begin{array}{ccccc} & i^{e}\text{ colonne}\downarrow & & \downarrow j^{e}\text{ colonne} & \end{array} \\ \begin{pmatrix} 1 \ddots 1 & & & & \\ & 0 & & 1 & \\ & & 1 \ddots 1 & & \\ & 1 & & 0 & \\ & & & & 1 \ddots 1 \end{pmatrix} \end{array}$$

Alors, l'élimination de GAUSS avec permutation revient à obtenir :

$$M^{(n-1)} P_{n-1,l_{n-1}} \cdot M^{(n-2)} P_{n-2,l_{n-2}} \cdots M^{(2)} P_{2,l_2} \cdot M^{(1)} P_{1,l_1} \cdot A = U$$

Matrice de permutation

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$P_{ij} = \begin{array}{c} \qquad\qquad \text{i}^{\text{e}} \text{ colonne} \downarrow \qquad \downarrow \text{j}^{\text{e}} \text{ colonne} \\ \begin{pmatrix} 1 & & & & & & & & \\ & \ddots & & & & & & & \\ & & 1 & & & & & & \\ & & & 0 & & & 1 & & \\ & & & & 1 & & & & \\ & & & & & \ddots & & & \\ & & & & & & 1 & & \\ & & & 1 & & & 0 & & \\ & & & & & & & 1 & \\ & & & & & & & & \ddots \\ & & & & & & & & & 1 \end{pmatrix} \end{array}$$

Alors, l'élimination de GAUSS avec permutation revient à obtenir :

$$M^{(n-1)} P_{n-1,l_{n-1}} \cdot M^{(n-2)} P_{n-2,l_{n-2}} \cdots M^{(2)} P_{2,l_2} \cdot M^{(1)} P_{1,l_1} \cdot A = U$$

La matrice M obtenue n'est plus triangulaire inférieure

Matrice de permutation

Permuter deux lignes d'une matrice correspond la multiplier à gauche par une matrice de la forme :

$$
\begin{array}{cc}
 & \begin{array}{ccccc} & i^e \text{ colonne} \downarrow & & \downarrow j^e \text{ colonne} & \end{array} \\
P_{ij} = & \begin{pmatrix}
1 \ddots 1 & & & & \\
 & 0 & & 1 & \\
 & & 1 \ddots 1 & & \\
 & 1 & & 0 & \\
 & & & & 1 \ddots 1
\end{pmatrix}
\end{array}
$$

Alors, l'élimination de GAUSS avec permutation revient à obtenir :

$$M^{(n-1)} P_{n-1,l_{n-1}} \cdot M^{(n-2)} P_{n-2,l_{n-2}} \cdots M^{(2)} P_{2,l_2} \cdot M^{(1)} P_{1,l_1} \cdot A = U$$

La matrice M obtenue n'est plus triangulaire inférieure

Propriétés des matrices de permutation

Par contre, il est possible de commuter les matrices de permutation et les matrices $M^{(k)}$.
Si $k < i < j$:

$$P_{ij} \times M^{(k)} = \bar{M}^{(k)} \times P_{ij}$$

où $\bar{M}^{(k)}$ est la matrice $M^{(k)}$ dont les coefficients i et j ont été échangés.

Propriétés des matrices de permutation

Par contre, il est possible de commuter les matrices de permutation et les matrices $M^{(k)}$.
Si $k < i < j$:

$$P_{ij} \times M^{(k)} \quad = \quad \bar{M}^{(k)} \times P_{ij}$$

où $\bar{M}^{(k)}$ est la matrice $M^{(k)}$ dont les coefficients i et j ont été échangés.
$\bar{M}^{(k)}$ est de la même forme que $M^{(k)}$ donc elle est triangulaire inférieure et elle est inversée de la même façon

Propriétés des matrices de permutation

Par contre, il est possible de commuter les matrices de permutation et les matrices $M^{(k)}$.
Si $k < i < j$:

$$P_{ij} \times M^{(k)} \quad = \quad \bar{M}^{(k)} \times P_{ij}$$

où $\bar{M}^{(k)}$ est la matrice $M^{(k)}$ dont les coefficients i et j ont été échangés.
$\bar{M}^{(k)}$ est de la même forme que $M^{(k)}$ donc elle est triangulaire inférieure et elle est inversée de la même façon
Finalement,

$$\hat{M}^{(n-1)} \dots \hat{M}^{(2)} \hat{M}^{(1)} \quad \cdot \quad P_{n-1,l_{n-1}} \cdots P_{2l_2} P_{1l_1} \quad \cdot \quad A \quad = \quad U$$

Propriétés des matrices de permutation

Par contre, il est possible de commuter les matrices de permutation et les matrices $M^{(k)}$.
Si $k < i < j$:

$$P_{ij} \times M^{(k)} = \bar{M}^{(k)} \times P_{ij}$$

où $\bar{M}^{(k)}$ est la matrice $M^{(k)}$ dont les coefficients i et j ont été échangés.
$\bar{M}^{(k)}$ est de la même forme que $M^{(k)}$ donc elle est triangulaire inférieure et elle est inversée de la même façon
Finalement,

$$\underbrace{\hat{M}^{(n-1)} \dots \hat{M}^{(2)} \hat{M}^{(1)}}_{M} \cdot \underbrace{P_{n-1,l_{n-1}} \cdots P_{2l_2} P_{1l_1}}_{P} \cdot A = U$$

$$M \cdot P \cdot A = U$$

Propriétés des matrices de permutation

Par contre, il est possible de commuter les matrices de permutation et les matrices $M^{(k)}$.
Si $k < i < j$:

$$P_{ij} \times M^{(k)} \quad = \quad \bar{M}^{(k)} \times P_{ij}$$

où $\bar{M}^{(k)}$ est la matrice $M^{(k)}$ dont les coefficients i et j ont été échangés.
$\bar{M}^{(k)}$ est de la même forme que $M^{(k)}$ donc elle est triangulaire inférieure et elle est inversée de la même façon
Finalement,

$$\begin{array}{ccccccc} \hat{M}^{(n-1)} \ldots \hat{M}^{(2)} \hat{M}^{(1)} & \cdot & P_{n-1,l_{n-1}} \cdots P_{2l_2} P_{1l_1} & \cdot & A & = & U \\ M & \cdot & P & \cdot & A & = & U \\ & & & & P \cdot A & = & L \cdot U \end{array}$$

Décomposition PLU

- on applique l'algorithme avec recherche de pivot partiel, en calculant U et L
- A chaque fois qu'on permute une ligne de U, on permute aussi les lignes de L *en dessous de la diagonale*.
- On conserve la permutation P (l'ordre des lignes) car

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

Pour résoudre $Ax = b$, il suffit de résoudre $LUx = Pb$

! *on ne peut pas utiliser l'algorithme avec recherche du pivot total*

Algorithme

Données : $A = (A[i,j]), n$

début

  $U \leftarrow A$; ;

  pour $k = 1 \dots n$ **faire**

    // Recherche partiel du pivot

    $p \leftarrow U[k,k]$; $l \leftarrow k$

    pour $i = k \dots n$ **faire**

      si $|U[i,k]| > p$ **alors**

        $p \leftarrow U[i,k]$; $l \leftarrow i$

    si $l \neq k$ **alors**

      //Permutation de lignes

      pour $j = 1 \dots n$ **faire**

        $temp \leftarrow U[k,j]$; $U[k,j] \leftarrow U[l,j]$

        $U[l,j] \leftarrow temp$

    pour $i = k+1 \dots n$ **faire**

      $q \leftarrow U[i,k]$; $U[i,k] \leftarrow 0$

      pour $j = k+1 \dots n$ **faire**

        $U[i,j] = U[i,j] - U[k,j].\frac{q}{p}$

fin

retourner U

Données : $A = (A[i,j]), n$

début

$U \leftarrow A$; $L \leftarrow I$;

pour $k = 1 \dots n$ **faire**

  // Recherche partiel du pivot

  $p \leftarrow U[k,k]$; $l \leftarrow k$

  pour $i = k \dots n$ **faire**

    si $|U[i,k]| > p$ **alors**

      $p \leftarrow U[i,k]$; $l \leftarrow i$

  si $l \neq k$ **alors**

    //Permutation de lignes

    pour $j = 1 \dots n$ **faire**

      $temp \leftarrow U[k,j]$; $U[k,j] \leftarrow U[l,j]$

      $U[l,j] \leftarrow temp$

  pour $i = k+1 \dots n$ **faire**

    $q \leftarrow U[i,k]$; $U[i,k] \leftarrow 0$

    $L[i,k] \leftarrow \frac{q}{p}$

    pour $j = k+1 \dots n$ **faire**

      $U[i,j] = U[i,j] - U[k,j].\frac{q}{p}$

fin

retourner L et U

Données : $A = (A[i,j]), n$

début

- $U \leftarrow A$; $L \leftarrow I$;
- **pour** $k = 1 \ldots n$ **faire**
  - // Recherche partiel du pivot
  - $p \leftarrow U[k,k]$; $l \leftarrow k$
  - **pour** $i = k \ldots n$ **faire**
    - **si** $|U[i,k]| > p$ **alors**
      - $p \leftarrow U[i,k]$; $l \leftarrow i$
  - **si** $l \neq k$ **alors**
    - //Permutation de lignes
    - **pour** $j = 1 \ldots n$ **faire**
      - $temp \leftarrow U[k,j]$; $U[k,j] \leftarrow U[l,j]$
      - $U[l,j] \leftarrow temp$
      - **si** $j < k$ **alors**
        - //Uniquement sous la diagonale
        - $temp \leftarrow L[k,j]$; $L[k,j] \leftarrow L[l,j]$
        - $L[l,j] \leftarrow temp$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow U[i,k]$; $U[i,k] \leftarrow 0$
    - $L[i,k] \leftarrow \frac{q}{p}$
    - **pour** $j = k+1 \ldots n$ **faire**
      - $U[i,j] = U[i,j] - U[k,j].\frac{q}{p}$

fin

retourner L et U

Algorithme

Données : $A = (A[i,j]), n$

début

- $U \leftarrow A$; $L \leftarrow I$; $P \leftarrow I$
- **pour** $k = 1 \ldots n$ **faire**
  - // Recherche partiel du pivot
  - $p \leftarrow U[k,k]$; $l \leftarrow k$
  - **pour** $i = k \ldots n$ **faire**
    - **si** $|U[i,k]| > p$ **alors**
      - $p \leftarrow U[i,k]$; $l \leftarrow i$
  - **si** $l \neq k$ **alors**
    - //Permutation de lignes
    - **pour** $j = 1 \ldots n$ **faire**
      - $temp \leftarrow U[k,j]$; $U[k,j] \leftarrow U[l,j]$
      - $U[l,j] \leftarrow temp$
      - **si** $j < k$ **alors**
        - //Uniquement sous la diagonale
        - $temp \leftarrow L[k,j]$; $L[k,j] \leftarrow L[l,j]$
        - $L[l,j] \leftarrow temp$
      - $temp \leftarrow P[k,j]$; $P[k,j] \leftarrow P[l,j]$
      - $P[l,j] \leftarrow temp$
  - **pour** $i = k+1 \ldots n$ **faire**
    - $q \leftarrow U[i,k]$; $U[i,k] \leftarrow 0$
    - $L[i,k] \leftarrow \frac{q}{p}$
    - **pour** $j = k+1 \ldots n$ **faire**
      - $U[i,j] = U[i,j] - U[k,j].\frac{q}{p}$

fin

retourner P, L et U

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U

Exemple

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

Exemple

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
2			

Exemple

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
2			$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$

Propriétés mathématiques
Principe général des algorithmes
Triangularisation
Forme matricielle de la triangularisation
Conditions
Recherche de pivots maximaux
Recherche de pivots maximaux
Pivot partiel
Élimination avec pivot partiel
Pivot total
Résumé
Effet sur la décomposition LU
Matrice de permutation
Propriétés des matrices de permutation
Décomposition PLU
Algorithme
Exemple
Exemple(suite)
Conditionnement

Exemple

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
2	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$		$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$

Exemple

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
2	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$

Exemple

Calcul de la décomposition PLU de $\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$

k	P	L	U
0	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ -1 & -1 & 1 & 0 \\ 1 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
1	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 0 & 3 & 1 \\ 0 & -1 & 0 & -1 \\ 0 & 2 & 0 & 4 \end{pmatrix}$
2	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & -1 & 0 & -1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$
2	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -\frac{1}{2} & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$

Exemple(suite)

k	P	L	U
2	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -\frac{1}{2} & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$

Exemple(suite)

k	P	L	U
2	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & -\frac{1}{2} & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 3 & 1 \end{pmatrix}$
3	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 1 & -\frac{1}{2} & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 2 & 1 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 3 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$

Exemple(suite)

k	P	L	U
2	$\begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 1&0&0&0\\0&1&0&0\\1&-\frac{1}{2}&1&0\\-1&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 1&1&2&1\\0&2&0&4\\0&0&0&1\\0&0&3&1 \end{pmatrix}$
3	$\begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&1&0&0\\0&0&1&0 \end{pmatrix}$	$\begin{pmatrix} 1&0&0&0\\0&1&0&0\\-1&0&1&0\\1&-\frac{1}{2}&0&1 \end{pmatrix}$	$\begin{pmatrix} 1&1&2&1\\0&2&0&4\\0&0&3&1\\0&0&0&1 \end{pmatrix}$

Donc

$$\begin{pmatrix} 1&0&0&0\\0&0&0&1\\0&1&0&0\\0&0&1&0 \end{pmatrix} \times \begin{pmatrix} 1&1&2&1\\-1&-1&1&0\\1&0&2&0\\0&2&0&4 \end{pmatrix} = \begin{pmatrix} 1&0&0&0\\0&1&0&0\\-1&0&1&0\\1&-\frac{1}{2}&0&1 \end{pmatrix} \times \begin{pmatrix} 1&1&2&1\\0&2&0&4\\0&0&3&1\\0&0&0&1 \end{pmatrix}$$

Conditionnement

Exemple

Considérons le système linéaire suivant (R. S. WILSON)

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

il a pour solution le vecteur $x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

Considérons le système linéaire suivant (R. S. WILSON)

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

il a pour solution le vecteur $x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

Mais en changeant un peu le vecteur d'arrivée, $b = \begin{pmatrix} 32,1 \\ 22,9 \\ 33,1 \\ 30,9 \end{pmatrix}$

on trouve $x = \begin{pmatrix} 9,2 \\ -12,6 \\ 4,5 \\ -1,1 \end{pmatrix}$

Exemple

Considérons le système linéaire suivant (R. S. WILSON)

$$\begin{pmatrix} 10 & 7 & 8 & 7 \\ 7 & 5 & 6 & 5 \\ 8 & 6 & 10 & 9 \\ 7 & 5 & 9 & 10 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 32 \\ 23 \\ 33 \\ 31 \end{pmatrix}$$

il a pour solution le vecteur $x = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

Mais en changeant un peu le vecteur d'arrivée, $b = \begin{pmatrix} 32,1 \\ 22,9 \\ 33,1 \\ 30,9 \end{pmatrix}$

on trouve $x = \begin{pmatrix} 9,2 \\ -12,6 \\ 4,5 \\ -1,1 \end{pmatrix}$

Une erreur relative de $\frac{1}{200}$ sur les données entraîne une erreur de $\frac{10}{1}$ sur le résultat : 2000 *fois plus !*

Conditionnement

Rappel : Si r est la solution d'un problème de donnée a et $r + \Delta r$ celle du même problème avec les données $a + \Delta a$, on appelle conditionnement du problème la valeur

$$C_\alpha = \sup_{\|\Delta a\| \leq \alpha} \frac{\|\Delta r\|}{\|\Delta a\|}$$

En utilisant cette définition, on peut analyser la sensibilité du problème $Ax = b$ au données :

- si b est remplacé par $b + \Delta b$
- si A est remplacée par $A + \Delta A$

Norme matricielle

Définition (Normes induites (ou subordonnées)) *Soit $\|.\|_v$ une norme vectorielle définie sur $\mathbb{C}^n$ la fonction qui $\forall A \in \mathcal{M}_n(\mathbb{C})$ associe*

$$\|A\| \quad = \quad \max_{x \in \mathbb{C}^n, x \neq 0} \frac{\|Ax\|_v}{\|x\|_v}$$

est une norme matricielle dite norme matricielle induite *ou subordonnée*

Norme matricielle

Définition (Normes induites (ou subordonnées)) *Soit* $\|.\|_v$ *une norme vectorielle définie sur* $\mathbb{C}^n$ *la fonction qui* $\forall A \in \mathcal{M}_n(\mathbb{C})$ *associe*

$$\|A\| = \max_{x\in\mathbb{C}^n, x\neq 0} \frac{\|Ax\|_v}{\|x\|_v}$$

est une norme matricielle dite norme matricielle induite *ou subordonnée*

Par exemple, la norme matricielle induite par la norme 2 sur une matrice symétrique est

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

est $$\|A\| = \max_{\lambda\in spec(A)} |\lambda|$$

C'est à dire la plus grande valeur propre de A.

Par définition, lorsque la norme $\|.\|_m$ est induite par la norme vectorielle $\|.\|_v$ alors

$$\|Ax\|_v \leq \|A\|_m \|x\|_v$$

Norme matricielle

Définition (Normes induites (ou subordonnées)) *Soit* $\|.\|_v$ *une norme vectorielle définie sur* $\mathbb{C}^n$ *la fonction qui* $\forall A \in \mathcal{M}_n(\mathbb{C})$ *associe*

$$\|A\| = \max_{x\in\mathbb{C}^n, x\neq 0} \frac{\|Ax\|_v}{\|x\|_v}$$

est une norme matricielle dite norme matricielle induite *ou subordonnée*

Par exemple, la norme matricielle induite par la norme 2 sur une matrice symétrique est

$$\|x\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$$

est $$\|A\| = \max_{\lambda\in spec(A)} |\lambda|$$

C'est à dire la plus grande valeur propre de A.

Par définition, lorsque la norme $\|.\|_m$ est induite par la norme vectorielle $\|.\|_v$ alors

$$\|Ax\|_v \leq \|A\|_m \|x\|_v$$

Conditionnement de la matrice

On peut utiliser ces normes matricielles pour majorer la sensibilité de la solution au problème sur les données

Conditionnement de la matrice

On peut utiliser ces normes matricielles pour majorer la sensibilité de la solution au problème sur les données
Si A est une matrice inversible et si u est solution de $Ax = b$,

Conditionnement de la matrice

On peut utiliser ces normes matricielles pour majorer la sensibilité de la solution au problème sur les données
Si A est une matrice inversible et si u est solution de $Ax = b$,

- $u + \Delta u$ est solution de $Ax = b + \Delta b$ avec

$$
\begin{aligned}
\Delta u &= A^{-1}(\Delta b) \\
\text{Donc } \frac{\Delta u}{u} &\leq \|A\| \, \|A^{-1}\| \, \frac{\Delta b}{b}
\end{aligned}
$$

Conditionnement de la matrice

On peut utiliser ces normes matricielles pour majorer la sensibilité de la solution au problème sur les données
Si A est une matrice inversible et si u est solution de $Ax = b$,

- $u + \Delta u$ est solution de $Ax = b + \Delta b$ avec

$$\begin{aligned} \Delta u &= A^{-1}(\Delta b) \\ \text{Donc } \frac{\Delta u}{u} &\leq \|A\| \, \|A^{-1}\| \, \frac{\Delta b}{b} \end{aligned}$$

- $u + \Delta u$ est solution de $(A + \Delta A)x = b$ avec

$$\begin{aligned} \Delta u &= A^{-1}(\Delta A(u + \Delta u)) \\ \text{Donc } \frac{\Delta u}{u + \Delta u} &\leq \|A\| \, \|A^{-1}\| \, \frac{\|\Delta A\|}{\|A\|} \end{aligned}$$

Conditionnement de la matrice

On peut utiliser ces normes matricielles pour majorer la sensibilité de la solution au problème sur les données
Si A est une matrice inversible et si u est solution de $Ax = b$,

- $u + \Delta u$ est solution de $Ax = b + \Delta b$ avec

$$\begin{aligned} \Delta u &= A^{-1}(\Delta b) \\ \text{Donc } \frac{\Delta u}{u} &\leq \|A\| \, \|A^{-1}\| \, \frac{\Delta b}{b} \end{aligned}$$

- $u + \Delta u$ est solution de $(A + \Delta A)x = b$ avec

$$\begin{aligned} \Delta u &= A^{-1}(\Delta A(u + \Delta u)) \\ \text{Donc } \frac{\Delta u}{u + \Delta u} &\leq \|A\| \, \|A^{-1}\| \, \frac{\|\Delta A\|}{\|A\|} \end{aligned}$$

Définition (Conditionnement) *On appelle conditionnement de la matrice A la valeur*

$$cond(A) = \|A\| \, \|A^{-1}\|$$

Propriété du conditionnement

- Le conditionnement d'une matrice est toujours $\geq$ à 1

Propriété du conditionnement

- Le conditionnement d'une matrice est toujours $\geq$ à 1
- L'erreur relative sur la solution est inférieure à l'erreur relative sur les données multipliée par $cond(A)$.

Propriété du conditionnement

- Le conditionnement d'une matrice est toujours $\geq$ à 1
- L'erreur relative sur la solution est inférieure à l'erreur relative sur les données multipliée par $cond(A)$.
- Cette borne est optimale

Propriété du conditionnement

- Le conditionnement d'une matrice est toujours $\geq$ à 1
- L'erreur relative sur la solution est inférieure à l'erreur relative sur les données multipliée par $cond(A)$.
- Cette borne est optimale
- La valeur dépend de la norme vectorielle utilisée, dans le cas de la norme 2 pour les matrices symétriques,

$$cond(A)_2 = \frac{|\lambda_n|}{|\lambda_1|}$$

où λ_n (resp. λ_1) est la plus grande (resp. la plus petite) valeur propre
Dans le cas de la matrice donnée en exemple, $cond(A) \simeq 2984$.

Propriété du conditionnement

- Le conditionnement d'une matrice est toujours $\geq$ à 1
- L'erreur relative sur la solution est inférieure à l'erreur relative sur les données multipliée par $cond(A)$.
- Cette borne est optimale
- La valeur dépend de la norme vectorielle utilisée, dans le cas de la norme 2 pour les matrices symétriques,

$$cond(A)_2 = \frac{|\lambda_n|}{|\lambda_1|}$$

  où λ_n (resp. λ_1) est la plus grande (resp. la plus petite) valeur propre
  Dans le cas de la matrice donnée en exemple, $cond(A) \simeq 2984$.
- Il existe des méthodes pour améliorer le conditionnement.

Conclusion

Les algorithmes vus en cours sont des algorithmes généraux mais ils sont

- coûteux
- instables
- Il existe d'autres décompositions
  - Décomposition LL' pour les matrices symétriques définies positives (Choleski),
  - Décomposition LDL' pour les matrices symétriques
- Il y a des algorithmes plus efficaces pour les matrices spéciales
  - tridiagonales,
  - creuses.

Vous pouvez tester le package « linalg » de MAPLE, SCILAB, MATLAB et la librairie LAPACK.