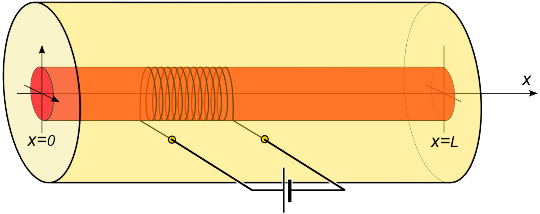
MT00 - Chapitre 2

Résolution des systèmes linéaires

Elias Khoury

Printemps 2024

Équation de la chaleur



$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0, L[,\ t > 0,$$

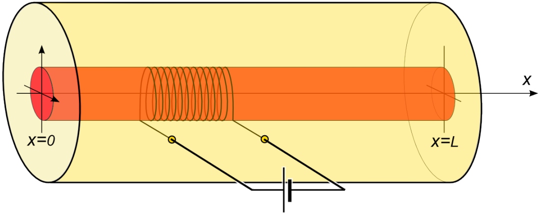
- Conditions aux limites

$$\forall t > 0,\ u(0,t) = u(L,t) = u_{ext}$$

- Conditions initiales

$$\forall x \in]0, L[,\ u(x,0) = u_{ext}$$

Équation de la chaleur

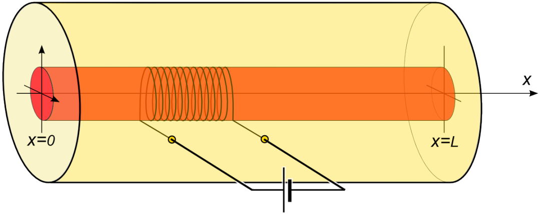


$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0, L[,\ t > 0,$$

- $u(x,t)$: température (K) à l'abscisse x (m) et au temps t (s),
- ρ : masse volumique (kg.m^{-3}),
- c : capacité calorifique (J.kg^{-1}K^{-1}),
- λ : conductivité thermique (W.m^{-1}.K^{-1})
- f : flux interne de chaleur (W.m^{-3})
- Conditions aux limites

$$\forall t > 0,\ u(0,t) = u(L,t)$$

Équation de la chaleur

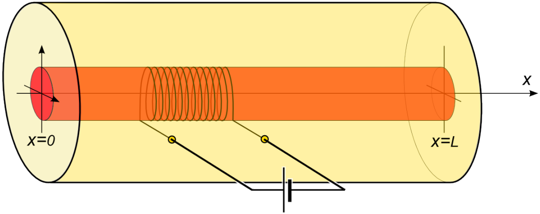


$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0, L[,\ t > 0,$$

- $u(x,t)$: température (K) à l'abscisse x (m) et au temps t (s),
- ρ : masse volumique (kg.m^{-3}),
- c : capacité calorifique (J.kg^{-1}K^{-1}),
- λ : conductivité thermique (W.m^{-1}.K^{-1})
- f : flux interne de chaleur (W.m^{-3})
- Conditions aux limites

$$\forall t > 0,\ u(0,t) = u(L,t)$$

Équation de la chaleur

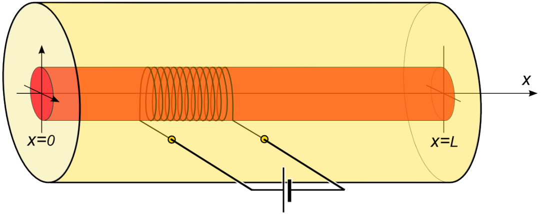


$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0,L[,\ t > 0,$$

- $u(x,t)$: température (K) à l'abscisse x (m) et au temps t (s),
- ρ : masse volumique (kg.m^{-3}),
- c : capacité calorifique (J.kg^{-1}K^{-1}),
- λ : conductivité thermique (W.m^{-1}.K^{-1})
- f : flux interne de chaleur (W.m^{-3})
- Conditions aux limites

$$\forall t > 0,\ u(0,t) = u(L,t)$$

Équation de la chaleur

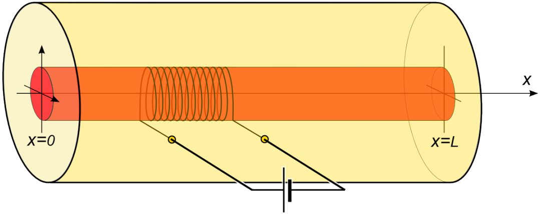


$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0, L[,\ t > 0,$$

- $u(x,t)$: température (K) à l'abscisse x (m) et au temps t (s),
- ρ : masse volumique (kg.m^{-3}),
- c : capacité calorifique (J.kg^{-1}K^{-1}),
- λ : conductivité thermique (W.m^{-1}.K^{-1})
- f : flux interne de chaleur (W.m^{-3})
- Conditions aux limites

$$\forall t > 0,\ u(0,t) = u(L,t)$$

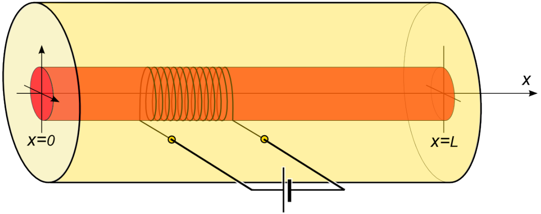
Équation de la chaleur



$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0, L[,\ t > 0,$$

- $u(x,t)$: température (K) à l'abscisse x (m) et au temps t (s),
- ρ : masse volumique (kg.m^{-3}),
- c : capacité calorifique (J.kg^{-1}K^{-1}),
- λ : conductivité thermique (W.m^{-1}.K^{-1})
- f : flux interne de chaleur (W.m^{-3})

Équation de la chaleur



$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0,L[,\ t > 0,$$

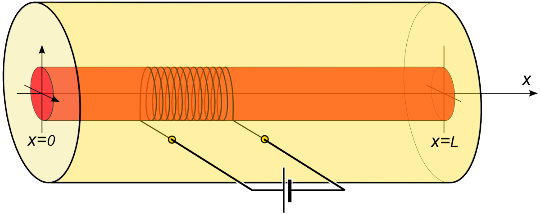
- Conditions aux limites

$$\forall t > 0,\ \ u(0,t) = u(L,t) = u_{ext}$$

- Conditions initiales

$$\forall x \in]0,L[,\ \ u(x,0) = u_{ext}$$

Équation de la chaleur



$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x,t),\ x \in]0,L[,\ t > 0,$$

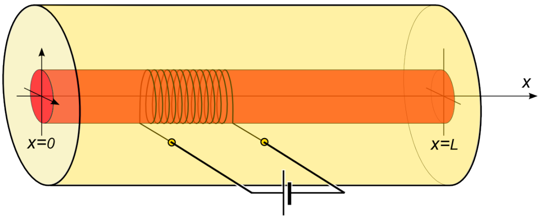
- Conditions aux limites

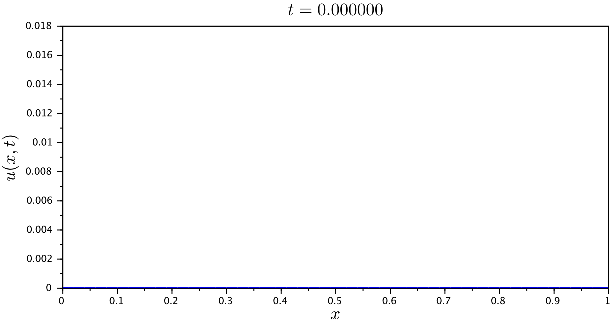
$$\forall t > 0,\ \ u(0,t) = u(L,t) = u_{ext}$$

- Conditions initiales

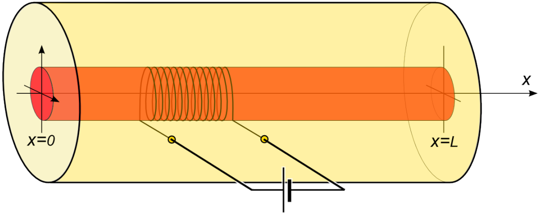
$$\forall x \in]0,L[,\ \ u(x,0) = u_{ext}$$

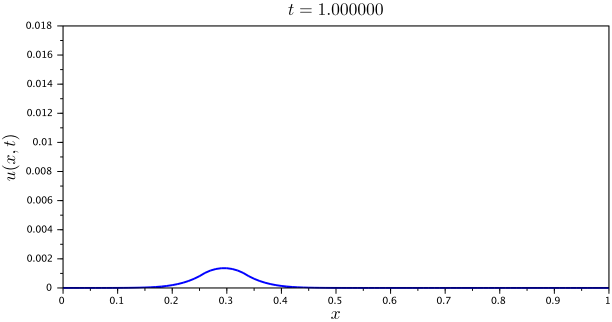
Équation de la chaleur



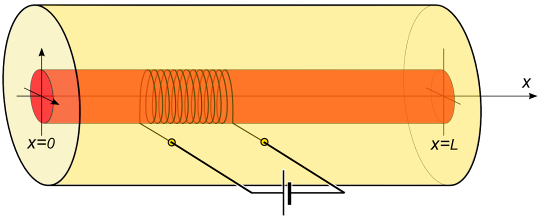


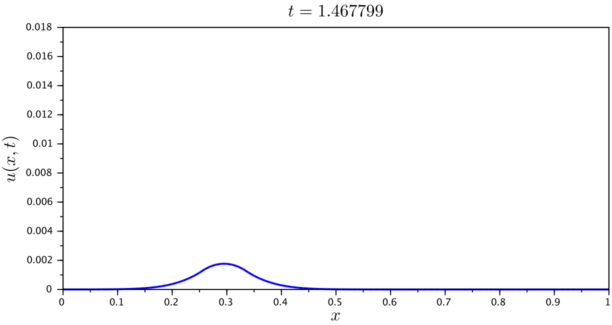
Équation de la chaleur



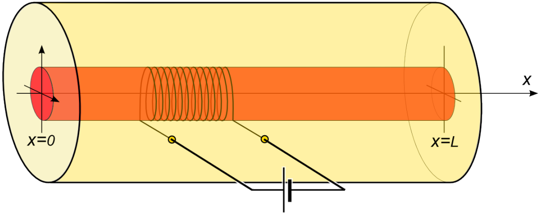


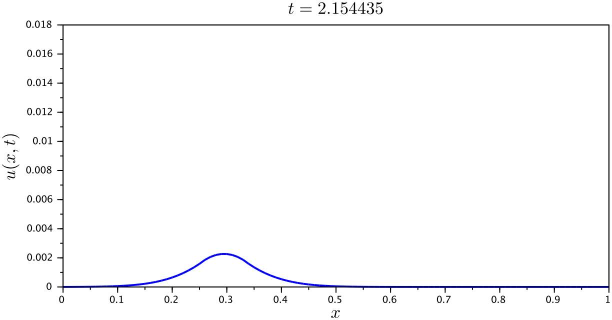
Équation de la chaleur



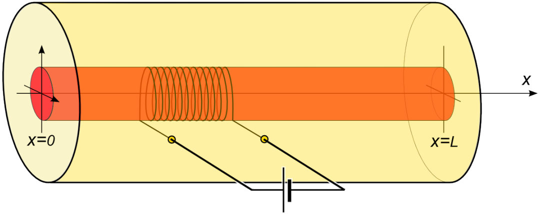


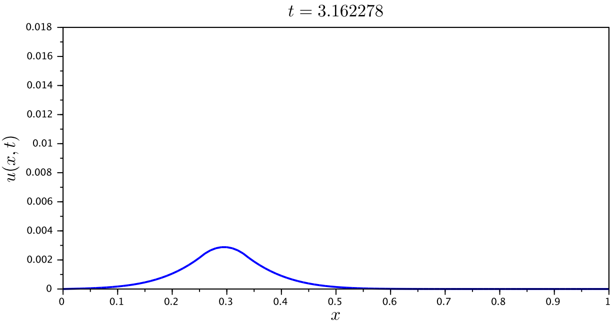
Équation de la chaleur



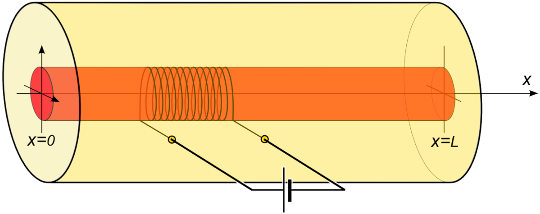


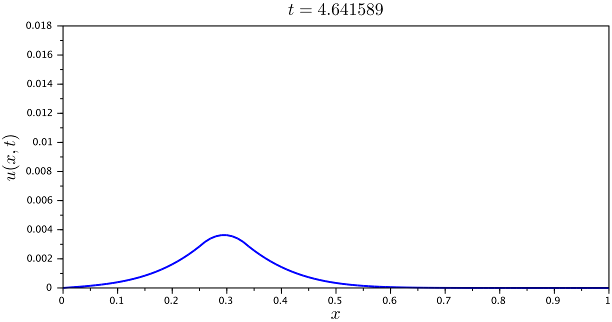
Équation de la chaleur



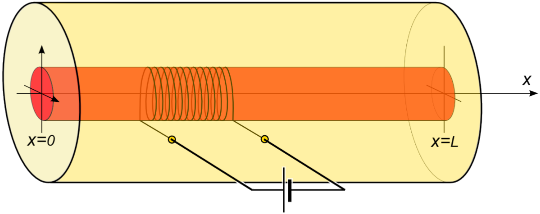


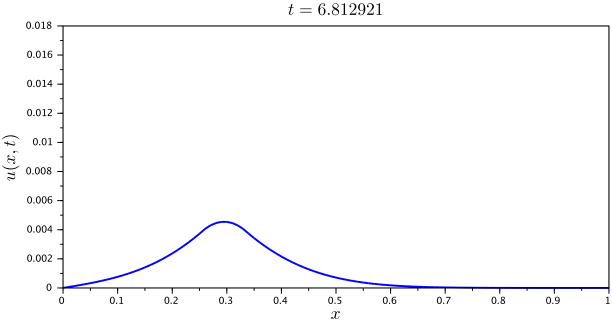
Équation de la chaleur



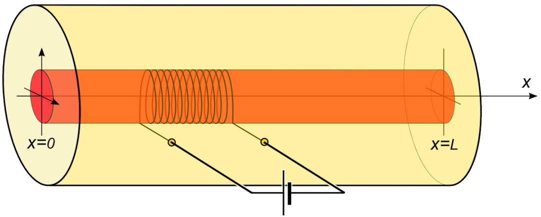


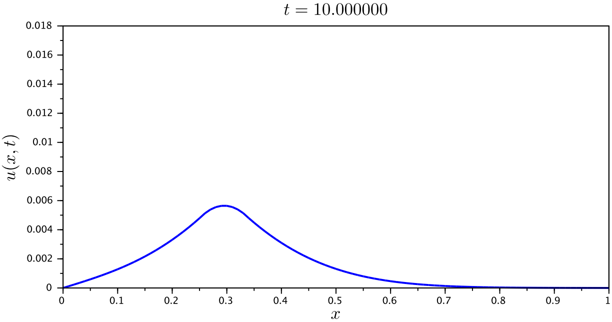
Équation de la chaleur



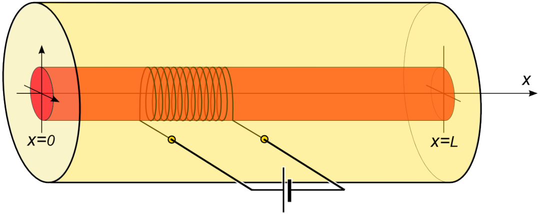


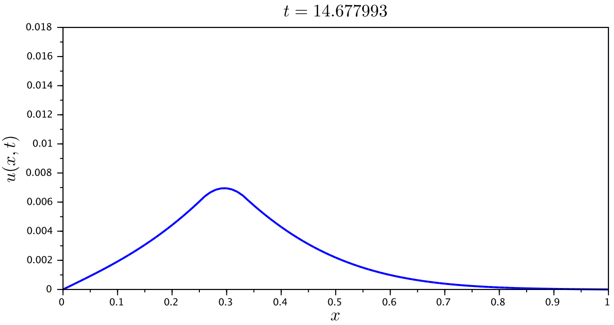
Équation de la chaleur



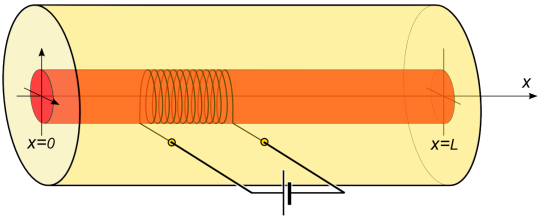


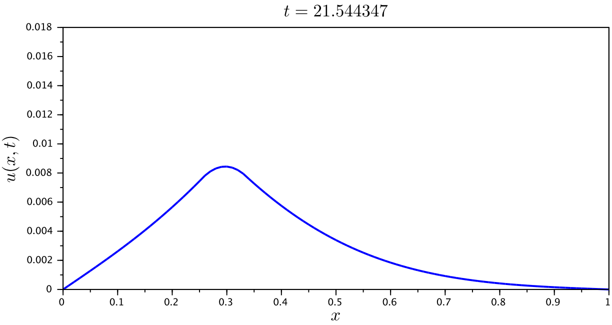
Équation de la chaleur



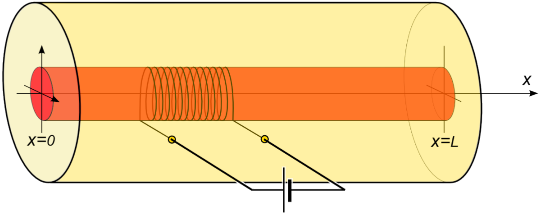


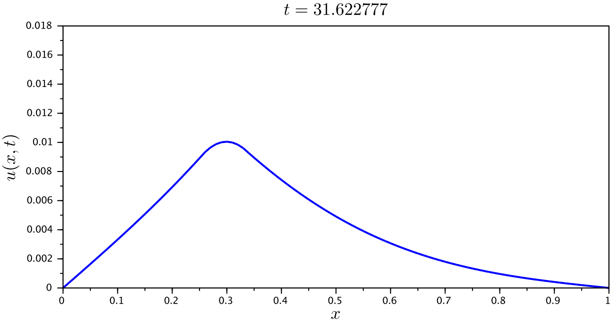
Équation de la chaleur



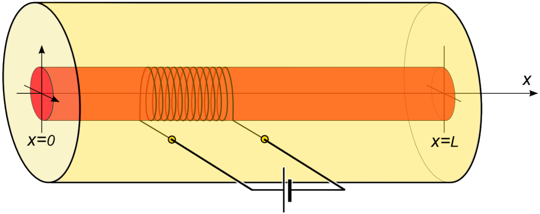


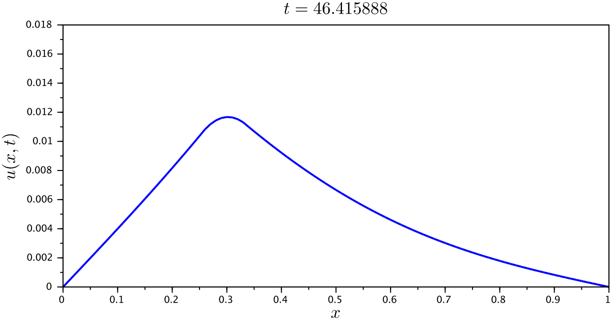
Équation de la chaleur



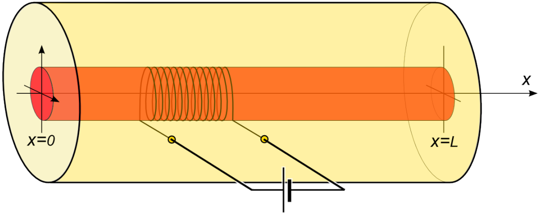


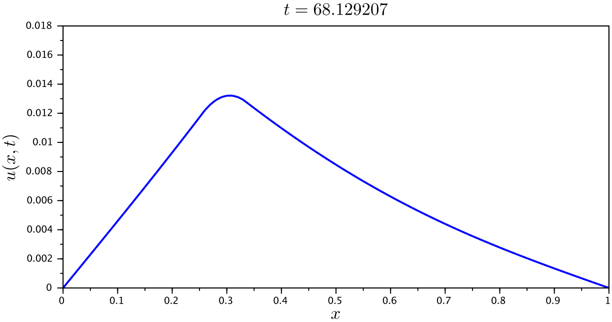
Équation de la chaleur



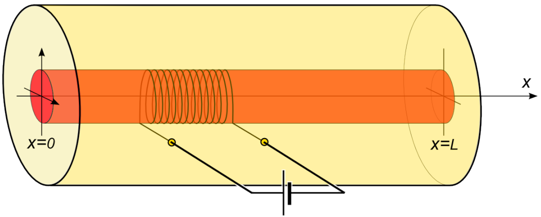


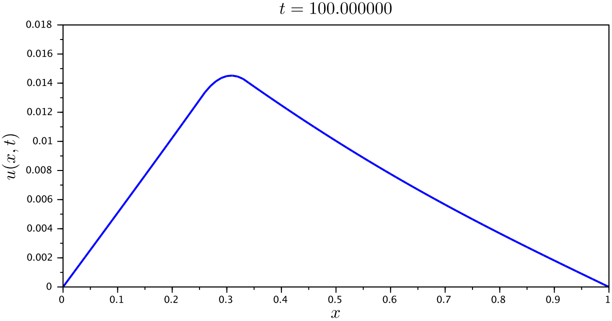
Équation de la chaleur



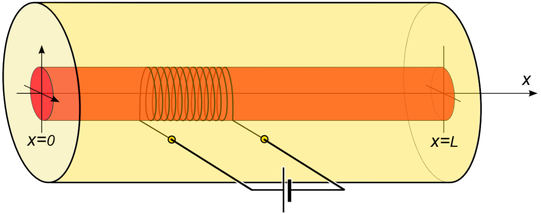


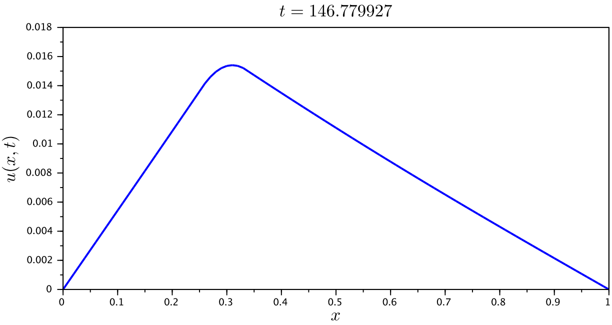
Équation de la chaleur



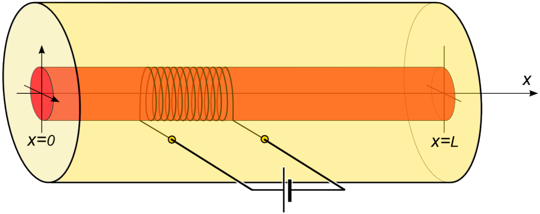


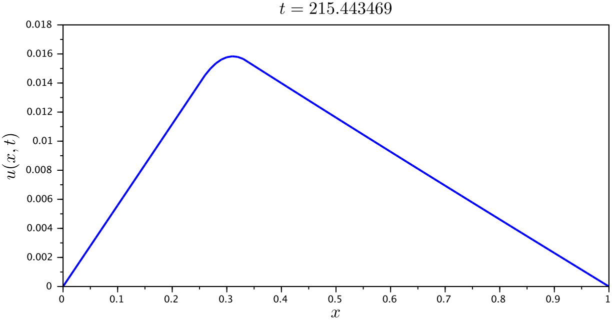
Équation de la chaleur



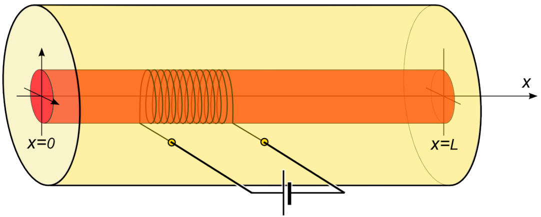


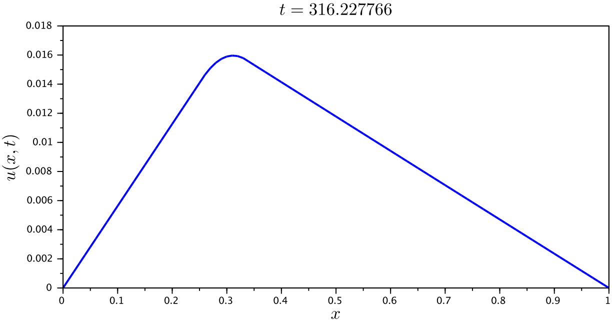
Équation de la chaleur



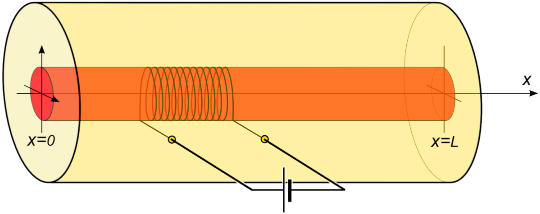


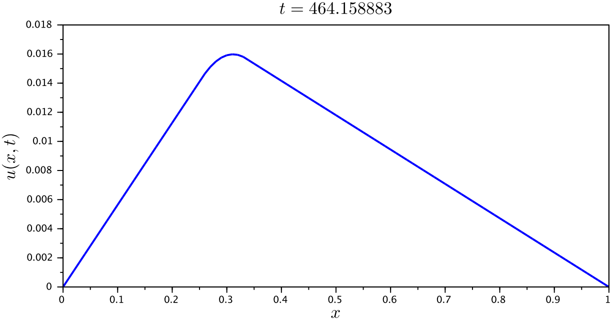
Équation de la chaleur



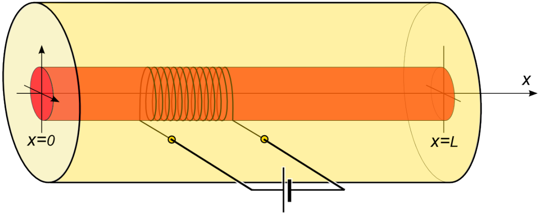


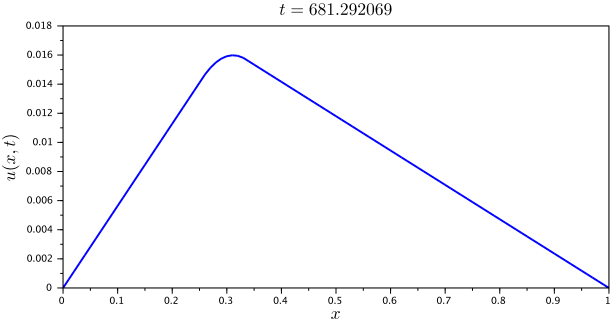
Équation de la chaleur



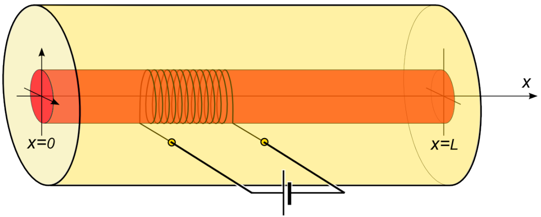


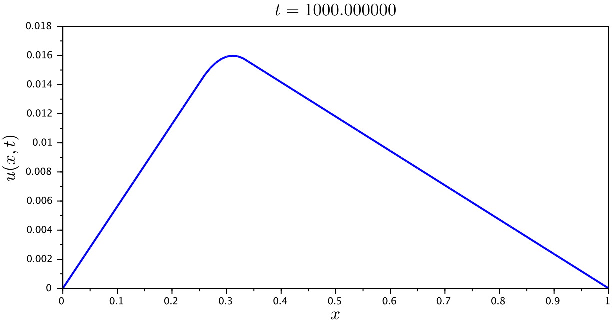
Équation de la chaleur





Équation de la chaleur





Équation de la chaleur

Solution stationnaire

- Hypothèse : $\forall t > 0,\ f(x,t) = f(x)$

$$\Longrightarrow \lim_{t\to\infty} u(x,t) = u(x),$$

$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x),\ x \in]0,L[,\ t > 0,$$

- u est solution de l'équation

$$-\lambda u''(x) = f(x),\ x \in]0,L[,$$
$$u(0) = u(L) = u_{ext}$$

Équation de la chaleur

Solution stationnaire

- Hypothèse : $\forall t > 0,\ f(x,t) = f(x)$

$$\Longrightarrow \lim_{t\to\infty} u(x,t) = u(x),$$

$$\rho c \frac{\partial u}{\partial t}(x,t) - \lambda \frac{\partial^2 u}{\partial x^2}(x,t) = f(x),\ x \in]0,L[,\ t > 0,$$

- u est solution de l'équation

$$-\lambda u''(x) = f(x),\ x \in]0,L[,$$
$$u(0) = u(L) = u_{ext}$$

- Discrétisation de l'intervalle $[0, L]$:

$$x_k = kh, \; h = \frac{L}{n}$$

- On cherche une approximation de $u(x_k)$ pour $k = 0, \ldots, n$
- Moyen : approximation de la dérivée seconde de $u(x_k)$:

$$u(x_k + h) = u(x_k) + hu'(x_k) + \frac{h^2}{2}u''(x_k) + h^2\varepsilon_1(h),$$
$$u(x_k - h) = u(x_k) - hu'(x_k) + \frac{h^2}{2}u''(x_k) + h^2\varepsilon_2(h),$$

Équation de la chaleur

Approximation de la solution stationnaire

- Discrétisation de l'intervalle $[0, L]$:

$$x_k = kh, \; h = \frac{L}{n}$$

- On cherche une approximation de $u(x_k)$ pour $k = 0, \ldots, n$
- Moyen : approximation de la dérivée seconde de $u(x_k)$:

$$u(x_k + h) = u(x_k) + hu'(x_k) + \frac{h^2}{2}u''(x_k) + h^2\varepsilon_1(h),$$
$$u(x_k - h) = u(x_k) - hu'(x_k) + \frac{h^2}{2}u''(x_k) + h^2\varepsilon_2(h),$$

Équation de la chaleur
Approximation de la solution stationnaire

- Discrétisation de l'intervalle $[0, L]$:

$$x_k = kh, \; h = \frac{L}{n}$$

- On cherche une approximation de $u(x_k)$ pour $k = 0, \ldots, n$
- Moyen : approximation de la dérivée seconde de $u(x_k)$:

$$u(x_k + h) = u(x_k) + hu'(x_k) + \frac{h^2}{2} u''(x_k) + h^2 \varepsilon_1(h),$$
$$u(x_k - h) = u(x_k) - hu'(x_k) + \frac{h^2}{2} u''(x_k) + h^2 \varepsilon_2(h),$$

Équation de la chaleur

Approximation de la solution stationnaire

$$u''(x_k) = \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} + \varepsilon(h),\ k = 1, \dots, n-1$$

- Hypothèse : $u_{ext} = 0$
- Équation de la chaleur stationnaire en x_k, $k = 1, \dots, n-1$:

$$\begin{aligned}
-\lambda \frac{-2u(x_1) + u(x_2)}{h^2} &= f(x_1) + \lambda\varepsilon(h), \\
-\lambda \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} &= f(x_k) + \lambda\varepsilon(h),\ k = 2, \dots, n-2 \\
-\lambda \frac{u(x_{n-2}) - 2u(x_{n-1})}{h^2} &= f(x_{n-1}) + \lambda\varepsilon(h)
\end{aligned}$$

$\Longrightarrow$ si on néglige $\varepsilon(h)$, on peut approcher $u(x_k)$, $k = 1, \dots, n-1$

Équation de la chaleur

Approximation de la solution stationnaire

$$u''(x_k) = \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} + \varepsilon(h),\ k = 1, \dots, n-1$$

- Hypothèse : $u_{ext} = 0$
- Équation de la chaleur stationnaire en x_k, $k = 1, \dots, n-1$:

$$-\lambda \frac{-2u(x_1) + u(x_2)}{h^2} = f(x_1) + \lambda\varepsilon(h),$$

$$-\lambda \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} = f(x_k) + \lambda\varepsilon(h),\ k = 2, \dots, n-2$$

$$-\lambda \frac{u(x_{n-2}) - 2u(x_{n-1})}{h^2} = f(x_{n-1}) + \lambda\varepsilon(h)$$

$\Longrightarrow$ si on néglige $\varepsilon(h)$, on peut approcher $u(x_k)$, $k = 1, \dots, n-1$

Équation de la chaleur

Approximation de la solution stationnaire

$$u''(x_k) = \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} + \varepsilon(h),\ k = 1, \ldots, n-1$$

- Hypothèse : $u_{ext} = 0$
- Équation de la chaleur stationnaire en x_k, $k = 1, \ldots, n-1$:

$$\begin{aligned}
-\lambda \frac{-2u(x_1) + u(x_2)}{h^2} &= f(x_1) + \lambda\varepsilon(h), \\
-\lambda \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} &= f(x_k) + \lambda\varepsilon(h),\ k = 2, \ldots, n-2 \\
-\lambda \frac{u(x_{n-2}) - 2u(x_{n-1})}{h^2} &= f(x_{n-1}) + \lambda\varepsilon(h)
\end{aligned}$$

$\Longrightarrow$ si on néglige $\varepsilon(h)$, on peut approcher $u(x_k)$, $k = 1, \ldots, n-1$

Équation de la chaleur

Approximation de la solution stationnaire

$$u''(x_k) = \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} + \varepsilon(h),\ k = 1, \dots, n-1$$

- Hypothèse : $u_{ext} = 0$
- Équation de la chaleur stationnaire en x_k, $k = 1, \dots, n-1$:

$$\begin{aligned}
-\lambda \frac{-2u(x_1) + u(x_2)}{h^2} &= f(x_1) + \lambda\varepsilon(h), \\
-\lambda \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1})}{h^2} &= f(x_k) + \lambda\varepsilon(h),\ k = 2, \dots, n-2 \\
-\lambda \frac{u(x_{n-2}) - 2u(x_{n-1})}{h^2} &= f(x_{n-1}) + \lambda\varepsilon(h)
\end{aligned}$$

$\Longrightarrow$ si on néglige $\varepsilon(h)$, on peut approcher $u(x_k)$, $k = 1, \dots, n-1$

Équation de la chaleur

Approximation de la solution stationnaire

- On pose $v = [v_1, \dots, v_{n-1}]^\top$ où v_k est l'approximation de $u(x_k)$
- v est solution du système d'équations linéaires

$$-\frac{\lambda}{h^2}\begin{pmatrix} -2 & 1 & & & 0 \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \\ 0 & & & 1 & -2 \end{pmatrix}\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{n-2} \\ v_{n-1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{n-2}) \\ f(x_{n-1}) \end{pmatrix},$$

$$v_0 = v_n = 0.$$

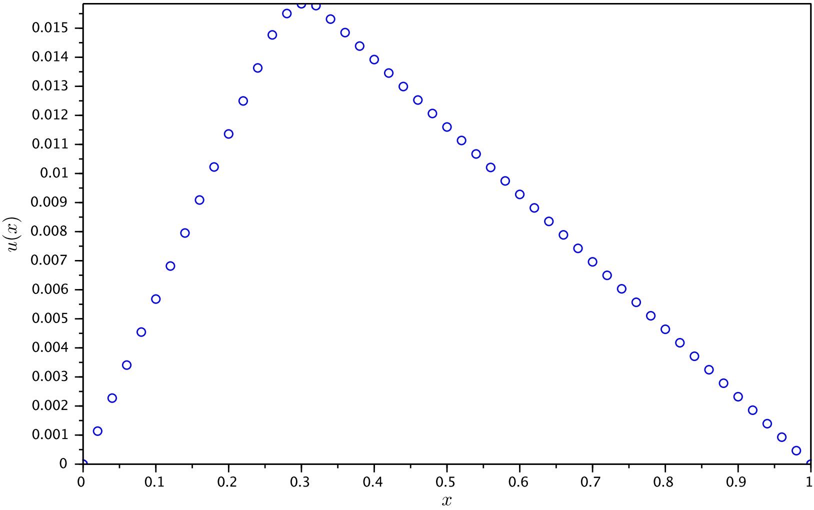
Équation de la chaleur

Approximation de la solution stationnaire (avec Scilab)

```
L=1; n=50; h=L/n; lambda=1;

A=zeros(n-1,n-1);
A(1,1:2)=[-2 1];
for i=2:n-2;
   A(i,i-1:i+1)=[1 -2 1];
end
A(n-1,n-2:n-1)=[1 -2];
A=-A*lambda/h^2;
x=linspace(h,L-h,n-1);
f=(x>L/4 & x< L/3)';

v=A\f;    // Resolution du systeme

plot(x,v,"o")
```

Équation de la chaleur

Approximation de la solution stationnaire (avec Scilab)



L'algorithme de Gauss
Méthode de Cramer

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ avec $\det A \neq 0$, $b \in \mathbb{R}^n$

$$Ax = b \Longleftrightarrow x_i = \frac{\det(A_1, \dots, A_{i-1}, b, A_{i+1}, \dots, A_n)}{\det A}, \; i = 1, \dots, n.$$

Le calcul de chaque déterminant nécessite $nn!$ opérations :

$$\det A = \sum_{\sigma \in S(\{1,\dots,n\})} \mathrm{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

$\Rightarrow$ calcul de x en $n(n+1)!$ opérations

- Exemple pour $n = 16$:

5.690998810^{15} opérations soit plus de 15 heures sur un GPU à 10^{11} FLOPS (100 GFLOPS) !

L'algorithme de Gauss

Méthode de Cramer

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ avec $\det A \neq 0$, $b \in \mathbb{R}^n$

$$Ax = b \Longleftrightarrow x_i = \frac{\det(A_1, \ldots, A_{i-1}, b, A_{i+1}, \ldots, A_n)}{\det A}, \; i = 1, \ldots, n.$$

Le calcul de chaque déterminant nécessite $nn!$ opérations :

$$\det A = \sum_{\sigma \in S(\{1,\ldots,n\})} \operatorname{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

$\Rightarrow$ calcul de x en $n(n+1)!$ opérations

- Exemple pour $n = 16$:

5.690998810^{15} opérations soit plus de 15 heures
sur un GPU à 10^{11} FLOPS (100 GFLOPS) !

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ avec $\det A \neq 0$, $b \in \mathbb{R}^n$

$$Ax = b \Longleftrightarrow x_i = \frac{\det(A_1, \ldots, A_{i-1}, b, A_{i+1}, \ldots, A_n)}{\det A}, \; i = 1, \ldots, n.$$

Le calcul de chaque déterminant nécessite $nn!$ opérations :

$$\det A = \sum_{\sigma \in S(\{1,\ldots,n\})} \mathrm{sign}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)}$$

$\Rightarrow$ calcul de x en $n(n+1)!$ opérations

- Exemple pour $n = 16$:

5.690998810[15] opérations soit plus de 15 heures
sur un GPU à 10^{11} FLOPS (100 GFLOPS) !

L'algorithme de Gauss

Résolution d'un système triangulaire

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ triangulaire inférieure avec det $A \neq 0$, $b \in \mathbb{R}^n$

$$A = \begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad j > i \Rightarrow a_{ij} = 0.$$

- Solution

$$x_1 = \frac{b_1}{a_{11}},$$

$$x_i = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right) / a_{ii}, \; i = 2, \dots, n$$

L'algorithme de Gauss

Résolution d'un système triangulaire

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ triangulaire inférieure avec $\det A \neq 0$, $b \in \mathbb{R}^n$

$$A = \begin{pmatrix} a_{11} & & & \\ a_{21} & a_{22} & & \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, \quad j > i \Rightarrow a_{ij} = 0.$$

- Solution

$$x_1 = \frac{b_1}{a_{11}},$$

$$x_i = \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j \right) / a_{ii},\ i = 2, \dots, n$$

L'algorithme de Gauss

Résolution d'un système triangulaire

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ triangulaire supérieure avec $\det A \neq 0$, $b \in \mathbb{R}^n$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ & a_{22} & \dots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix}, \quad i > j \Rightarrow a_{ij} = 0.$$

- Solution

$$x_n = \frac{b_n}{a_{nn}}, \quad x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) / a_{ii},\ i = 1, \dots, n-1$$

Nombre d'opérations : $2 \times \dfrac{n(n-1)}{2} + n = n^2$

L'algorithme de Gauss

Résolution d'un système triangulaire

- $A \in \mathcal{M}_{n,n}(\mathbb{R})$ triangulaire supérieure avec $\det A \neq 0$, $b \in \mathbb{R}^n$

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ & a_{22} & \dots & a_{2n} \\ & & \ddots & \vdots \\ & & & a_{nn} \end{pmatrix}, \quad i > j \Rightarrow a_{ij} = 0.$$

- Solution

$$x_n = \frac{b_n}{a_{nn}}, \quad x_i = \left(b_i - \sum_{j=i+1}^{n} a_{ij} x_j \right) / a_{ii},\ i = 1, \dots, n-1$$

Nombre d'opérations : $2 \times \dfrac{n(n-1)}{2} + n = n^2$

Construire $U \in \mathcal{M}_{n,n}(\mathbb{R})$ triangulaire supérieure et $y \in \mathbb{R}^n$ tels que

$$Ax = b \Longleftrightarrow Ux = y$$

- Exemple :

$$\begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ 2x_1 & +6x_2 & +x_3 & +2x_4 & = -1 \\ -x_1 & & +4x_3 & -2x_4 & = 1 \\ x_1 & +6x_2 & +5x_3 & +4x_4 & = 4 \end{array}$$

- Étape 1 :

$$\begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ 2x_1 & +6x_2 & +x_3 & +2x_4 & = -1 \\ -x_1 & & +4x_3 & -2x_4 & = 1 \\ x_1 & +6x_2 & +5x_3 & +4x_4 & = 4 \end{array}$$

$$\Rightarrow \quad x_1 \quad +2x_2 \qquad\qquad +x_4 \quad = 0$$

- Étape 1 :

$$\begin{array}{lllllll}
x_1 & +2x_2 & & +x_4 & = 0 & \\
2x_1 & +6x_2 & +x_3 & +2x_4 & = -1 & L_2 \leftarrow L_2 - 2L_1 \\
-x_1 & & +4x_3 & -2x_4 & = 1 & \\
x_1 & +6x_2 & +5x_3 & +4x_4 & = 4 &
\end{array}$$

$$\Rightarrow \quad \begin{array}{lllll}
x_1 & +2x_2 & & +x_4 & = 0 \\
 & +2x_2 & +x_3 & & = -1
\end{array}$$

- Étape 1 :

$$\begin{array}{lllll} x_1 & +2x_2 & & +x_4 & =0 \\ 2x_1 & +6x_2 & +x_3 & +2x_4 & =-1 \\ -x_1 & & +4x_3 & -2x_4 & =1 \\ x_1 & +6x_2 & +5x_3 & +4x_4 & =4 \end{array} \qquad L_3 \leftarrow L_3 + L_1$$

$$\Rightarrow \quad \begin{array}{lllll} x_1 & +2x_2 & & +x_4 & =0 \\ & +2x_2 & +x_3 & & =-1 \\ & +2x_2 & +4x_3 & -x_4 & =1 \end{array}$$

L'algorithme de Gauss

Principe de l'algorithme

- Étape 1 :

$$\begin{array}{llllll}
x_1 & +2x_2 & & +x_4 & = 0 & \\
2x_1 & +6x_2 & +x_3 & +2x_4 & = -1 & \\
-x_1 & & +4x_3 & -2x_4 & = 1 & \\
x_1 & +6x_2 & +5x_3 & +4x_4 & = 4 & L_4 \leftarrow L_4 - L_1
\end{array}$$

$$\Rightarrow \quad \begin{array}{lllll}
x_1 & +2x_2 & & +x_4 & = 0 \\
 & +2x_2 & +x_3 & & = -1 \\
 & +2x_2 & +4x_3 & -x_4 & = 1 \\
 & +4x_2 & +5x_3 & +3x_4 & = 4
\end{array}$$

L'algorithme de Gauss

Principe de l'algorithme

- Étape 2 :

$$\begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & \color{blue}{+2x_2} & \color{blue}{+x_3} & & \color{blue}{= -1} \\ & +2x_2 & +4x_3 & -x_4 & = 1 \\ & +4x_2 & +5x_3 & +3x_4 & = 4 \end{array}$$

$$\Rightarrow \quad \begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & & & \\ & & & & \end{array}$$

- Étape 2 :

$$\begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & +2x_2 & +4x_3 & -x_4 & = 1 \qquad L_3 \leftarrow L_3 - L_2 \\ & +4x_2 & +5x_3 & +3x_4 & = 4 \end{array}$$

$$\Rightarrow \quad \begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & +3x_3 & -x_4 & = 2 \end{array}$$

- Étape 2 :

$$\begin{array}{llllll}
x_1 & +2x_2 & & +x_4 & = 0 & \\
 & +2x_2 & +x_3 & & = -1 & \\
 & +2x_2 & +4x_3 & -x_4 & = 1 & \\
 & +4x_2 & +5x_3 & +3x_4 & = 4 & \quad L_4 \leftarrow L_4 - 2L_2
\end{array}$$

$$\Rightarrow \quad \begin{array}{lllll}
x_1 & +2x_2 & & +x_4 & = 0 \\
 & +2x_2 & +x_3 & & = -1 \\
 & & +3x_3 & -x_4 & = 2 \\
 & & +3x_3 & +3x_4 & = 6
\end{array}$$

L'algorithme de Gauss

Principe de l'algorithme

- Étape 3 :

$$\begin{array}{rrrrl} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & +3x_3 & -x_4 & = 2 \\ & & +3x_3 & +3x_4 & = 6 \end{array}$$

$$\Rightarrow \quad \begin{array}{rrrrl} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & +3x_3 & -x_4 & = 2 \end{array}$$

L'algorithme de Gauss

Principe de l'algorithme

- Étape 3 :

$$\begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & \color{blue}{+3x_3} & \color{blue}{-x_4} & \color{blue}{= 2} \\ & & \color{red}{+3x_3} & \color{red}{+3x_4} & \color{red}{= 6} \qquad \color{red}{L_4 \leftarrow L_4 -} \color{blue}{L_3} \end{array}$$

$$\Rightarrow \quad \begin{array}{lllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & +3x_3 & -x_4 & = 2 \\ & & & \color{red}{+4x_4} & \color{red}{= 4} \end{array}$$

- Dernière étape dite "remontée"

$$\begin{array}{llllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & +3x_3 & -x_4 & = 2 \\ & & & +4x_4 & = 4 \end{array}$$

$$x_4 = 1$$

- Dernière étape dite "remontée"

$$
\begin{array}{lllll}
x_1 & +2x_2 & & +x_4 & = 0 \\
 & +2x_2 & +x_3 & & = -1 \\
 & & +3x_3 & -x_4 & = 2 \\
 & & & +4x_4 & = 4
\end{array}
$$

$$x_4 = 1, \quad x_3 = 1$$

- Dernière étape dite "remontée"

$$\begin{array}{llllll} x_1 & +2x_2 & & +x_4 & = 0 \\ & +2x_2 & +x_3 & & = -1 \\ & & +3x_3 & -x_4 & = 2 \\ & & & +4x_4 & = 4 \end{array}$$

$$x_4 = 1, \quad x_3 = 1, \quad x_2 = -1$$

- Dernière étape dite "remontée"

$$
\begin{array}{ccccc}
x_1 & +2x_2 & & +x_4 & = 0 \\
 & +2x_2 & +x_3 & & = -1 \\
 & & +3x_3 & -x_4 & = 2 \\
 & & & +4x_4 & = 4
\end{array}
$$

$$x_4 = 1, \quad x_3 = 1, \quad x_2 = -1, \quad x_1 = 1$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{1} & 2 & 0 & 1 \\ 2 & 6 & 1 & 2 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix}, \ b^{(1)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix}$$

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ & & & \\ & & & \\ & & & \end{pmatrix}, \ b^{(2)} = \begin{pmatrix} 0 \\ \\ \\ \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{1} & 2 & 0 & 1 \\ 2 & 6 & 1 & 2 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix}, \; b^{(1)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix} \quad L_2 \leftarrow L_2 - 2L_1$$

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ & & & \\ & & & \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 0 \\ -1 \\ \\ \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{1} & 2 & 0 & 1 \\ 2 & 6 & 1 & 2 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix}, \; b^{(1)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix} \quad L_3 \leftarrow L_3 + L_1$$

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 2 & 4 & -1 \\ & & & \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ \ \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{1} & 2 & 0 & 1 \\ 2 & 6 & 1 & 2 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix}, \; b^{(1)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix} \quad L_4 \leftarrow L_4 - L_1$$

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 2 & 4 & -1 \\ 0 & 4 & 5 & 3 \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 2 :

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & \boxed{2} & 1 & 0 \\ 0 & 2 & 4 & -1 \\ 0 & 4 & 5 & 3 \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix}$$

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ & & & \\ & & & \end{pmatrix}, \; b^{(3)} = \begin{pmatrix} 0 \\ -1 \\ \\ \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 2 :

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & \boxed{2} & 1 & 0 \\ 0 & 2 & 4 & -1 \\ 0 & 4 & 5 & 3 \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix} \quad L_3 \leftarrow L_3 - L_2$$

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & -1 \\ & & & \end{pmatrix}, \; b^{(3)} = \begin{pmatrix} 0 \\ -1 \\ 2 \\ \ \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 2 :

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & \boxed{2} & 1 & 0 \\ 0 & 2 & 4 & -1 \\ 0 & 4 & 5 & 3 \end{pmatrix}, \; b^{(2)} = \begin{pmatrix} 0 \\ -1 \\ 1 \\ 4 \end{pmatrix} \quad L_4 \leftarrow L_4 - 2L_2$$

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 3 & 3 \end{pmatrix}, \; b^{(3)} = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 6 \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 3 :

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & \boxed{3} & -1 \\ 0 & 0 & 3 & 3 \end{pmatrix}, \; b^{(3)} = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 6 \end{pmatrix}$$

$$A^{(4)} = U = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & -1 \\ & & & \end{pmatrix}, \; b^{(4)} = y = \begin{pmatrix} 0 \\ -1 \\ 2 \\ \end{pmatrix}$$

L'algorithme de Gauss

Principe de l'algorithme (sous forme matricielle)

- Étape 3 :

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & \boxed{3} & -1 \\ 0 & 0 & 3 & 3 \end{pmatrix}, \quad b^{(3)} = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 6 \end{pmatrix} \quad L_4 \leftarrow L_4 - L_3$$

$$A^{(4)} = U = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 4 \end{pmatrix}, \quad b^{(4)} = y = \begin{pmatrix} 0 \\ -1 \\ 2 \\ 4 \end{pmatrix}$$

L'algorithme de Gauss

Étape k pour une matrice $n \times n$

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \cdots & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & \boxed{a_{kk}^{(k)}} & \cdots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{ik}^{(k)} & \cdots & a_{in}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}$$

L'algorithme de Gauss

Étape k pour une matrice $n \times n$

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \cdots & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & \boxed{a_{kk}^{(k)}} & \cdots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{ik}^{(k)} & \cdots & a_{in}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}$$

Si le **pivot** $a_{kk}^{(k)} \neq 0$

L'algorithme de Gauss

Étape k pour une matrice $n \times n$

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \cdots & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & \boxed{a_{kk}^{(k)}} & \cdots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{ik}^{(k)} & \cdots & a_{in}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix} \quad L_i \leftarrow L_i - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} L_k$$

Si le **pivot** $a_{kk}^{(k)} \neq 0$, alors pour $i = k+1, \ldots, n$:

L'algorithme de Gauss

Étape k pour une matrice $n \times n$

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \cdots & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & \boxed{a_{kk}^{(k)}} & \cdots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{ik}^{(k)} & \cdots & a_{in}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix} \quad L_i \leftarrow L_i - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} L_k$$

Si le **pivot** $a_{kk}^{(k)} \neq 0$, alors pour $i = k+1, \ldots, n$:

$$a_{ik}^{(k+1)} = 0 \text{ et } a_{ij}^{(k+1)} = a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} a_{kj}^{(k)}, \text{ pour } j = k+1, \ldots, n.$$

L’algorithme de Gauss

pour $k = 1$ jusqu’à $n - 1$ **faire**

fin pour

L'algorithme de Gauss

pour $k = 1$ jusqu'à $n - 1$ **faire**
  si $|a_{kk}| < \varepsilon$ **alors**
    Arrêter l'algorithme et donner un message d'erreur
  sinon

  fin si
fin pour

L'algorithme de Gauss

pour $k = 1$ jusqu'à $n - 1$ **faire**
    si $|a_{kk}| < \varepsilon$ **alors**
        Arrêter l'algorithme et donner un message d'erreur
    sinon
        pour $i = k + 1$ jusqu'à n **faire**

        fin pour
    fin si
fin pour

L'algorithme de Gauss

pour $k = 1$ jusqu'à $n - 1$ **faire**
  si $|a_{kk}| < \varepsilon$ **alors**
    Arrêter l'algorithme et donner un message d'erreur
  sinon
    pour $i = k + 1$ jusqu'à n **faire**
      $c \leftarrow a_{ik} / a_{kk}$
      $a_{ik} \leftarrow 0$
      $b_i \leftarrow b_i - c b_k$

    fin pour
  fin si
fin pour

L'algorithme de Gauss

pour $k = 1$ jusqu'à $n - 1$ **faire**
    si $|a_{kk}| < \varepsilon$ **alors**
        Arrêter l'algorithme et donner un message d'erreur
    sinon
        pour $i = k + 1$ jusqu'à n **faire**
            $c \leftarrow a_{ik}/a_{kk}$
            $a_{ik} \leftarrow 0$
            $b_i \leftarrow b_i - cb_k$
            pour $j = k + 1$ jusqu'à n **faire**
                $a_{ij} \leftarrow a_{ij} - ca_{kj}$
            fin pour
        fin pour
    fin si
fin pour

L’algorithme de Gauss

pour $k = 1$ jusqu’à $n - 1$ **faire**
  si $|a_{kk}| < \varepsilon$ **alors**
    Arrêter l’algorithme et donner un message d’erreur
  sinon
    pour $i = k + 1$ jusqu’à n **faire**
      $c \leftarrow a_{ik}/a_{kk}$
      $a_{ik} \leftarrow 0$
      $b_i \leftarrow b_i - cb_k$
      pour $j = k + 1$ jusqu’à n **faire**
        $a_{ij} \leftarrow a_{ij} - ca_{kj}$
      fin pour
    fin pour
  fin si
fin pour
si $|a_{nn}| < \varepsilon$ **alors**
  Arrêter l’algorithme et donner un message d’erreur
fin si

L'algorithme de Gauss

Nombre d'opérations

Quel est le nombre d'opérations de l'algorithme de Gauss ?

Quel est le nombre d'opérations de l'algorithme de Gauss ?

Proposition

Le nombre d'opérations pour de l'algorithme de Gauss (élimination et remontée) est de l'ordre de n^3.

Quel est le nombre d'opérations de l'algorithme de Gauss ?

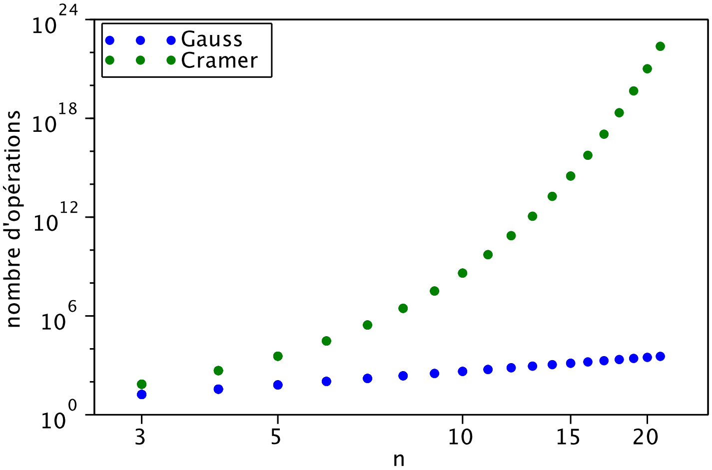
Proposition

Le nombre d'opérations pour de l'algorithme de Gauss (élimination et remontée) est de l'ordre de n^3.

- Pour $n = 16$, avec la remontée, on obtient moins de 2000 opérations et un temps de calcul de 0,016μs

L'algorithme de Gauss

Nombre d'opérations



L'algorithme de Gauss

Permutation de lignes

- L'algorithme de Gauss échoue dès qu'il existe k tel que $a_{kk}^{(k)} = 0$

$$A = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 2 \\ 3 & 1 & 1 \end{pmatrix}, \; b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

L'algorithme de Gauss

Permutation de lignes

- L'algorithme de Gauss échoue dès qu'il existe k tel que $a_{kk}^{(k)} = 0$

$$A = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 2 \\ 3 & 1 & 1 \end{pmatrix}, \, b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

- Il suffit d'échanger les lignes {1,2}

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

L'algorithme de Gauss

Permutation de lignes

- L'algorithme de Gauss échoue dès qu'il existe k tel que $a_{kk}^{(k)} = 0$

$$A = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 1 & 2 \\ 3 & 1 & 1 \end{pmatrix},\ b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

- Il suffit d'échanger les lignes {1,2} ou {1,3}

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

L'algorithme de Gauss

Permutations de lignes

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \dots & a_{1k}^{(k)} & \dots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{ik}^{(k)} & \dots & a_{in}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix}$$

Proposition

Si $\det A \neq 0$ alors

$$\forall k \in \{1, \dots, n\}, \ \exists i \in \{k, \dots, n\}, \ a_{ik}^{(k)} \neq 0.$$

L'algorithme de Gauss

Permutations de lignes

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \dots & a_{1k}^{(k)} & \dots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & {\color{blue}0} & {\color{blue}\dots} & {\color{blue}a_{kn}^{(k)}} \\ & & \vdots & & \vdots \\ & & {\color{red}0} & {\color{red}\dots} & {\color{red}a_{in}^{(k)}} \\ & & \vdots & & \vdots \\ & & 0 & \dots & a_{nn}^{(k)} \end{pmatrix}$$

Proposition

Si $\det A \neq 0$ alors

$$\forall k \in \{1, \dots, n\}, \exists i \in \{k, \dots, n\}, a_{ik}^{(k)} \neq 0.$$

L'algorithme de Gauss

Permutations de lignes

$$A^{(k)} = \begin{pmatrix} a_{11}^{(k)} & \dots & a_{1k}^{(k)} & \dots & a_{1n}^{(k)} \\ & \ddots & & & \vdots \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{mk}^{(k)} & \dots & a_{mn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{pmatrix} \begin{array}{l} \\ \\ L_k \leftarrow L_m \\ \\ L_m \leftarrow L_k \\ \\ \\ \end{array}$$

En pratique, on permute systématiquement les lignes $\{k, m\}$ où

$$\forall i \in \{k, \dots, n\}, \ |a_{m,k}| \geq |a_{i,k}|$$

L'algorithme de Gauss

Permutation de lignes

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ \boxed{2} & 6 & 1 & 2 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix} \begin{array}{l} L_1 \leftarrow L_2 \\ L_2 \leftarrow L_1 \\ \\ \\ \end{array}$$

$$A^{(2)} = \begin{pmatrix} 2 & 6 & 1 & 2 \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

L'algorithme de Gauss

Permutation de lignes

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{2} & 6 & 1 & 2 \\ 1 & 2 & 0 & 1 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix} \quad L_2 \leftarrow L_2 - \frac{1}{2}L_1$$

$$A^{(2)} = \begin{pmatrix} 2 & 6 & 1 & 2 \\ 0 & -1 & -\frac{1}{2} & 0 \\ & & & \\ & & & \end{pmatrix}$$

L'algorithme de Gauss

Permutation de lignes

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{2} & 6 & 1 & 2 \\ 1 & 2 & 0 & 1 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix} \quad L_3 \leftarrow L_3 + \frac{1}{2} L_1$$

$$A^{(2)} = \begin{pmatrix} 2 & 6 & 1 & 2 \\ 0 & -1 & -\frac{1}{2} & 0 \\ 0 & 3 & \frac{9}{2} & -1 \\ & & & \end{pmatrix}$$

L'algorithme de Gauss

Permutation de lignes

- Étape 1 :

$$A^{(1)} = \begin{pmatrix} \boxed{2} & 6 & 1 & 2 \\ 1 & 2 & 0 & 1 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix} \quad L_4 \leftarrow L_4 - \frac{1}{2} L_1$$

$$A^{(2)} = \begin{pmatrix} 2 & 6 & 1 & 2 \\ 0 & -1 & -\frac{1}{2} & 0 \\ 0 & 3 & \frac{9}{2} & -1 \\ 0 & 3 & \frac{5}{2} & 3 \end{pmatrix}$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

Notations

On note :

- $U = A^{(n)}$,
- $\underline{A}_i$ la ligne numéro i de la matrice A,
- $\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ utilisé pour éliminer x_k de l'équation $i > k$.

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

Notations

On note :

- $U = A^{(n)}$,
- $\underline{A}_i$ la ligne numéro i de la matrice A,
- $\ell_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ utilisé pour éliminer x_k de l'équation $i > k$.

- Pour $n = 2$ les seules opérations effectuées sont :

$$\underline{A}_2^{(1)} = \underline{A}_2$$
$$\underline{A}_2^{(2)} = \underline{A}_2^{(1)} - \ell_{21}\underline{A}_1^{(1)}$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

Notations

On note :

- $U = A^{(n)}$,
- $\underline{A}_i$ la ligne numéro i de la matrice A,
- $\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ utilisé pour éliminer x_k de l'équation $i > k$.

- Pour $n = 2$ les seules opérations effectuées sont :

$$\begin{aligned} \underline{A}_2^{(1)} &= \underline{A}_2 \\ \underline{A}_2^{(2)} &= \underline{A}_2^{(1)} - \ell_{21}\underline{A}_1^{(1)} \end{aligned} \qquad \Rightarrow \qquad \underline{A}_1 = \underline{U}_1, \quad \underline{A}_2 = \ell_{21}\underline{U}_1 + \underline{U}_2$$

Notations

On note :

- $U = A^{(n)}$,
- $\underline{A}_i$ la ligne numéro i de la matrice A,
- $\ell_{ik} = \dfrac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$ utilisé pour éliminer x_k de l'équation $i > k$.

- Pour $n = 2$ les seules opérations effectuées sont :

$$\begin{aligned} \underline{A}_2^{(1)} &= \underline{A}_2 \\ \underline{A}_2^{(2)} &= \underline{A}_2^{(1)} - \ell_{21}\underline{A}_1^{(1)} \end{aligned} \quad \Rightarrow \quad \underline{A}_1 = \underline{U}_1, \quad \underline{A}_2 = \ell_{21}\underline{U}_1 + \underline{U}_2$$

Relations valables pour tout n

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$\underline{A}_3^{(1)} = \underline{A}_3$$
$$\underline{A}_3^{(2)} = \underline{A}_3^{(1)} - \ell_{31}\underline{A}_1^{(1)}$$
$$\underline{A}_3^{(3)} = \underline{A}_3^{(2)} - \ell_{32}\underline{A}_2^{(2)}$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$
\begin{aligned}
\cancel{\underline{A}_3^{(1)}} &= \underline{A}_3 \\
+\underline{A}_3^{(2)} &= \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)} \\
\underline{A}_3^{(3)} &= \underline{A}_3^{(2)} - \ell_{32}\underline{A}_2^{(2)}
\end{aligned}
$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$\cancel{\underline{A}_3^{(1)}} = \underline{A}_3$$
$$+\cancel{\underline{A}_3^{(2)}} = \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)}$$
$$+\underline{A}_3^{(3)} = \cancel{\underline{A}_3^{(2)}} - \ell_{32}\underline{A}_2^{(2)}$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$
\begin{aligned}
&\cancel{\underline{A}_3^{(1)}} = \underline{A}_3 \\
+&\cancel{\underline{A}_3^{(2)}} = \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)} \\
+&\underline{A}_3^{(3)} = \cancel{\underline{A}_3^{(2)}} - \ell_{32}\underline{A}_2^{(2)}
\end{aligned}
$$

$$\underline{A}_3^{(3)} = \underline{A}_3^{(1)} - \ell_{31}\underline{A}_1^{(1)} - \ell_{32}\underline{A}_2^{(2)}$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$
\begin{aligned}
\cancel{\underline{A}_3^{(1)}} &= \underline{A}_3 \\
+\cancel{\underline{A}_3^{(2)}} &= \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)} \\
+\underline{A}_3^{(3)} &= \cancel{\underline{A}_3^{(2)}} - \ell_{32}\underline{A}_2^{(2)}
\end{aligned}
$$

$$
\begin{aligned}
\underline{A}_3^{(3)} &= \underline{A}_3^{(1)} - \ell_{31}\underline{A}_1^{(1)} - \ell_{32}\underline{A}_2^{(2)} \\
\underline{U}_3 &= \underline{A}_3 - \ell_{31}\underline{U}_1 - \ell_{32}\underline{U}_2
\end{aligned}
$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$\begin{aligned} \cancel{\underline{A}_3^{(1)}} &= \underline{A}_3 \\ +\cancel{\underline{A}_3^{(2)}} &= \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)} \\ +\underline{A}_3^{(3)} &= \cancel{\underline{A}_3^{(2)}} - \ell_{32}\underline{A}_2^{(2)} \end{aligned}$$

$$\begin{aligned} \underline{A}_3^{(3)} &= \underline{A}_3^{(1)} - \ell_{31}\underline{A}_1^{(1)} - \ell_{32}\underline{A}_2^{(2)} \\ \underline{U}_3 &= \underline{A}_3 - \ell_{31}\underline{U}_1 - \ell_{32}\underline{U}_2 \\ \underline{A}_3 &= \ell_{31}\underline{U}_1 + \ell_{32}\underline{U}_2 + \underline{U}_3 \end{aligned}$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$\cancel{\underline{A}_3^{(1)}} = \underline{A}_3$$
$$+\cancel{\underline{A}_3^{(2)}} = \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)}$$
$$+\underline{A}_3^{(3)} = \cancel{\underline{A}_3^{(2)}} - \ell_{32}\underline{A}_2^{(2)}$$

$$\underline{A}_3^{(3)} = \underline{A}_3^{(1)} - \ell_{31}\underline{A}_1^{(1)} - \ell_{32}\underline{A}_2^{(2)}$$
$$\underline{U}_3 = \underline{A}_3 - \ell_{31}\underline{U}_1 - \ell_{32}\underline{U}_2$$
$$\underline{A}_3 = \ell_{31}\underline{U}_1 + \ell_{32}\underline{U}_2 + \underline{U}_3$$

$$\underline{A}_1 = \underline{U}_1,$$
$$\underline{A}_2 = \ell_{21}\underline{U}_1 + \underline{U}_2,$$
$$\underline{A}_3 = \ell_{31}\underline{U}_1 + \ell_{32}\underline{U}_2 + \underline{U}_3$$

La factorisation $A = LU$

Ecriture matricielle de l'élimination de Gauss

- Pour $n = 3$ les opérations effectuées sur la ligne 3 sont

$$\cancel{\underline{A}_3^{(1)}} = \underline{A}_3$$
$$+\cancel{\underline{A}_3^{(2)}} = \cancel{\underline{A}_3^{(1)}} - \ell_{31}\underline{A}_1^{(1)}$$
$$+\underline{A}_3^{(3)} = \cancel{\underline{A}_3^{(2)}} - \ell_{32}\underline{A}_2^{(2)}$$

$$\underline{A}_3^{(3)} = \underline{A}_3^{(1)} - \ell_{31}\underline{A}_1^{(1)} - \ell_{32}\underline{A}_2^{(2)}$$
$$\underline{U}_3 = \underline{A}_3 - \ell_{31}\underline{U}_1 - \ell_{32}\underline{U}_2$$
$$\underline{A}_3 = \ell_{31}\underline{U}_1 + \ell_{32}\underline{U}_2 + \underline{U}_3$$

$$\begin{aligned} \underline{A}_1 &= \underline{U}_1, \\ \underline{A}_2 &= \ell_{21}\underline{U}_1 + \underline{U}_2, \\ \underline{A}_3 &= \ell_{31}\underline{U}_1 + \ell_{32}\underline{U}_2 + \underline{U}_3 \end{aligned}$$

$$A = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} \underline{U}_1 \\ \underline{U}_2 \\ \underline{U}_3 \end{pmatrix}$$

La factorisation $A = LU$

Proposition

Soit A une matrice $n \times n$ inversible. Si l'algorithme de Gauss peut être appliqué à la matrice A sans permutations de lignes, alors il existe une une factorisation unique

$$A = LU,$$

où $U = A^{(n)}$ et L est une matrice triangulaire inférieure dont les termes diagonaux sont égaux à 1.

La factorisation $A = LU$

Proposition

Soit A une matrice $n \times n$ inversible. Si l'algorithme de Gauss peut être appliqué à la matrice A sans permutations de lignes, alors il existe une une factorisation unique

$$A = LU,$$

où $U = A^{(n)}$ et L est une matrice triangulaire inférieure dont les termes diagonaux sont égaux à 1.

Démonstration

On montre que

$$\underline{A}_j = \sum_{k=1}^{j-1} \ell_{jk} \underline{U}_k + \underline{U}_j, \; j = 1, \ldots n, \;\text{ où } \ell_{jk} = \frac{a_{jk}^{(k)}}{a_{kk}^{(k)}}.$$

Unicité ?

La factorisation $A = LU$

Unicité

Nous avons besoin des deux résultats préliminaires suivants :

Lemme

Soient A et B deux matrices triangulaires inférieures. Leur produit AB est triangulaire inférieure et on a

$$(AB)_{ii} = a_{ii} b_{ii}.$$

Lemme

Soit A une matrice triangulaire inférieure inversible. Alors A^{-1} est triangulaire inférieure et

$$(A^{-1})_{ii} = \frac{1}{a_{ii}}.$$

La factorisation $A = LU$

Algorithme

pour $k = 1$ jusqu'à $n - 1$ **faire**
    si $|a_{kk}| < \varepsilon$ **alors**
        Arrêter l'algorithme et donner un message d'erreur
    sinon
        pour $i = k + 1$ jusqu'à n **faire**
            $\ell_{ik} \leftarrow a_{ik} / a_{kk}$
            $a_{ik} \leftarrow 0$
            pour $j = k + 1$ jusqu'à n **faire**
                $a_{ij} \leftarrow a_{ij} - \ell_{ik} a_{kj}$
            fin pour
        fin pour
    fin si
fin pour
si $|a_{nn}| < \varepsilon$ **alors**
    Arrêter l'algorithme et donner un message d'erreur
fin si

La factorisation $A = LU$

Algorithme

6: $\ell_{ik} \leftarrow a_{ik} / a_{kk}$

La factorisation $A = LU$

Exemple

Étape 1

$$A^{(1)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ 2 & 6 & 1 & 2 \\ -1 & 0 & 4 & -2 \\ 1 & 6 & 5 & 4 \end{pmatrix} \begin{array}{l} L_2 \leftarrow L_2 - 2L_1 \\ L_3 \leftarrow L_3 + 1L_1 \\ L_4 \leftarrow L_4 - 1L_1 \end{array}$$

$$L = \begin{pmatrix} 1 & & & \\ 2 & 1 & & \\ -1 & & 1 & \\ 1 & & & 1 \end{pmatrix}$$

La factorisation $A = LU$

Exemple

Étape 2

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ & 2 & 1 & 0 \\ & 2 & 4 & -1 \\ & 4 & 5 & 3 \end{pmatrix} \quad \begin{array}{l} \\ \\ L_3 \leftarrow L_3 - 1 L_2 \\ L_4 \leftarrow L_4 - 2 L_2 \end{array}$$

$$L = \begin{pmatrix} 1 & & & \\ 2 & 1 & & \\ -1 & 1 & 1 & \\ 1 & 2 & & 1 \end{pmatrix}$$

La factorisation $A = LU$

Exemple

Étape 3

$$A^{(3)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ & 2 & 1 & 0 \\ & & \color{blue}{3} & \color{blue}{-1} \\ & & 3 & 3 \end{pmatrix} \quad L_4 \leftarrow L_4 \color{red}{-1} \color{blue}{L_3}$$

$$L = \begin{pmatrix} 1 & & & \\ 2 & 1 & & \\ -1 & 1 & 1 & \\ 1 & 2 & \color{red}{1} & 1 \end{pmatrix}$$

La factorisation $A = LU$

Exemple

Étape 4

$$U = A^{(4)} = \begin{pmatrix} 1 & 2 & 0 & 1 \\ & 2 & 1 & 0 \\ & & 3 & -1 \\ & & & 4 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & & & \\ 2 & 1 & & \\ -1 & 1 & 1 & \\ 1 & 2 & 1 & 1 \end{pmatrix}$$

La factorisation $A = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff LUx = b \\ &\iff Ly = b,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $A = LU$

Utilisation pratique

$$Ax = b \iff LUx = b$$
$$\iff Ly = b,\ Ux = y$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $A = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\Longleftrightarrow LUx = b \\ &\Longleftrightarrow Ly = b,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $A = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff LUx = b \\ &\iff Ly = b,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $A = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\Longleftrightarrow LUx = b \\ &\Longleftrightarrow Ly = b,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $A = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff LUx = b \\ &\iff Ly = b,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $A = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff LUx = b \\ &\iff Ly = b,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = b$ « descente »
3. Résolution de $Ux = y$ « remontée »

Pour résoudre le système avec un nouveau second membre, seules la « descente » et la « remontée » sont nécessaires !

La factorisation $PA = LU$

Matrice de permutation

- Soit σ une permutation de $\{1, 2, \dots, n\}$ et $\{e_1, \dots, e_n\}$ la base canonique de $\mathbb{R}^n$. La matrice de permutation P associée à σ est définie par

$$P = \left(e_{\sigma(1)}\ e_{\sigma(2)} \dots e_{\sigma(n)}\right)$$

Proposition

Si A est inversible alors il existe U triangulaire supérieure, L une matrice triangulaire inférieure dont les termes diagonaux sont égaux à 1 et P une matrice de permutation tels que

$$PA = LU.$$

Attention : la factorisation n'est plus unique
et P n'est pas connue à l'avance !

La factorisation $PA = LU$

Matrice de permutation

- Soit σ une permutation de $\{1, 2, \dots, n\}$ et $\{e_1, \dots, e_n\}$ la base canonique de $\mathbb{R}^n$. La matrice de permutation P associée à σ est définie par

$$P = \left(e_{\sigma(1)}\ e_{\sigma(2)} \dots e_{\sigma(n)}\right)$$

Proposition

Si A est inversible alors il existe U triangulaire supérieure, L une matrice triangulaire inférieure dont les termes diagonaux sont égaux à 1 et P une matrice de permutation tels que

$$PA = LU.$$

Attention : la factorisation n'est plus unique
et P n'est pas connue à l'avance !

La factorisation $PA = LU$
Exemple

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

- Permutation : $\sigma = \{1, 2, 3\}$

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

Initialement, il n'y a pas de permutation

La factorisation $PA = LU$

Exemple

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

- Permutation : $\sigma = \{3, 2, 1\}$

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

Étape 1 : échange des lignes 1 et 3

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

- Permutation : $\sigma = \{3, 2, 1\}$

$$\begin{pmatrix} 3 & 2 & 1 \\ 1 & 1 & 2 \\ 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \frac{1}{3} & 1 & \\ 0 & & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \\ & \frac{1}{3} & \frac{5}{3} \\ & 1 & 1 \end{pmatrix}$$

Étape 1 : élimination

La factorisation $PA = LU$

Exemple

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

- Permutation : $\sigma = \{3, 1, 2\}$

$$\begin{pmatrix} 3 & 2 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 1 & & \\ 0 & 1 & \\ \frac{1}{3} & & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 & 1 \\ & 1 & 1 \\ & \frac{1}{3} & \frac{5}{3} \end{pmatrix}$$

Étape 2 : échange des lignes 2 et 3

La factorisation $PA = LU$

Exemple

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

- Permutation : $\sigma = \{3, 1, 2\}$

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

Étape 2 : élimination

La factorisation $PA = LU$

Exemple

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}$$

- Permutation : $\sigma = \{3, 1, 2\}$

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

Fin de l'algorithme !

La factorisation $PA = LU$

Utilisation pratique

$$Ax = b$$

La factorisation $PA = LU$

Utilisation pratique

$$Ax = b \Longleftrightarrow PAx = Pb$$

La factorisation $PA = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff PAx = Pb \\ &\iff LUx = Pb \end{aligned}$$

La factorisation $PA = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff PAx = Pb \\ &\iff LUx = Pb \\ &\iff Ly = Pb,\ Ux = y \end{aligned}$$

La factorisation $PA = LU$

Utilisation pratique

$$
\begin{aligned}
Ax = b &\iff PAx = Pb \\
&\iff LUx = Pb \\
&\iff Ly = Pb, \; Ux = y
\end{aligned}
$$

1. Factorisation de A

La factorisation $PA = LU$

Utilisation pratique

$$
\begin{aligned}
Ax = b &\iff PAx = Pb \\
&\iff LUx = Pb \\
&\iff Ly = Pb,\ Ux = y
\end{aligned}
$$

1. Factorisation de A
2. Résolution de $Ly = Pb$ « descente »

La factorisation $PA = LU$

Utilisation pratique

$$\begin{aligned} Ax = b &\iff PAx = Pb \\ &\iff LUx = Pb \\ &\iff Ly = Pb,\ Ux = y \end{aligned}$$

1. Factorisation de A
2. Résolution de $Ly = Pb$ « descente »
3. Résolution de $Ux = y$ « remontée »

La factorisation $PA = LU$

Dans Scilab

```
-->A=[0 1 1;1 1 2;3 2 1]

-->[L,U,P]=lu(A)
 P  =

    0.    0.    1.
    1.    0.    0.
    0.    1.    0.
 U  =

    3.    2.    1.
    0.    1.    1.
    0.    0.    1.3333333
 L  =

    1.            0.            0.
    0.            1.            0.
    0.3333333     0.3333333     1.
```

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

comme une équation.

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

  comme une équation.

- Exemple pour $n = 3$:

$$\begin{pmatrix} 2 & 1 & -2 \\ 4 & 5 & -3 \\ -2 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{pmatrix}$$

  On identifie, dans l'ordre :

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

comme une équation.

- Exemple pour $n = 3$:

$$\begin{pmatrix} 2 & 1 & -2 \\ 4 & 5 & -3 \\ -2 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{pmatrix}$$

On identifie, dans l'ordre :

- $\underline{A}_1$ et $\underline{L}_1 U$ puis A_1 et LU_1

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

comme une équation.

- Exemple pour $n = 3$:

$$\begin{pmatrix} 2 & 1 & -2 \\ 4 & 5 & -3 \\ -2 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{pmatrix}$$

On identifie, dans l'ordre :

- $\underline{A}_1$ et $\underline{L}_1 U$ puis A^1 et LU^1

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

  comme une équation.

- Exemple pour $n = 3$:

$$\begin{pmatrix} 2 & 1 & -2 \\ 4 & 5 & -3 \\ -2 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{pmatrix}$$

  On identifie, dans l'ordre :

  - $\underline{A}_1$ et $\underline{L}_1 U$ puis A_1 et LU_1
  - $\underline{A}_2$ et $\underline{L}_2 U$ puis A_2 et LU_2

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

comme une équation.

- Exemple pour $n = 3$:

$$\begin{pmatrix} 2 & 1 & -2 \\ 4 & 5 & -3 \\ -2 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{pmatrix}$$

On identifie, dans l'ordre :

- $\underline{A}_1$ et $\underline{L}_1 U$ puis A_1 et LU_1
- $\underline{A}_2$ et $\underline{L}_2 U$ puis A_2 et LU_2

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

- But : identifier les coefficients de L et U en considérant

$$A = LU$$

comme une équation.

- Exemple pour $n = 3$:

$$\begin{pmatrix} 2 & 1 & -2 \\ 4 & 5 & -3 \\ -2 & 5 & 3 \end{pmatrix} = \begin{pmatrix} 1 & & \\ \ell_{21} & 1 & \\ \ell_{31} & \ell_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ & u_{22} & u_{23} \\ & & u_{33} \end{pmatrix}$$

On identifie, dans l'ordre :

  - $\underline{A}_1$ et $\underline{L}_1 U$ puis A_1 et LU_1
  - $\underline{A}_2$ et $\underline{L}_2 U$ puis A_2 et LU_2
  - $\underline{A}_3$ et $\underline{L}_3 U$

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

pour $i = 1$ jusqu'à n **faire**
    pour $j = i$ jusqu'à n **faire**
    fin pour
    pour $j = i + 1$ jusqu'à n **faire**
    fin pour
fin pour

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

```
pour i = 1 jusqu'à n faire
    pour j = i jusqu'à n faire
        Identifier a_ij et (LU)_ij, en déduire u_ij
    fin pour
    pour j = i + 1 jusqu'à n faire

    fin pour
fin pour
```

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

```
pour i = 1 jusqu'à n faire
    pour j = i jusqu'à n faire
        u_ij ← a_ij − Σ_{k=1}^{i−1} ℓ_ik u_kj
    fin pour
    pour j = i + 1 jusqu'à n faire

    fin pour
fin pour
```

pour $i = 1$ jusqu'à n **faire**

  pour $j = i$ jusqu'à n **faire**

    $u_{ij} \leftarrow a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} u_{kj}$

  fin pour

  pour $j = i + 1$ jusqu'à n **faire**

  fin pour

fin pour

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

pour $i = 1$ jusqu'à n **faire**
    pour $j = i$ jusqu'à n **faire**
        $u_{ij} \leftarrow a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} u_{kj}$
    fin pour
    pour $j = i + 1$ jusqu'à n **faire**
        Identifier a_{ji} et $(LU)_{ji}$, en déduire ℓ_{ji}
    fin pour
fin pour

Calcul direct de la factorisation $A = LU$

Algorithme de Doolittle

pour $i = 1$ jusqu'à n **faire**
  pour $j = i$ jusqu'à n **faire**
    $u_{ij} \leftarrow a_{ij} - \sum_{k=1}^{i-1} \ell_{ik} u_{kj}$
  fin pour
  pour $j = i + 1$ jusqu'à n **faire**
    $\ell_{ji} \leftarrow (a_{ji} - \sum_{k=1}^{i-1} \ell_{jk} u_{ki}) / u_{ii}$
  fin pour
fin pour

Calcul direct de la factorisation $A = LU$

Conditions

Définition

Soit A une matrice $n \times n$, on appelle sous-matrice principale d'ordre k de la matrice A et on note $[A]_k$ la matrice $k \times k$

$$[A]_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}$$

Calcul direct de la factorisation $A = LU$

Conditions

Définition

Soit A une matrice $n \times n$, on appelle sous-matrice principale d'ordre k de la matrice A et on note $[A]_k$ la matrice $k \times k$

$$[A]_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & \ddots & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix}$$

Théorème

Soit A une matrice $n \times n$ inversible. Si toutes les sous-matrices principales de A sont régulières alors la factorisation LU de A est possible sans échange de lignes.

La factorisation $A = LDL^\top$

Existence, algorithme

Théorème

Soit A une matrice $n \times n$ symétrique et inversible. Si toutes les sous-matrices principales de A sont régulières alors elle admet une décomposition unique sous la forme

$$A = LDL^\top,$$

où L est triangulaire inférieure à diagonale unité et D est une matrice diagonale.

La factorisation $A = LDL^\top$

Existence, algorithme

Théorème

Soit A une matrice $n \times n$ symétrique et inversible. Si toutes les sous-matrices principales de A sont régulières alors elle admet une décomposition unique sous la forme

$$A = LDL^\top,$$

où L est triangulaire inférieure à diagonale unité et D est une matrice diagonale.

- Algorithme : on identifie A_j et $(LDL^\top)_j$ pour $j = 1, \ldots, n$

La factorisation $A = BB^T$

Matrices définies positives

Définition

Soit A une matrice $n \times n$ symétrique. La matrice A est définie positive si

$$\forall x \in \mathbb{R}^n,\ x \neq 0 \Longrightarrow x^\top A x > 0$$

La factorisation $A = BB^T$

Matrices définies positives

Définition

Soit A une matrice $n \times n$ symétrique. La matrice A est définie positive si

$$\forall x \in \mathbb{R}^n,\ x \neq 0 \Longrightarrow x^\top A x > 0$$

Proposition

Une condition nécessaire et suffisante pour que A (symétrique) soit définie positive (respectivement semi-définie positive) est que toute ses valeurs propres soit strictement positives (respectivement positives ou nulles).

La factorisation $A = BB^T$

Matrices définies positives

Définition

Soit A une matrice $n \times n$ symétrique. La matrice A est définie positive si

$$\forall x \in \mathbb{R}^n,\ x \neq 0 \Longrightarrow x^\top A x > 0$$

Proposition

Une condition nécessaire et suffisante pour que A (symétrique) soit définie positive (respectivement semi-définie positive) est que toute ses valeurs propres soit strictement positives (respectivement positives ou nulles).

Exemple

$$A = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

La factorisation $A = BB^T$

Factorisation de Cholesky, existence, unicité

Théorème

Soit A une matrice $n \times n$ symétrique. Si A est définie positive alors toutes ses sous-matrices principales sont régulières

Théorème

Soit A une matrice $n \times n$ symétrique. Si A est définie positive alors toutes ses sous-matrices principales sont régulières

Théorème

Soit A une matrice $n \times n$ symétrique. Si A est définie positive alors il existe une unique matrice B triangulaire inférieure ayant des éléments diagonaux positifs telle que

$$A = BB^\top$$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$
- $A_1 = B(B^\top)_1 = b_{11}B_1 \Leftrightarrow B_1 = \frac{1}{b_{11}}A_1$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$
- $A_1 = B(B^\top)_1 = b_{11}B_1 \Leftrightarrow B_1 = \frac{1}{b_{11}}A_1$

- $a_{22} = \underline{B}_2(B^\top)_2 = b_{21}^2 + b_{22}^2$ d'où $b_{22} = \sqrt{a_{22} - b_{21}^2}$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$
- $A_1 = B(B^\top)_1 = b_{11}B_1 \Leftrightarrow B_1 = \frac{1}{b_{11}}A_1$

- $a_{22} = \underline{B}_2(B^\top)_2 = b_{21}^2 + b_{22}^2$ d'où $b_{22} = \sqrt{a_{22} - b_{21}^2}$
- $A_2 = B(B^\top)_2 = b_{21}B_1 + b_{22}B_2 \Leftrightarrow B_2 = \frac{1}{b_{22}}(A_2 - b_{21}B_1)$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$
- $A_1 = B(B^\top)_1 = b_{11}B_1 \Leftrightarrow B_1 = \frac{1}{b_{11}}A_1$

- $a_{22} = \underline{B}_2(B^\top)_2 = b_{21}^2 + b_{22}^2$ d'où $b_{22} = \sqrt{a_{22} - b_{21}^2}$
- $A_2 = B(B^\top)_2 = b_{21}B_1 + b_{22}B_2 \Leftrightarrow B_2 = \frac{1}{b_{22}}(A_2 - b_{21}B_1)$

- $a_{jj} = \underline{B}_j(B^\top)_j = \sum_{k=1}^{j} b_{jk}^2$ d'où $b_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$
- $A_1 = B(B^\top)_1 = b_{11}B_1 \Leftrightarrow B_1 = \frac{1}{b_{11}}A_1$

- $a_{22} = \underline{B}_2(B^\top)_2 = b_{21}^2 + b_{22}^2$ d'où $b_{22} = \sqrt{a_{22} - b_{21}^2}$
- $A_2 = B(B^\top)_2 = b_{21}B_1 + b_{22}B_2 \Leftrightarrow B_2 = \frac{1}{b_{22}}(A_2 - b_{21}B_1)$

- $a_{jj} = \underline{B}_j(B^\top)_j = \sum_{k=1}^{j} b_{jk}^2$ d'où $b_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$
- $A_j = B(B^\top)_j = \sum_{k=1}^{j} b_{jk}B_k \Leftrightarrow B_j = \frac{1}{b_{jj}}(A_j - \sum_{k=1}^{j-1} b_{jk}B_k)$

La factorisation $A = BB^T$

Algorithme

- $a_{11} = \underline{B}_1(B^\top)_1 = b_{11}^2$ d'où $b_{11} = \sqrt{a_{11}}$
- $A_1 = B(B^\top)_1 = b_{11}B_1 \Leftrightarrow B_1 = \frac{1}{b_{11}}A_1$

- $a_{22} = \underline{B}_2(B^\top)_2 = b_{21}^2 + b_{22}^2$ d'où $b_{22} = \sqrt{a_{22} - b_{21}^2}$
- $A_2 = B(B^\top)_2 = b_{21}B_1 + b_{22}B_2 \Leftrightarrow B_2 = \frac{1}{b_{22}}(A_2 - b_{21}B_1)$

- $a_{jj} = \underline{B}_j(B^\top)_j = \sum_{k=1}^{j} b_{jk}^2$ d'où $b_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$
- $A_j = B(B^\top)_j = \sum_{k=1}^{j} b_{jk}B_k \Leftrightarrow B_j = \frac{1}{b_{jj}}(A_j - \sum_{k=1}^{j-1} b_{jk}B_k)$
- $b_{nn} = \sqrt{a_{nn} - \sum_{k=1}^{n-1} b_{nk}^2}$

La factorisation $A = BB^T$

Algorithme

1: $b_{11} \leftarrow \sqrt{a_{11}}$

La factorisation $A = BB^T$

Algorithme

1: $b_{11} \leftarrow \sqrt{a_{11}}$
2: **pour** $i = 2$ jusqu'à n **faire**

4: **fin pour**

La factorisation $A = BB^T$

Algorithme

1: $b_{11} \leftarrow \sqrt{a_{11}}$
2: **pour** $i = 2$ jusqu'à n **faire**
3: $\quad b_{i1} \leftarrow \dfrac{a_{i1}}{b_{11}}$
4: **fin pour**

La factorisation $A = BB^T$

Algorithme

$b_{11} \leftarrow \sqrt{a_{11}}$
pour $i = 2$ jusqu'à n **faire**
    $b_{i1} \leftarrow \dfrac{a_{i1}}{b_{11}}$
fin pour
pour $j = 2$ jusqu'à $n - 1$ **faire**

fin pour

La factorisation $A = BB^T$

Algorithme

$b_{11} \leftarrow \sqrt{a_{11}}$
pour $i = 2$ jusqu'à n **faire**
    $b_{i1} \leftarrow \dfrac{a_{i1}}{b_{11}}$
fin pour
pour $j = 2$ jusqu'à $n - 1$ **faire**
    $b_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$

fin pour

La factorisation $A = BB^T$

Algorithme

$b_{11} \leftarrow \sqrt{a_{11}}$
pour $i = 2$ jusqu'à n **faire**
    $b_{i1} \leftarrow \dfrac{a_{i1}}{b_{11}}$
fin pour
pour $j = 2$ jusqu'à $n - 1$ **faire**
    $b_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$
    pour $i = j + 1$ jusqu'à n **faire**

    fin pour
fin pour

La factorisation $A = BB^T$

Algorithme

$b_{11} \leftarrow \sqrt{a_{11}}$
pour $i = 2$ jusqu'à n **faire**
    $b_{i1} \leftarrow \dfrac{a_{i1}}{b_{11}}$
fin pour
pour $j = 2$ jusqu'à $n - 1$ **faire**
    $b_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$
    pour $i = j + 1$ jusqu'à n **faire**
        $b_{ij} \leftarrow \dfrac{1}{b_{jj}}(a_{ij} - \sum_{k=1}^{j-1} b_{jk} b_{ik})$
    fin pour
fin pour

La factorisation $A = BB^T$

Algorithme

$b_{11} \leftarrow \sqrt{a_{11}}$
pour $i = 2$ jusqu'à n **faire**
    $b_{i1} \leftarrow \dfrac{a_{i1}}{b_{11}}$
fin pour
pour $j = 2$ jusqu'à $n-1$ **faire**
    $b_{jj} \leftarrow \sqrt{a_{jj} - \sum_{k=1}^{j-1} b_{jk}^2}$
    pour $i = j+1$ jusqu'à n **faire**
        $b_{ij} \leftarrow \dfrac{1}{b_{jj}}\left(a_{ij} - \sum_{k=1}^{j-1} b_{jk} b_{ik}\right)$
    fin pour
fin pour
$b_{nn} \leftarrow \sqrt{a_{nn} - \sum_{k=1}^{n-1} b_{nk}^2}$