# Séance 4 : résolution de systèmes linéaires par des méthodes directes : méthode de Gauss

Problème à résoudre : rechercher l'unique solution $x \in \mathbb{R}^n$ du système

$$\boxed{Ax = b}$$

où $A \in GL_n(\mathbb{R})$, $b \in \mathbb{R}^n$.

Deux types de méthodes sont possibles :

↗ méthodes directes : on recherche la solution exacte (nbre fini d'opérations)

↘ méthodes itératives : on construit une suite d'approximations de la solution (nbre d'itérations)

Dans les 2 cas, numériquement, on obtient une solution approchée.

Il existe une méthode directe explicite : la méthode de Cramer :

$$\boxed{x = A^{-1}b = \frac{1}{\det(A)} \, {}^t\mathrm{com}(A).b}$$

où $\mathrm{com}(A)$ est la comatrice de $A$ :

$$\mathrm{com}(A)_{i,j} = (-1)^{i+j} \det \begin{pmatrix} * & A & * \\ * & & * \end{pmatrix}$$

Cette méthode est inutilisable en pratique en raison du nombre d'opérations nécessaires si $n$ grand. Par exemple, si $n = 100$, elle nécessite :

$$\begin{cases} * \ 1 \text{ calcul de déterminant } n \times n \\ * \ n^2 \text{ calculs de déterminant } (n-1)\times(n-1) \end{cases}$$

soit pour $\det(A)$ seulement :

$$\left.\begin{array}{l} n! \text{ additions} \\ (n-1)\,n! \text{ multiplications} \end{array}\right\} \begin{array}{l} N(n+1)! \\ \text{opérations.} \end{array}$$

c'est à dire plus de $10^{100}$ opérations ! Sur un ordinateur effectuant $10^9$ opérations par seconde (1 Gflops), il faudrait plus $10^{11}$ secondes $> 3000$ ans !!

Cette méthode est à rejeter, numériquement.

Il faudra aussi veiller, non seulement au coût des méthodes directes, mais aussi à leur robustesse vis à vis des erreurs d'arrondis dans leur exécution (conditionnement numérique à ne pas confondre avec conditionnement "théorique" vu à la séance 3).

# I) Méthode du pivot de Gauss

Cette méthode est universelle, avec un coût modéré et une robustesse très bonne.

Elle consiste à se ramener à la résolution d'un problème équivalent plus simple, en l'occurrence un système triangulaire :

$$Tx = \beta$$

avec $T = \begin{pmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{pmatrix}$ non nuls

Pour un tel système, la résolution se fait par un principe de remontée :

$$\begin{cases} t_{1,1}x_1 + \cdots & = \beta_1 \\ \quad \ddots & \\ \quad t_{n-1,n-1}x_{n-1} + \cdots & = \beta_{n-1} \\ \quad\quad t_{n,n}x_n & = \beta_n \end{cases}$$

soit $\begin{cases} x_n = \dfrac{\beta_n}{t_{n,n}} \\ \vdots \\ x_1 = \dfrac{1}{t_{1,1}}\left( \cdots \right) \end{cases}$

Cette résolution s'effectue en $\left\{ \begin{array}{ll} n \text{ div.} & \\ \text{de l'ordre de } n^2/2 & \text{add.} \\ \quad \text{"} \quad n^2/2 & \text{mult.} \end{array} \right)$ soit de l'ordre de $n^2$ opérations (résolu en $10^{-5}$ s avec l'ordinateur précédent).

* Le principe de cette transformation en un système triangulaire consiste à faire apparaître des zéros dans chaque colonne sous la diagonale de A.

* On utilise une transformation élémentaire $G : \begin{pmatrix} A \to G(A) \\ b \to G(b) \end{pmatrix}$

telle que $G(A)x = G(b)$ soit un système équivalent et $G(A)$ est une matrice avec une colonne de 0 sous la diagonale :

$$\underbrace{G(A)}_{A_1} = \begin{pmatrix} * & * & \cdots & * \\ 0 & & & \\ \vdots & & A_1' & \\ 0 & & & \end{pmatrix}$$

On appliquera ensuite G à $A_1'$, etc...

* Description de la transformation G :

Etape 1 : échange de lignes.

Si $A = [a_{ij}]$, on note $i_0$ une ligne de A telle que :

$$|a_{i_0,1}| = \underset{1 \leq i \leq n}{\mathrm{Max}} |a_{i,1}| \quad (\neq 0)$$

pivôt

On échange les lignes 1 et $i_0$ de A et de b

On obtient des éléments $\tilde{A}$ et $\tilde{b}$

## Etape 2 : combinaison de lignes

Pour chaque ligne 2 à n de $\tilde{A}$ (et de $\tilde{b}$) on remplace la ligne $\tilde{L}_i$ par la ligne : $\tilde{L}_i \longrightarrow \tilde{L}_i - \dfrac{\tilde{a}_{i,1}}{\tilde{a}_{1,1}} \tilde{L}_1$ $(2 \leq i \leq n)$

On note $G(A)$ et $G(b)$ les nouveaux éléments. Ainsi on a bien

$$\underbrace{G(A)}_{A_1} = \begin{pmatrix} \tilde{a}_{1,1} & \cdots & \cdots \\ 0 & & \\ \vdots & A_1' & \\ 0 & & \end{pmatrix}$$

On montre à présent que les deux systèmes sont bien équivalents à. Pour cela on exhibe une matrice inversible correspondant à une multiplication à gauche par les 2 étapes :

→ Pour l'étape 1, il s'agit de la matrice de permutation :

$$T_{i_0,1} = \begin{array}{c} \\ \\ \\ i_0 \rightarrow \\ \\ \\ \end{array} \begin{pmatrix} 0 & & & 1 & & & \\ & 1 & & \vdots & & 0 & \\ & & \ddots & & & & \\ & & & 1 & & & \\ 1 & \cdots & \cdots & 0 & & & \\ & 0 & & & 1 & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$
$$\qquad\qquad\qquad\quad \uparrow i_0$$

$(\det T_{i_0,1} = -1)$

→ Pour l'étape 2, il s'agit de la matrice de transvection :

$$E(\alpha_2, \dots \alpha_n) = \begin{pmatrix} 1 & & & 0 \\ \alpha_2 & \ddots & & \\ \vdots & & \ddots & \\ \alpha_n & 0 & & 1 \end{pmatrix}$$

avec $\alpha_i = \dfrac{-\tilde{a}_{i,1}}{\tilde{a}_{1,1}}$

$(\det E = 1)$

* On a $G(A) = E_1 \cdot T_1 \cdot A$

donc $G(A)$ est bien inversible.

Or $\det(G(A)) = \tilde{a}_{1,1} \det(A'_1)$,

on en déduit que $A'_1$ est inversible

On peut mettre en place de manière itérative l'algorithme du pivot de Gauss :

Initialisation : on se donne $A$ et $b$

Itération 1 : on transforme $A$ et $b$ en $G(A)$ et $G(b)$

Itération 2 : on transforme $A'_1$ et $b'_1$ et $G(A'_1)$ et $G(b'_1)$ (à bon droit) et on note

$$A_2 = \begin{pmatrix} \tilde{a}_{11} & \cdots & \alpha \\ 0 & & \\ \vdots & G(A'_1) & \\ 0 & & \end{pmatrix}$$

On montre qu'on a

$$A_2 = E_2 . T_2 . E_1 . T_1 . A$$

où

$$E_2 = \begin{pmatrix} 1 & & & & \\ 0 & 1 & & & 0 \\ \vdots & * & \ddots & & \\ \vdots & * & & \ddots & \\ 0 & * & 0 & & 1 \end{pmatrix}$$

(et $b_2 = E_2 . T_2 . E_1 . T_1 . A$)

Itération $(n-1)$ : on aboutit à une matrice :

$$A_{n-1} = \begin{pmatrix} * & & * \\ & \ddots & \\ 0 & & * \end{pmatrix} \quad (\text{et } b_{n-1})$$

telle que

$$\boxed{A_{n-1} = E_{n-1} T_{n-1} \cdots E_2 T_2 A}$$

(et $b_{n-1}$ : idem).

On parle de l'algorithme du pivot de Gauss avec stratégie de pivot partiel.

On compte à présent le nombre d'opérations (hors remontée) :

* $n$ tris (négligeable)
* Par l'itération 1 :

$$(n-1).(\underbrace{n}_{\text{add.}} + \underbrace{n}_{\text{mult}}) + \underbrace{n}_{\text{div}} \sim 2n^2$$

$\vdots$

* Par l'itération $k$ : $\sim 2(n-k)^2$

soit :

$$\sum_{k=1}^{n} 2k^2 = 2\,\frac{n(n+1)(2n+1)}{6}$$

$$\sim \frac{2n^3}{3} \text{ opérations}$$

(la remontée est négligeable)

Par exemple, si $n=100$, sur un ordinateur à 1 Gflops, il faut moins de $10^{-3}$ s pour résoudre le système.

Remarques :

* Le coût en $O(n^3)$ est satisfaisant et ne pourra être amélioré facilement.

* La robustesse numérique de la méthode est assurée par la stratégie de pivôt (voir TD4 avec un exemple où l'absence de cette stratégie conduit à une erreur grossière sur la solution).

* L'algorithme de Gauss sert aussi à calculer des déterminants (coût $\sim 2n^3/3$) :

$$\det(A) = \pm \det(A_{n-1})$$

triangulaire

en fonction des échanges de lignes

ou calculer l'inverse de $A$.

L'algorithme de Gauss sert de base à la factorisation LU d'une matrice :

A faire : TD/TP n° 4

Contenu :

→ l'implémentation de l'algorithme de Gauss

* Implémentation de la remontée + test
* Implémentation de Gauss et test :
  → cas standard ($n = 100$)
  → cas pathologique (Hilbert)
  → sans stratégie de pivot (Ex 1)

(à renvoyer par mail → vendredi 15/10)

Ex 1

$$A = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} \qquad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

et on se met à la place de l'ordinateur travaillant avec précision 3 chiffres significatifs :

$$1{,}0001 \rightsquigarrow 1$$

* Solution exacte :

$$\begin{cases} 10^{-4}x + y = 1 \\ x + y = 2 \end{cases} \iff \begin{cases} 1 = (1 - 10^{-4})x \\ x + y = 2 \end{cases}$$

soit : $\begin{cases} x = \dfrac{1}{0{,}9999} \approx 1 \\ y = 2 - x \approx 1 \end{cases}$

## * Avec stratégie de pivôt (et calcul approché)

$$A = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$A = \begin{pmatrix} 1 & 1 \\ 10^{-4} & 1 \end{pmatrix} \quad b = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 1 \\ 0 & (\hat{\approx})\,1 \end{pmatrix} \quad L_2 \to L_2 - 10^{-4} L_1 \quad \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$\begin{cases} x = 2 - y = 1 \\ y = 1 \end{cases}$$

solution trouvée : $(1, 1)$

## * Sans stratégie de pivôt

$$A = \begin{pmatrix} 10^{-4} & 1 \\ 1 & 1 \end{pmatrix} \quad b = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

$$A = \begin{pmatrix} 10^{-4} & 1 \\ 0 & -10^{4} \end{pmatrix} \left(L_2 \to L_2 - 10^{4} L_1\right) \quad b = \begin{pmatrix} 1 \\ -10^{4} \end{pmatrix}$$

$$\begin{cases} 10^{-4} x + y = 1 \\ y = 1 \end{cases}$$

solution trouvée : $(0, 1)$

erreur grossière !