Plaquette 4 – L'algorithme du pivot : coût, pivot partiel, décomposition LU et Python

Systèmes linéaires — Prépa MPSI / MP2I (1re année)

Exercices

Méthode

Le pivot de Gauss n'est pas qu'une méthode de calcul à la main : c'est **l'algorithme** qu'exécutent tous les logiciels de calcul. On le découpe en deux phases, l'**élimination** (on rend le système triangulaire supérieur) puis la **remontée**. On apprend ici à compter ses opérations (de l'ordre de $\frac{2}{3}n^3$), à choisir le pivot pour limiter les erreurs d'arrondi (**pivot partiel** : le plus grand en valeur absolue), à mémoriser les multiplicateurs sous la forme $A = LU$ pour résoudre plusieurs systèmes à moindre coût, et à programmer tout cela en Python. On y rencontre aussi les limites du calcul flottant : un rang faux, un système mal conditionné.

Exercice 1 – Remontée d'un système triangulaire et son coût

Soit le système triangulaire

$$\begin{cases} 2x - y + 3z + t = -2 \\ y + 2z - t = -1 \\ 3z + 2t = -1 \\ 4t = 4 \end{cases}$$

a) Le résoudre par remontée.

b) Justifier qu'un système triangulaire supérieur de taille n dont les coefficients diagonaux sont tous non nuls admet une unique solution, quel que soit le second membre.

c) Compter exactement le nombre de multiplications, de soustractions et de divisions effectuées par la remontée en taille n.

Exercice 2 – Descente bidiagonale : un compte épargne

Un compte rapporte 2 % par an. On note b_k le versement effectué en fin d'année k ($1 \leq k \leq n$) et x_k le solde juste après ce versement, le compte étant vide au départ. On pose $a = 1{,}02$.

a) Justifier que $x_1 = b_1$ et $x_k - a x_{k-1} = b_k$ pour $k \geq 2$. Écrire ce système sous la forme $LX = B$ en précisant L.

b) Résoudre le système par **descente** et montrer que $x_k = \sum_{j=1}^{k} a^{k-j} b_j$.

c) En déduire la matrice L^{-1}.

d) Application : $n = 3$, versements de 1000, 1500 et 2000 euros. Calculer x_3.

Exercice 3 – Compter les opérations du pivot de Gauss

On résout un système de Cramer $AX = B$ de taille n par la méthode du pivot, sans échange de lignes.

a) À l'étape k ($1 \leq k \leq n-1$), on élimine x_k des lignes $k+1$ à n. Combien de multiplications cela coûte-t-il (second membre compris) ?

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

b) En déduire que l'élimination coûte $\frac{(n-1)n(n+1)}{3}$ multiplications, et un équivalent du nombre total d'opérations quand $n \to +\infty$.

c) Un ordinateur effectue 10^9 opérations par seconde. Estimer la durée de la résolution pour $n = 1000$ puis $n = 10^4$.

d) Que devient la durée si l'on double n ? Comparer avec le coût de la remontée.

Exercice 4 – Pivot nul : permuter les lignes

a) Résoudre par la méthode du pivot

$$\begin{cases} 2y + z = 3 \\ x + y + z = 3 \\ 2x + 4y + 5z = 11 \end{cases}$$

Pourquoi ne peut-on pas commencer par L_1 ?

b) On considère $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 3 \end{pmatrix}$. Effectuer la première étape du pivot. Que se passe-t-il à la deuxième ?

c) Soit P la matrice obtenue en échangeant les lignes 2 et 3 de I_3. Calculer PA, puis trouver L triangulaire inférieure à diagonale de 1 et U triangulaire supérieure telles que $PA = LU$.

d) Résoudre $AX = (3, 4, 6)$ à l'aide de cette décomposition.

Exercice 5 – Petit pivot et erreurs d'arrondi

On considère, avec $\varepsilon = 10^{-4}$, le système

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

a) Donner sa solution exacte et des valeurs approchées à 10^{-4} près.

b) On calcule en arrondissant chaque résultat intermédiaire à **trois chiffres significatifs**. Résoudre en prenant ε comme pivot. Commenter.

c) Reprendre le calcul après avoir échangé les deux lignes (pivot partiel). Commenter.

d) Expliquer pourquoi le pivot partiel garantit $|m_{ik}| \leq 1$ et en quoi cela limite la propagation des erreurs.

Exercice 6 – Décomposition LU et plusieurs seconds membres

Soit $A = \begin{pmatrix} 2 & 1 & 1 \\ 4 & 3 & 3 \\ 8 & 7 & 9 \end{pmatrix}$.

a) Effectuer l'élimination de Gauss sans échange en notant les multiplicateurs. En déduire $A = LU$ avec L triangulaire inférieure à diagonale de 1 et U triangulaire supérieure.

b) Vérifier le produit LU.

c) Résoudre $AX = B_1$ avec $B_1 = (1, 1, 1)$ puis $AX = B_2$ avec $B_2 = (0, 2, 8)$, en utilisant L et U.

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

d) Pour p seconds membres en taille n, comparer le coût de « refaire le pivot à chaque fois » et celui de « factoriser une fois, puis descendre et remonter ».

Exercice 7 – Unicité de la décomposition LU

Soit $A \in GL_n(\mathrm{K})$. On suppose que $A = L_1U_1 = L_2U_2$ avec L_1, L_2 triangulaires inférieures à diagonale de 1, et U_1, U_2 triangulaires supérieures.

a) Justifier que U_1 et U_2 sont inversibles.

b) Montrer que $L_2^{-1}L_1 = U_2U_1^{-1}$.

c) On admet que l'inverse et le produit de matrices triangulaires inférieures à diagonale de 1 sont de même type, et de même pour les triangulaires supérieures inversibles. Conclure que $L_1 = L_2$ et $U_1 = U_2$.

d) L'unicité subsiste-t-elle si l'on n'impose pas la diagonale de 1 à L ? Donner un exemple en taille 1 puis en taille 2.

Exercice 8 – Quand la décomposition LU existe-t-elle ?

Pour $A \in M_n(\mathrm{K})$ et $1 \leq k \leq n$, on note A_k la sous-matrice formée des k premières lignes et k premières colonnes de A (sous-matrice principale).

a) Montrer que $A = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ n'admet pas de décomposition $A = LU$ avec L triangulaire inférieure à diagonale de 1 et U triangulaire supérieure.

b) On suppose $A = LU$ avec U inversible. Montrer, par un calcul par blocs, que $A_k = L_kU_k$ pour tout k, et en déduire que toutes les A_k sont inversibles.

c) Réciproquement, on admet que si $A_1, \ldots, A_k$ sont inversibles, les k premiers pivots de l'élimination sans échange sont non nuls. Conclure.

d) Application : $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 1 & 3 & 4 \end{pmatrix}$ admet-elle une décomposition LU sans échange ?

Exercice 9 – Système tridiagonal : équilibre d'un fil tendu

La discrétisation de $-u'' = 1$ sur $[0,1]$, avec $u(0) = u(1) = 0$, conduit (à un facteur près) au système de taille n

$$2x_1 - x_2 = 1, \qquad -x_{k-1} + 2x_k - x_{k+1} = 1 \ (2 \leq k \leq n-1), \qquad -x_{n-1} + 2x_n = 1$$

a) On effectue l'élimination sans échange. Montrer que le k-ième pivot vaut $d_k = \dfrac{k+1}{k}$ et que le second membre modifié vaut $b'_k = \dfrac{k+1}{2}$.

b) Pour $n = 4$, terminer la résolution.

c) Vérifier que $x_k = \dfrac{k(n+1-k)}{2}$ est solution pour tout n (avec la convention $x_0 = x_{n+1} = 0$).

d) Combien d'opérations l'élimination demande-t-elle ici ? Comparer avec $\frac{2}{3}n^3$.

Exercice 10 – Programmer la remontée en Python

Une matrice est représentée par la liste de ses lignes.

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

a) Compléter la fonction suivante, qui résout $UX = B$ pour U triangulaire supérieure à diagonale non nulle :

```
def remontee(U, B):
    n = len(B)
    X = [0] * n
    for i in range(n - 1, -1, -1):
        s = B[i]
        for j in range(...):
            s = s - U[i][j] * X[j]
        X[i] = ...
    return X
```

b) Qu'affiche l'appel ci-dessous ?

```
U = [[2, -1, 3, 1], [0, 1, 2, -1], [0, 0, 3, 2], [0, 0, 0, 4]]
print(remontee(U, [-2, -1, -1, 4]))
```

c) Que se passe-t-il si un coefficient diagonal est nul ? Proposer une protection.

d) Écrire de même une fonction *descente(L, B)* pour L triangulaire inférieure.

Exercice 11 – Un bug classique dans le pivot partiel

On dispose de la fonction *remontee* de l'exercice précédent. Un élève programme le pivot de Gauss avec pivot partiel :

```
def gauss(A, B):
    n = len(B)
    A = [ligne[:] for ligne in A]
    B = B[:]
    for k in range(n):
        p = max(range(k, n), key=lambda i: abs(A[i][k]))
        A[k], A[p] = A[p], A[k]
        for i in range(k + 1, n):
            m = A[i][k] / A[k][k]
            for j in range(k, n):
                A[i][j] = A[i][j] - m * A[k][j]
            B[i] = B[i] - m * B[k]
    return remontee(A, B)
```

a) Expliquer le rôle des lignes 3, 4 et 6.

b) Résoudre à la main le système $x + y + z = 6$, $2x + y + 3z = 13$, $4x + 2y + z = 11$.

c) L'appel *gauss([[1, 1, 1], [2, 1, 3], [4, 2, 1]], [6, 13, 11])* affiche environ *[-6.6, 14.3, 3.8]*. Vérifier que c'est faux et trouver l'erreur.

d) Corriger, puis dérouler l'algorithme corrigé sur ce système.

Exercice 12 – Calculer un rang en Python : flottants ou fractions

On considère la fonction suivante (une matrice est la liste de ses lignes) :

```
def rang(M):
    M = [ligne[:] for ligne in M]
    p, n = len(M), len(M[0])
    r = 0
    for j in range(n):
        piv = None
        for i in range(r, p):
```

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

```
            if M[i][j] != 0:
                piv = i
                break
        if piv is None:
            continue
        M[r], M[piv] = M[piv], M[r]
        for i in range(r + 1, p):
            m = M[i][j] / M[r][j]
            for k in range(j, n):
                M[i][k] = M[i][k] - m * M[r][k]
        r = r + 1
    return r
```

a) Expliquer pourquoi cette fonction renvoie le rang de M.

b) Que renvoie *rang([[1, 2, 3], [4, 5, 6], [7, 8, 9]])* ? Le justifier à la main.

c) *rang([[0.1, 0.3], [0.3, 0.9]])* renvoie *2*. Quel est le vrai rang ? Expliquer, sachant qu'en flottants *0.3 / 0.1* vaut *2.9999999999999996*.

d) Proposer deux remèdes.

Exercice 13 – Une méthode itérative : Jacobi

On veut résoudre $4x + y = 9$, $x + 3y = 5$ sans pivot. On part de $(x_0, y_0) = (0, 0)$ et l'on pose

$$x_{k+1} = \frac{9 - y_k}{4}, \qquad y_{k+1} = \frac{5 - x_k}{3}$$

a) Résoudre le système exactement.

b) Calculer les itérés jusqu'à (x_4, y_4) (valeurs à 10^{-4} près).

c) En notant $e_k = \max(|x_k - 2|, |y_k - 1|)$, montrer que $e_{k+1} \le \frac{1}{3} e_k$, puis $e_k \le 2 \times 3^{-k}$. À partir de quel rang est-on sûr d'avoir $e_k < 10^{-3}$?

d) Quelle propriété des coefficients a été utilisée ? Pourquoi une telle méthode est-elle intéressante pour de très grands systèmes creux ?

Exercice 14 – Un système mal conditionné

On considère les deux systèmes

$$(S)\ \begin{cases} x + y = 2 \\ x + 1{,}0001\, y = 2{,}0001 \end{cases} \qquad (S')\ \begin{cases} x + y = 2 \\ x + 1{,}0001\, y = 2{,}0002 \end{cases}$$

a) Résoudre (S) et (S').

b) Comparer la variation relative du second membre et celle de la solution.

c) Interpréter géométriquement.

d) Un capteur mesure le second membre avec une incertitude de 10^{-4}. Quelle confiance accorder à la solution calculée ? Que faut-il changer ?

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).

Exercice 15 – Synthèse : la matrice de Hilbert

Soit $H = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}$, de coefficients $h_{ij} = \frac{1}{i+j-1}$.

a) Calculer $B = H\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

b) Vérifier que $H^{-1} = \begin{pmatrix} 9 & -36 & 30 \\ -36 & 192 & -180 \\ 30 & -180 & 180 \end{pmatrix}$ en calculant la première colonne de HH^{-1} (on admettra les deux autres).

c) On perturbe la troisième composante de B de 0,01 : $B' = B + (0, 0, 0{,}01)$. Calculer la solution X' de $HX = B'$ sans refaire de pivot.

d) Comparer les erreurs relatives, puis expliquer pourquoi la matrice de Hilbert, qui apparaît en approchant une fonction par un polynôme au sens des moindres carrés, est redoutée en calcul numérique.

mathlvl.fr — Tous droits réservés. Toute reproduction, diffusion, revente ou exploitation commerciale, même partielle, est strictement interdite.
Tout contrevenant s'expose à des poursuites en vertu du Code de la propriété intellectuelle (art. L.335-2 et suivants).