

Calcul scientifique

TP N°4 : Résolution numérique de systèmes linéaires

Exercice 1 *Décomposition LU d'une matrice*

1. Importer le module `numpy`
2. Programmer une fonction `LU` qui renvoie la décomposition d'une LU d'une matrice A :
 - (a) Récupérer les dimensions de la matrice A ;
 - (b) Initialiser la matrice L à la matrice identité ;
 - (c) Initialiser la matrice U à la matrice A ;
 - (d) Implémenter les boucles permettant de calculer de façon itérative les éléments des matrices L et U .
3. Calculer la décomposition LU de la matrice :

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

4. Afficher les matrices A , L et U et vérifier votre résultat.
5. Programmer une fonction `solve_LU` qui permet de résoudre le système $LUx = b$:
 - (a) Écrire une fonction `descente` qui prend en argument une matrice triangulaire inférieure L et un vecteur b et qui renvoie le vecteur y solution du système $Ly = b$, calculé par algorithme de descente.
 - (b) Tester votre fonction `descente` avec la matrice L calculée précédemment et le vecteur $b_1 = \begin{pmatrix} 1.5 \\ 2 \\ 2.5 \end{pmatrix}$. Vérifier le résultat obtenu.
 - (c) Écrire une fonction `remontee` qui prend en argument une matrice triangulaire supérieure U et un vecteur y et qui renvoie le vecteur x solution du système $Ux = y$, calculé par algorithme de remontée.
 - (d) Définir la fonction `solve_LU` qui prend en argument les matrices L et U et un vecteur b et qui renvoie le vecteur x solution du système $LUx = b$.
6. Tester votre fonction `solve_LU` avec la matrice M et le vecteur b_1 définis précédemment. Vérifier que le résultat est correct.

Exercice 2 *Résolution numérique d'une EDO*

Soit f une fonction définie sur l'intervalle $[0, 1]$ et à valeurs dans $\mathbb{R}$. On considère l'équation différentielle ordinaire (EDO) suivante :

$$\begin{cases} -u''(x) + u(x) = f(x) & \forall x \in [0, 1] \\ u(0) = u(1) = 0 \end{cases} \quad (1)$$

On admet que f est suffisamment régulière pour que l'équation (1) admette une unique solution u . Nous allons utiliser la méthode des différences finies pour résoudre numériquement cette équation. Soit N un entier strictement positif. On discrétise l'intervalle $[0, 1]$ en N points équidistants $x_0, \dots, x_{N-1}$.

$$\begin{array}{ccccccc} 0 & & & & & & 1 \\ | & | & | & | & | & | & | \\ x_0 & x_1 & x_2 & \cdots & \cdots & x_{N-2} & x_{N-1} \end{array}$$

On va chercher à calculer la valeur de la solution u aux points x_i , $i = 0, \dots, N-1$.

En posant $h = \frac{1}{N-1}$, on a $x_i = ih$ pour $i = 0, \dots, N-1$. On admet qu'on peut approcher la dérivée seconde de u en x_i par la formule :

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1}))}{h^2} \quad (2)$$

En posant les vecteurs $\mathbf{u} = [u(x_0), \dots, u(x_{N-1})]^T$ et $\mathbf{f} = [0, f(x_1), \dots, f(x_{N-2}), 0]^T$, on peut écrire l'équation (1) sous la forme matricielle :

$$\frac{1}{h^2} A \mathbf{u} + \mathbf{u} = \mathbf{f} \quad (3)$$

où A est la matrice tridiagonale de taille N telle que :

$$A = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 2 & -1 & \cdots & 0 & 0 \\ 0 & -1 & 2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 2 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 \end{bmatrix} \quad (4)$$

Cette équation (3) peut se réécrire sous la forme :

$$M \mathbf{u} = \mathbf{f} \quad (5)$$

où $M = \frac{1}{h^2} A + I$ est une matrice carrée de taille N , I étant la matrice identité. Pour résoudre ce système, on va utiliser la factorisation LU de M , ainsi que l'algorithme de descente-remontée de l'exercice précédent.

1. Implémenter une fonction `matM` qui prend en argument `N` et qui renvoie la matrice M telle que définie précédemment.
2. Implémenter une fonction `vecF` qui prend en argument `N` ainsi qu'une fonction `f` et qui renvoie le vecteur $\mathbf{f}$ tel que défini ci-dessus.
3. Définir un vecteur `xplot` équidistribué sur $[0, 1]$ et de taille 1000, avec la fonction `np.linspace`.

On peut vérifier par un calcul (mais on ne demande pas de le faire) que la fonction $u_{\text{exact}}(x) = \sin(\pi x)^2$ est solution de l'équation (1) pour le second membre $f_1 : x \mapsto \sin^2(\pi x) - 2\pi^2 (1 - 2\sin^2(\pi x))$.

4. Définir un vecteur `u_exact` tel que `u_exact[i] = u_exact(xplot[i])`.
5. À l'aide des fonctions `LU` définir un vecteur `u_approx`, solution de l'équation matricielle (5) pour $N = 1000$.
6. Sur une même figure, tracer les courbes de u_{exact} et de u_{approx} .
7. On considère maintenant le second membre $f_2 : x \mapsto x^2 \cos(x) - x \sin(x)$. Sans recalculer la décomposition de M , calculer une solution approchée de l'équation (1) avec cette fonction f , puis tracer sa courbe.
8. (Bonus) On souhaite étudier la convergence de la solution numérique u_{approx} vers la solution exacte u_{exact} dans l'équation avec second membre f_1 . Calculer `u_approx` et estimer l'erreur correspondante $\|u_{\text{approx}} - u_{\text{exact}}\|_2$ pour $N = 10, 100, 1000, 2000$. Tracer l'évolution de cette erreur en fonction de $h = \frac{1}{N-1}$ en échelle $\log - \log$.