

---

## Méthodes directes de résolution de systèmes linéaires

---

On rappelle qu'une matrice carrée  $L = (l_{ij})_{1 \leq i, j \leq n}$  est dite triangulaire inférieure si tous les  $l_{ij}$  avec  $j > i$  sont nuls. Une combinaison linéaire de matrices triangulaires inférieures est une matrice triangulaire inférieure. Le produit de deux matrices triangulaires inférieures est également une matrice triangulaire inférieure. En termes savants, l'ensemble des matrices triangulaires inférieures est une sous-algèbre de  $M_n(\mathbb{R})$  (ou  $M_n(\mathbb{C})$ ). De même, l'ensemble des matrices triangulaires inférieures inversibles est un sous-groupe de  $GL_n(\mathbb{R})$ .

Les termes a priori non nuls de  $L$  sont dans le triangle inférieur, d'où la notation  $L$  pour "Lower" dans la décomposition  $LU$ . On a la même définition et les mêmes considérations pour *triangulaire supérieure*, notée  $U$  pour "upper".

On a le théorème de décomposition suivant, basé sur la méthode de Gauss :

**Théorème 1 (Factorisation LU d'une matrice)** Soit  $A = (a_{ij})_{1 \leq i, j \leq n}$  une matrice carrée d'ordre  $n$  telle que les  $n$  sous-matrices de taille  $k$ ,

$$\Delta_k = \begin{pmatrix} a_{11} & \dots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \dots & a_{kk} \end{pmatrix} \quad k = 1, \dots, n,$$

soient inversibles. Alors il existe une matrice triangulaire inférieure  $L = (l_{ij})_{1 \leq i, j \leq n}$  avec des 1 sur la diagonale et une matrice triangulaire supérieure  $U$  telles que  $A = LU$ . De plus, une telle factorisation est unique.

Remarquons que si  $A$  est inversible sans l'hypothèse sur les sous-matrices, on peut être obligée pour la méthode de Gauss d'effectuer des permutations sur les lignes lors de la recherche du pivot. Dans ce cas on peut obtenir une décomposition  $PA = LU$  où  $P$  est une matrice de permutation. Mais une telle décomposition n'est plus unique.

On peut montrer que pour une matrice pleine, le coût de la décomposition  $LU$  est en  $O(n^3)$  où  $n$  est la taille de la matrice.

### Décomposition LU des matrices-bande

**Définition 1** Une matrice  $A = (a_{ij})_{1 \leq i, j \leq n}$  est une matrice-bande de largeur de bande  $l$  si

$$\forall i, j \quad |j - i| > l \Rightarrow a_{ij} = 0.$$

En d'autres termes, seuls les coefficients des diagonales  $k$  avec  $|k| \leq l$  sont (a priori) non nuls. Géométriquement,  $l$  représente plutôt la demi-largeur de bande. Par exemple, une matrice tridiagonale est une matrice de largeur de bande 1.

Un point important en pratique, pour réduire le nombre de calculs : la factorisation  $LU$  garde la structure par bandes : si  $A$  est une matrice de largeur de bande  $l$ , alors les matrices  $L$  et  $U$  ont la même largeur de bande.

**Décomposition de Cholesky** Si une matrice  $A$  est symétrique définie positive, elle vérifie les conditions d'application du théorème précédent car les matrices  $\Delta_k$  sont symétriques définies positives. Une telle matrice admet donc une décomposition  $LU$ . Mais on peut faire un peu mieux grâce à la symétrie, et l'écrire sous la forme  $A = B \cdot B^t$  avec  $B$  triangulaire supérieure : c'est la *factorisation de Cholesky* : algorithmique en  $O(n^3)$ , et la structure-bande est également conservée.

## Méthodes itératives de résolution de systèmes linéaires

Pour résoudre un système linéaire  $Ax = b$  où  $A$  est une matrice carrée de taille  $n$ , les méthodes directes comme la méthode de Gauss ne sont pas toujours applicables (matrice pleine de grande taille, le nombre d'opérations est  $\sim 2/3n^3$ ) ou souhaitable (une valeur approchée de la solution suffit, par exemple). Dans ce cas, on a recours à des méthodes de point fixe, également appelées méthodes itératives.

On met le système sous une forme équivalente  $x = Bx + c$ , (cf. ci-après pour des exemples) et à partir d'un élément  $x_0$ , on construit la suite

$$x^{k+1} = Bx^k + c,$$

où  $B$  est une matrice carrée de taille  $n$ ,  $c$  un vecteur de  $\mathbb{R}^n$  et  $x \in \mathbb{R}^n$  (ou  $\mathbb{C}^n$ ). Si la suite converge, sa limite vérifie

$$x = Bx + c,$$

de sorte que par différence, l'erreur  $e^k = x^k - x$  vérifie  $e^{k+1} = Be^k$ , et par récurrence  $e^k = B^k e^0$ . On en déduit

**Proposition 1** *La suite  $(x^k)$  définie par  $x^{k+1} = Bx^k + c$  converge pour tout choix de l'élément initial  $x^0$  si et seulement si la suite de matrices  $(B^k)$  tend vers 0.*

Les deux questions principales qui se posent sont :

1. Etant donnée une méthode itérative de matrice  $B$ , déterminer si la méthode est convergente, i.e. si la suite  $(x^k)$  définie par  $x^{k+1} = Bx^k + c$  converge pour tout choix de l'élément initial  $x^0$  ;
2. Etant données deux méthodes itératives convergentes, comparer leur vitesse de convergence.

Le théorème suivant donne une réponse satisfaisante à ces questions, en terme de rayon spectral de  $B$ .

**Théorème 2 (CNS de convergence)** *Soit  $B \in M_n(\mathbb{C})$ . La suite  $(B^k)$  des puissances de  $B$  tend vers 0 si et seulement si  $\rho(B) < 1$ .*

*Dans ce cas, la convergence est d'autant plus rapide que  $\rho(B)$  est petite, dans le sens où (pour toute norme matricielle subordonnée)*

$$\lim_{k \rightarrow \infty} \|B^k\|^{1/k} = \rho(B). \tag{1}$$

Moralement, (1) dit que  $\|B^k\|$  se comporte comme  $\rho(B)^k$  (suite géométrique).

**Exemple 1 : la méthode de Jacobi**

Pour résoudre  $Ax = b$ , on écrit  $A = D + F$ , où  $D$  est la diagonale de  $A$ , supposée inversible, et  $F = A - D$  composée des parties triangulaire inférieure et triangulaire supérieure de  $A$ . On a

$$Ax = b \iff Dx = -Fx + b \iff x = -D^{-1}Fx + D^{-1}b,$$

de sorte que la méthode converge si et seulement si  $\rho(D^{-1}F) < 1$ . On peut alors montrer (admis) que si  $A$  est symétrique définie positive et tridiagonale, cette condition est vérifiée.

**Exemple 2 : méthode du gradient à pas fixe**

Si  $A$  est une matrice symétrique définie positive, l'algorithme de gradient à pas fixe pour la résolution de  $Ax = b$  s'écrit

$$x^{k+1} = x^k - \mu(Ax^k - b),$$

où  $\mu \in \mathbb{R}$  est un paramètre à fixer. D'après ce qui précède, il converge pour toute donnée initiale  $x^0$  ssi  $\rho(I - \mu A) < 1$ , i.e.  $0 < \mu < 2/\lambda_i(A)$  pour toute valeur propre  $\lambda_i(A)$  de  $A$ . L'algorithme converge alors vers la solution de  $Ax = b$ , qui est également l'unique minimiseur de  $J(x) = \frac{1}{2}\langle x, Ax \rangle - \langle b, x \rangle$  sur  $\mathbb{R}^n$  (exercice). On a un algorithme de minimisation.

## Calcul numérique de valeurs propres

Un algorithme simple pour le calcul de valeurs propres de  $A$  est la **méthode de la puissance**

$$y_{k+1} = Ay_k \quad k = 0, 1, 2, \dots, \tag{2}$$

où  $y_0 \in \mathbb{R}^N$  est arbitraire. Le théorème suivant montre que  $y_k = A^k y_0$  "tend" vers un vecteur propre de  $A$  et que le quotient de Rayleigh  $y_k^* A y_k / y_k^* y_k$  tend vers la valeur propre de  $A$  la plus grande en valeur absolue.

On rappelle que la notation  $z^*$  pour un vecteur ou une matrice désigne la transposée du conjugué.

**Théorème 3** Soit  $A$  une matrice diagonalisable de valeurs propres  $\lambda_1, \lambda_2, \dots, \lambda_n$  et de vecteurs propres  $v_1, \dots, v_n$  normalisés par  $\|v_i\|_2 = 1$ . Si

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|,$$

les vecteurs  $y_k$  donnés par (2) vérifient

$$y_k = \lambda_1^k (a_1 v_1 + O(|\lambda_2/\lambda_1|^k)),$$

où  $a_1$  est défini par  $y_0 = \sum_i a_i v_i$ . Si  $a_1 \neq 0$ , le quotient de Rayleigh satisfait

$$\frac{y_k^* A y_k}{y_k^* y_k} = \lambda_1 + O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right). \tag{3}$$

Si  $A$  est une matrice normale, l'erreur dans (3) est  $O(|\lambda_2/\lambda_1|^{2k})$ .

On rappelle que  $A$  est une matrice normale si  $A^*$  commute avec  $A$ , ou de manière équivalente, si  $A$  est diagonalisable dans une base orthonormale. *Remarques :*

1. Les éléments de  $y_k$  croissent exponentiellement avec  $k$ , donc il est recommandé de normaliser  $y_k$  après chaque itération, i.e. de remplacer  $y_k$  par  $y_k/\|y_k\|$ .
2. Si  $|\lambda_1| = |\lambda_2|$ , cette méthode ne s'applique plus en général.
3. Si  $|\lambda_2/\lambda_1|$  est proche de 1, la convergence est lente. Pour accélérer la convergence, on peut utiliser la méthode de la puissance inverse.

### Méthodes LR et QR

On rappelle que si  $A$  a toutes ses sous-matrices principales régulières, il existe une unique décomposition  $A = LR$  où  $L$  est triangulaire inférieure et  $R$  triangulaire supérieure avec des 1 sur la diagonale. (NB : dans le contexte du calcul de valeurs propres, la décomposition  $LU$  se note  $LR$  pour Left et Right).

L'algorithme LR pour le calcul des valeurs propres est : soit  $A_0 = A$ . Pour  $k = 0, 1, 2, \dots$ , faire

$$\begin{aligned} L_k R_k &= A_k \quad (\text{décomposition LR}), \\ A_{k+1} &= R_k L_k. \end{aligned}$$

Comme  $R_k A_k R_k^{-1} = A_{k+1}$ , les matrices  $A_k$  et  $A_{k+1}$  sont semblables, et donc  $A$  et  $A_k$  sont semblables. Cet algorithme converge vers une matrice triangulaire supérieure, sous certaines hypothèses sur  $A$ .

L'algorithme QR est similaire : on remplace la décomposition LR par la décomposition QR : si  $A$  est régulière, il existe une décomposition  $A = QR$  où  $Q$  est orthogonale et  $R$  est triangulaire supérieure.

## Exercices de programmation

1. Examinez la décomposition  $LU$  (`lu`) et de Cholesky (`chol`) des matrices suivantes :

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}, \quad C = \begin{pmatrix} 4 & 1 \\ 1 & 4 \end{pmatrix}.$$

2. Faire un programme `tp2exo2.sce` qui :  
- définit la matrice de taille  $N$  suivante :

$$A_N = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{pmatrix};$$

- calcule sa décomposition  $LU$  et indique le temps de calcul nécessaire (`tic`, `toc`) ;
- calcule sa décomposition de Cholesky et indique le temps de calcul nécessaire.

Observez la structure-bande des matrices obtenues.

3. Dans le programme précédent, quelle taille maximale de matrice est-elle possible ?

Pour quelle raison ? Pour  $N$  grand, quelle est la décomposition la plus rapide,  $LU$  ou Cholesky ? Pourquoi ?

4. Programmer en Scilab l'algorithme de Jacobi sous la forme d'une fonction `function x=Jacob(A,b, x0,kmax)`, où  $A$  est une matrice,  $b$  un vecteur,  $x_0$  la donnée initiale et  $kmax$  le nombre d'itérations. Le tester sur deux ou trois exemples. Observer la convergence et trouver également un cas pour lequel l'algorithme de Jacobi ne converge pas.

5. Pour l'algorithme de Jacobi, faire un programme `tp2exo5.sce` qui trace l'erreur  $\|x^k - x\|$  en fonction de  $k$ , pour un cas-test dont vous connaissez la solution  $x$ ;  $x^k$  est le  $k$ ème itéré de l'algorithme de Jacobi. Mettez en évidence la vitesse de convergence (échelle logarithmiques).

6. Reprenez la question 4 avec l'algorithme du gradient à pas fixe.

7. Reprenez la question 5 avec l'algorithme du gradient à pas fixe. Tracez cette fois sur un même graphique l'erreur  $\|x^k - x\|$  en fonction de  $k$ , pour trois pas différents (de manière à mettre en évidence l'influence du choix du pas sur la vitesse de convergence).

8. Examiner en Maple ou Scilab les itérés de la matrice

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

pour la méthode QR et la méthode LR. Les valeurs propres de  $A$  sont :

$$\lambda_i = 2 - 2 \cos\left(\frac{i\pi}{4}\right) \quad i = 1, 2, 3, \text{ i.e.}$$

$$0.5857864376, \quad 2, \quad 3.4142135624.$$

Que se passe-t-il si on remplace  $A$  par une matrice orthogonale ?

9. Tester sur la matrice précédente ou un autre cas-test la méthode de la puissance. A l'aide d'un graphique de l'erreur, mettre en évidence la vitesse de convergence démontrée dans le Théorème 3.

---

### Rapport de TP : date limite ve 12/12

Faites les programmes des questions 2, 4 et 7. Commentez dans votre rapport les résultats de ces programmes, et joignez le graphique obtenu dans la question 7. Répondez également à la question 3.

Concernant les programmes des questions 2, 4 et 7, envoyez-les moi par email en document attaché, à l'adresse

`pierre@math.univ-poitiers.fr`

Il n'est pas utile de joindre vos programmes à votre rapport.