

M54: Analyse numérique matricielle (cours)

Bernhard Beckermann

Laboratoire Painlevé, Université de Lille, France
Bernhard.Beckermann@univ-lille.fr

09/09/2025

Motivations (et bref contenu) de ce cours

L'analyse numérique matricielle sert pour...

- une simulation en aérodynamique (d'un avion, d'une voiture, ...)
- la Conception Assistée par Ordinateur d'une pièce mécanique (un moteur, une prothèse, ...)
- les techniques de compression de données (photos, big data,...)
- le fonctionnement d'un moteur de recherche (Google, ...)

Ces thèmes sont évoqués plus en détail dans d'autres modules de maths applis, mais comme brique de base on doit

- trouver un/plusieurs/tous les éléments propres d'une matrice A de grande taille
- trouver $x \in \mathbb{R}^n$ solution de $Ax = b$ dans un sens à préciser (les moindres carrés).

Motivations (et bref contenu) de ce cours

L'analyse numérique matricielle sert pour...

- une simulation en aérodynamique (d'un avion, d'une voiture, ...)
- la Conception Assistée par Ordinateur d'une pièce mécanique (un moteur, une prothèse, ...)
- les techniques de compression de données (photos, big data,...)
- le fonctionnement d'un moteur de recherche (Google, ...)

Ces thèmes sont évoqués plus en détail dans d'autres modules de maths applis, mais comme brique de base on doit

- trouver un/plusieurs/tous les éléments propres d'une matrice A de grande taille
- trouver $x \in \mathbb{R}^n$ solution de $Ax = b$ dans un sens à préciser (les moindres carrés).

Nos défis :

- montrer l'existence, l'unicité des objets en question (outil : normes matricielles)
- trouver des méthodes efficaces de résolution, en passant par des factorisations (SVD, LU, QR, etc) de A , en tenant compte d'une forme particulière de A
- comprendre l'impact des erreurs sur les données A, b
- comprendre l'effet de la précision finie sur ordinateur...

Table de matières

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algo de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Bibliographie (Lilliad)

- 1 Allaire, G. et Kaber, S.M. Algèbre linéaire numérique. Ellipses, 2002.
- 2 Amodei, L. et Dedieu, J.P. Analyse numérique matricielle. Dunod, 2008.
- 3 Brezinski, C. et Redivo-Zaglia, M. Méthodes numériques directes de l'algèbre matricielle. Ellipses, 2004.
- 4 Ciarlet, P.G. Introduction à l'analyse numérique matricielle et à l'optimisation. Masson, 1988.
- 5 Filbet, F. Analyse numérique. Algorithmique et étude mathématique. Dunod, 2009.
- 6 Lascaux, P. et Théodor, R. Analyse numérique matricielle appliquée à l'art de l'ingénieur. Seconde édition, Tome 1, Masson, 1993.
- 7 Lascaux, P. et Théodor, R. Analyse numérique matricielle appliquée à l'art de l'ingénieur. Seconde édition, Tome 2, Masson, 1993.
- 8 Quarteroni, A. et Saleri, F. Calcul Scientifique. Cours, exercices corrigés et illustrations en MATLAB et Octave. Springer, 2006.
- 9 Schatzman, M. Analyse numérique : Une approche mathématique. Dunod, 2004.

La précision finie sur ordinateur

Un nombre machine est un nombre en virgule flottante (en bases 2, 16 ou ici 10) avec une mantisse et un exposant de taille limitée. Étant donné un nombre α réel, l'ordinateur stocke alors le nombre machine $\text{float}(\alpha)$, avec

$$|\alpha - \text{float}(\alpha)| \leq \epsilon |\alpha|$$

avec la précision machine $\epsilon \approx 10^{-8}$ (resp. 10^{-16}) en simple précision (resp. double précision).

Un bon modèle pour comprendre l'erreur sur ordinateur d'une opération élémentaire entre deux nombres machines α, β est

$$\alpha \otimes \beta = \text{float}(\alpha \times \beta), \quad \times \in \{+, -, \cdot, /\}.$$

Il reste à savoir comment ces "petites" erreurs s'accumulent ?

La précision finie sur ordinateur

Un nombre machine est un nombre en virgule flottante (en bases 2, 16 ou ici 10) avec une mantisse et un exposant de taille limitée. Étant donné un nombre α réel, l'ordinateur stocke alors le nombre machine $\text{float}(\alpha)$, avec

$$|\alpha - \text{float}(\alpha)| \leq \epsilon |\alpha|$$

avec la précision machine $\epsilon \approx 10^{-8}$ (resp. 10^{-16}) en simple précision (resp. double précision).

Un bon modèle pour comprendre l'erreur sur ordinateur d'une opération élémentaire entre deux nombres machines α, β est

$$\alpha \otimes \beta = \text{float}(\alpha \times \beta), \quad \times \in \{+, -, \cdot, /\}.$$

Il reste à savoir comment ces "petites" erreurs s'accumulent ?

Exemple 0.1

Calculer $\beta = (2 + \alpha) - \alpha$ sur ordinateur en double précision pour $\alpha = 10^{20}$ donne $\tilde{\beta} = 0$, avec une erreur relative $|\tilde{\beta} - \beta|/|\beta| = 1$ (de 100%).

On vient de rencontrer un exemple de **cancellation** : l'erreur relative augmente fortement si on forme la différence de deux nombres de taille comparable et de même signe. La cancellation n'apparaît pas pour d'autres opérations arithmétiques élémentaires.

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algorithme de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Notions de base : espace euclidien \mathbb{K}^n

- $\mathbb{K} \in \{\mathbb{R}, \mathbb{C}\}$ corps des nombres réels ou complexes. Un élément de \mathbb{K} est dit scalaire, noté par des lettres grecs ;
- Un \mathbb{K} -espace euclidien E de dimension finie n est un \mathbb{K} -espace vectoriel muni d'un produit scalaire $\langle \cdot, \cdot \rangle$.
- **Dans ce cours** (presque) toujours : $E = \mathbb{K}^n$ espace de vecteurs = **vecteurs colonnes** x notés en minuscules, avec élément x_j à la position $j \in \{1, \dots, n\}$.
- $E = \mathbb{K}^n$ est muni de la **base canonique** $\{e_1, \dots, e_n\}$:
 $\forall x \in \mathbb{K}^n : x = x_1 e_1 + \dots + x_n e_n$, avec $e_j \in \mathbb{K}^n$ comportant un $1 \in \mathbb{K}$ à la position j , et sinon que des $0 \in \mathbb{K}$.
- $E = \mathbb{K}^n$ est muni du **produit scalaire**

$$\forall x, y \in \mathbb{K}^n : \langle x, y \rangle = \sum_{j=1}^n \overline{x_j} y_j = \overline{\langle y, x \rangle},$$

avec $\langle x, \alpha y + \beta z \rangle = \alpha \langle x, y \rangle + \beta \langle x, z \rangle$, et $\langle \alpha y + \beta z, x \rangle = \overline{\alpha} \langle y, x \rangle + \overline{\beta} \langle z, x \rangle$ (attention : forme bilinéaire seulement pour $\mathbb{K} = \mathbb{R}$) ;

- **Norme euclidienne** induite $\|x\|_2 = \sqrt{x^* x}$.
- **Inégalité de Cauchy-Schwarz** : $|x^* y| \leq \|x\|_2 \|y\|_2$.
 $x \in \mathbb{K}^n$ est dit **orthogonal** à $y \in \mathbb{K}^n$ si $x^* y = 0$, écriture $x \perp y$;
- Pour $K \subset \mathbb{K}^n$ sous-espace vectoriel :
complément orthogonal $K^\perp = \{y \in \mathbb{K}^n : \forall x \in K, x^* y = 0\}$, $(K^\perp)^\perp = K$.

Notions de base : matrices $\mathcal{M}_{m,n}(\mathbb{K})$

- $\mathcal{M}_{m,n}(\mathbb{K})$ l'ensemble des matrices à m lignes et n colonnes (de type/taille/format (m, n)) à éléments dans \mathbb{K} .
Convention : une matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$ est notée en majuscules, avec élément $a_{i,j}$ à la position ligne $i \in \{1, \dots, m\}$ et colonne $j \in \{1, \dots, n\}$.
- On note par $0_{m,n} = 0$ la matrice comportant que des éléments 0 (la taille de 0 étant clair du contexte).
- $\mathcal{M}_m(\mathbb{K}) = \mathcal{M}_{m,m}(\mathbb{K})$ l'ensemble des matrices carrées d'ordre m . On note par $I_m = I$ la matrice identité.
- $\mathbb{R}^n \ni x \mapsto y = Ax \in \mathbb{R}^m$ est une application linéaire.
Réciproquement, une application linéaire $\varphi : E = \mathbb{R}^n \mapsto \widetilde{E} = \mathbb{R}^m$ admet une **représentation matricielle** $y = \varphi(x) = Ax$ dans les bases canoniques $\{e_1, \dots, e_n\}$ et $\{\widetilde{e}_1, \dots, \widetilde{e}_m\}$ si A admet les éléments $a_{j,k} = \widetilde{e}_j^* \varphi(e_k)$.
- **Transposée** $A^T \in \mathcal{M}_{n,m}(\mathbb{K})$ de $A \in \mathcal{M}_{m,n}(\mathbb{K})$ avec éléments $(A^T)_{i,j} = a_{j,i}$.
Adjointe $A^* \in \mathcal{M}_{n,m}(\mathbb{K})$ avec éléments $(A^*)_{i,j} = \overline{a_{j,i}}$.

$$\forall x \in \mathbb{K}^n \forall y \in \mathbb{K}^m : \quad \langle y, Ax \rangle = \langle A^* y, x \rangle.$$

Produit scalaire est produit matriciel : $\langle x, y \rangle = x^* y \in \mathbb{K}$ avec vecteur ligne $x^* = (\overline{x_1}, \dots, \overline{x_n})$

- **Image** $Im(A) = \{Ax : x \in \mathbb{K}^n\}$, le **rang** $rang(A) = \dim Im(A)$ et **noyau** $ker(A) = \{x \in \mathbb{K}^n : Ax = 0\}$ d'une matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$.
- Pour $A \in \mathcal{M}_{m,n}(\mathbb{K})$: $rang(A) = rang(A^*) = n - \dim(ker(A))$ et $ker(A)^\perp = Im(A^*)$.

Définition 1.1

Une matrice $A \in \mathcal{M}_n(\mathbb{K})$ est dite

- **inversible** s'il existe $B \in \mathcal{M}_n(\mathbb{K})$ avec $BA = I$ (ou $AB = I$).
Dans ce cas, B est unique, dite inverse de A , et notée A^{-1} ;
- **symétrique** (resp. **hermitienne**) si $A^T = A$ (resp. $A = A^*$) ;
- **orthogonale** (resp. **unitaire**) si $A^T A = I$ (resp. $A^* A = I$) ;
- **normale** si $AA^* = A^* A$;
- **semi-définie positive** si $\forall x \in \mathbb{K}^n, x^* Ax \geq 0$;
- **définie positive** si elle est semi-définie positive, et $x^* Ax = 0$ implique $x = 0$;
- **diagonale** si $a_{j,k} = 0$ pour $j \neq k$;
- **triangulaire supérieure** (resp. **inférieure**) si $a_{j,k} = 0$ pour $j > k$ (resp. $j < k$) .
- **semblable** à $B \in \mathcal{M}_n(\mathbb{K})$ s'il existe une matrice inversible S avec $B = S^{-1}AS$.
Elle est dit **diagonalisable** si elle est semblable à une matrice diagonale.

Définition 1.2

$\lambda \in \mathbb{C}$ est dit **valeur propre** de $A \in \mathcal{M}_n(\mathbb{K})$ s'il existe $x \in \mathbb{C}^n \setminus \{0\}$ dit **vecteur propre** de sorte que $Ax = \lambda x$.

Le couple (λ, x) est dit **élément propre** de A .

On note par $Sp(A)$ dit **spectre** l'ensemble des valeurs propres de A , et par $\chi_A(\lambda) = \det(A - \lambda I)$ le **polynôme caractéristique** de A .

Lemme 1.3

$Sp(A)$ est l'ensemble des racines de χ_A . Par conséquent, A admet au moins une valeur propre, et au plus n .

Définition 1.4

On appelle **rayon spectral** de $A \in \mathcal{M}_n(\mathbb{K})$ la quantité $\rho(A) = \max\{|\lambda| : \lambda \in Sp(A)\}$.

Définition 1.2

$\lambda \in \mathbb{C}$ est dit **valeur propre** de $A \in \mathcal{M}_n(\mathbb{K})$ s'il existe $x \in \mathbb{C}^n \setminus \{0\}$ dit **vecteur propre** de sorte que $Ax = \lambda x$.

Le couple (λ, x) est dit **élément propre** de A .

On note par $Sp(A)$ dit **spectre** l'ensemble des valeurs propres de A , et par $\chi_A(\lambda) = \det(A - \lambda I)$ le **polynôme caractéristique** de A .

Lemme 1.3

$Sp(A)$ est l'ensemble des racines de χ_A . Par conséquent, A admet au moins une valeur propre, et au plus n .

Définition 1.4

On appelle **rayon spectral** de $A \in \mathcal{M}_n(\mathbb{K})$ la quantité $\rho(A) = \max\{|\lambda| : \lambda \in Sp(A)\}$.

Théorème 1.5 (Matrices diagonalisables)

- Il existe une matrice non diagonalisable.
- $A \in \mathcal{M}_n(\mathbb{K})$ est diagonalisable avec $B = S^{-1}AS$ diagonale ssi A admet une base de vecteurs propres donnée par les colonnes de S , avec valeurs propres associées sur la diagonale de B .
- Si $A \in \mathcal{M}_n(\mathbb{K})$ admet n valeurs propres distinctes alors A est diagonalisable.

Définition 1.6

Soit E un espace vectoriel sur le corps \mathbb{K} . Une application $E \ni x \mapsto \|x\| \in \mathbb{R}$ est dite norme si

positivité et séparation $\forall x \in E, \|x\| \geq 0, \quad (\|x\| = 0 \text{ ssi } x = 0);$

homogénéité $\forall x \in E, \forall \alpha \in \mathbb{K} : \quad \|\alpha x\| = |\alpha| \|x\|;$

inégalité triangulaire $\forall x, y \in E : \quad \|x + y\| \leq \|x\| + \|y\|.$

Dans ce cas, E muni de la norme $\|\cdot\|$ est dit espace normé.

Notions de base : normes vectorielles

Définition 1.6

Soit E un espace vectoriel sur le corps \mathbb{K} . Une application $E \ni x \mapsto \|x\| \in \mathbb{R}$ est dite norme si

- | | |
|--------------------------|---|
| positivité et séparation | $\forall x \in E, \ x\ \geq 0, \quad (\ x\ = 0 \text{ ssi } x = 0);$ |
| homogénéité | $\forall x \in E, \forall \alpha \in \mathbb{K} : \quad \ \alpha x\ = \alpha \ x\ ;$ |
| inégalité triangulaire | $\forall x, y \in E : \quad \ x + y\ \leq \ x\ + \ y\ .$ |

Dans ce cas, E muni de la norme $\|\cdot\|$ est dit espace normé.

Exemple 1.7

Pour un entier $n \geq 2$, $\mathbb{K} = \mathbb{R}$, $E = \mathbb{K}^n$ on peut définir trois **normes vectorielles usuelles**

$$\|x\|_2 = \sqrt{\sum_{j=1}^n |x_j|^2}, \quad \|x\|_1 = \sum_{j=1}^n |x_j|, \quad \|x\|_\infty = \max_{j=1, \dots, n} |x_j|.$$

Ces formules donnent également des normes dans le cas $\mathbb{K} = \mathbb{C}$.

Notions de base : des révisions

- 1 Revoir opérations sur les matrices : addition (terme par terme) et multiplication (si dimensions sont compatibles). En général $AB \neq BA$. On ne divise pas par des matrices ! Une matrice inversible doit être carrée !!
- 2 \mathbb{K}^n est un espace vectoriel sur \mathbb{K} , $\mathcal{M}_{m,n}(\mathbb{K})$ aussi.
- 3 \mathbb{K}^n muni avec $\|\cdot\|_p$ pour $p \in \{1, 2, \infty\}$ est un espace normé.
- 4 Rappel sous-espace vectoriel, combinaison linéaire, système libre (=linéairement indépendant), système générateur, dimension, base d'un espace vectoriel. Base orthonormée. Construction par l'algorithme de Gram-Schmidt.
- 5 Tout système libre de $k < n$ vecteurs du \mathbb{K}^n peut se compléter pour former une base du \mathbb{K}^n .
- 6 $(AB)^* = B^* A^*$, de même pour la transposée (et l'inverse si cela existe).
 $\det(AB) = \det(A) \det(B)$.
- 7 Si $A \in \mathcal{M}_{m,n}(\mathbb{K})$, $b \in \mathbb{K}^m$, le système $Ax = b$ admet une solution y si $b \in \text{Im}(A)$.
Dans ce cas, l'ensemble des solutions est donné par $y + \text{Ker}(A)$.
- 8 $A \in \mathcal{M}_n(\mathbb{K})$ est inversible ssi $\det(A) \neq 0$ ssi $\exists B \in \mathcal{M}_n(\mathbb{K})$ avec $AB = I$ (ou $BA = I$).
- 9 le déterminant d'une matrice triangulaire A se calcule en prenant le produit des éléments sur la diagonale.
- 10 $\det(A^{-1}) = 1/\det(A)$, $\det(A^T) = \det(A)$, $(A^*)^{-1} = (A^{-1})^*$.

Notions de base : matrices par blocs et produits par blocs

Soient $A_{i,j} \in \mathcal{M}_{\alpha_i, \beta_j}(\mathbb{K})$, $B_{m,n} \in \mathcal{M}_{\gamma_m, \delta_n}(\mathbb{K})$. Les deux matrices

$$A = \begin{bmatrix} \beta_1 & \beta_2 \\ A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{matrix} \alpha_1 \\ \alpha_2 \end{matrix} \in \mathcal{M}_{\alpha_1 + \alpha_2, \beta_1 + \beta_2}(\mathbb{K}), \quad B = \begin{bmatrix} \delta_1 & \delta_2 \\ B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} \begin{matrix} \gamma_1 \\ \gamma_2 \end{matrix}$$

sont dits **matrices par blocs**.

A et B ont des **blocs de taille compatibles** si $\beta_1 = \gamma_1$ et $\beta_2 = \gamma_2$, et dans ce cas

$$AB = \begin{bmatrix} \delta_1 & \delta_2 \\ A_{1,1}B_{1,1} + A_{1,2}B_{2,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} \end{bmatrix} \begin{matrix} \alpha_1 \\ \alpha_2 \end{matrix} \in \mathcal{M}_{\alpha_1 + \alpha_2, \delta_1 + \delta_2}(\mathbb{K}).$$

Cette formule de **produits par blocs** est généralisable au cas $\alpha_2 = 0$ ou $\delta_2 = 0$, ou au cas de ≥ 3 blocs par lignes et/ou blocs par colonne.

Notions de base : matrices par blocs et produits par blocs

Soient $A_{i,j} \in \mathcal{M}_{\alpha_i, \beta_j}(\mathbb{K})$, $B_{m,n} \in \mathcal{M}_{\gamma_m, \delta_n}(\mathbb{K})$. Les deux matrices

$$A = \begin{bmatrix} \beta_1 & \beta_2 \\ A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix} \begin{matrix} \alpha_1 \\ \alpha_2 \end{matrix} \in \mathcal{M}_{\alpha_1 + \alpha_2, \beta_1 + \beta_2}(\mathbb{K}), \quad B = \begin{bmatrix} \delta_1 & \delta_2 \\ B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \end{bmatrix} \begin{matrix} \gamma_1 \\ \gamma_2 \end{matrix}$$

sont dits **matrices par blocs**.

A et B ont des **blocs de taille compatibles** si $\beta_1 = \gamma_1$ et $\beta_2 = \gamma_2$, et dans ce cas

$$AB = \begin{bmatrix} \delta_1 & \delta_2 \\ A_{1,1}B_{1,1} + A_{1,2}B_{2,1} & A_{1,1}B_{1,2} + A_{1,2}B_{2,2} \\ A_{2,1}B_{1,1} + A_{2,2}B_{2,1} & A_{2,1}B_{1,2} + A_{2,2}B_{2,2} \end{bmatrix} \begin{matrix} \alpha_1 \\ \alpha_2 \end{matrix} \in \mathcal{M}_{\alpha_1 + \alpha_2, \delta_1 + \delta_2}(\mathbb{K}).$$

Cette formule de **produits par blocs** est généralisable au cas $\alpha_2 = 0$ ou $\delta_2 = 0$, ou au cas de ≥ 3 blocs par lignes et/ou blocs par colonne.

Corollaire 1.8

Le produit de deux matrices triangulaires supérieures par blocs est triangulaire supérieure par blocs.

1 Notions et Rappels

2 La factorisation de Schur et la SVD

3 Normes matricielles et conditionnement

4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU

5 L'algo de Gauss sur ordinateur

6 Comment exploiter une structure dans la matrice A ?

7 Le problème des moindres carrés et la décomposition QR

8 Calcul de la décomposition QR pleine par Householder et Givens

9 Calcul numérique de valeurs propres

10 La méthode de la puissance

11 La méthode QR

La factorisation de Schur

Théorème 2.1 (Factorisation de Schur)

*Pour tout $A \in \mathcal{M}_n(\mathbb{C})$ il existe $U \in \mathcal{M}_n(\mathbb{C})$ unitaire de sorte que $T = U^*AU$ est triangulaire supérieure.*

Si A est normale alors T est diagonale.

La factorisation de Schur

Théorème 2.1 (Factorisation de Schur)

Pour tout $A \in \mathcal{M}_n(\mathbb{C})$ il existe $U \in \mathcal{M}_n(\mathbb{C})$ unitaire de sorte que $T = U^*AU$ est triangulaire supérieure.

Si A est normale alors T est diagonale.

Preuve de la première phrase par récurrence sur n .

$n = 1$: $A \in \mathcal{M}_1(\mathbb{K})$ est triangulaire supérieure, donc on peut choisir $U = [1]$, et $T = A$.

$n - 1 \implies n$: d'après le lemme 1.3, il existe un élément propre (λ_1, y_1) de A , avec $\|y_1\| = 1$. Le théorème de la base incomplète nous donne y_2, \dots, y_n de sorte que y_1, \dots, y_n forment une base orthonormée du \mathbb{C}^n . Notons $Q_n = [y_1, \dots, y_n] \in \mathcal{M}_n(\mathbb{C})$ unitaire, alors

$$Q_n^* A Q_n = Q_n^* [\lambda_1 y_1, *] = \begin{bmatrix} \lambda_1 & * \\ 0 & A_{n-1} \end{bmatrix}, \quad A_{n-1} \in \mathcal{M}_{n-1}(\mathbb{C}).$$

Par hypothèse de récurrence, il existe une factorisation de Schur $T_{n-1} = U_{n-1}^* A U_{n-1}$, alors

$U = Q_n \begin{bmatrix} 1 & 0 \\ 0 & U_{n-1} \end{bmatrix}$ est unitaire, et

$$U^* A U = \begin{bmatrix} 1 & 0 \\ 0 & U_{n-1}^* \end{bmatrix} \begin{bmatrix} \lambda_1 & * \\ 0 & A_{n-1} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & U_{n-1} \end{bmatrix} = \begin{bmatrix} \lambda_1 & * \\ 0 & T_{n-1} \end{bmatrix}$$

est triangulaire supérieure.

Il reste à discuter le cas particulier où A est une matrice normale.

M.q. si A est normale alors aussi T :

$$TT^* - T^*T = U^*(AA^* - A^*A)U = 0.$$

M.q. T normale et triangulaire supérieure implique que T est diagonale :

On montrera par récurrence sur l'indice ligne $j = 1, \dots, n-1$ que $t_{j,k} = 0$ pour $k = j+1, \dots, n$.

$$(TT^*)_{j,j} = \sum_{k=j}^n t_{j,k} \overline{t_{j,k}}$$

et

$$(T^*T)_{j,j} = \sum_{k=1}^j \overline{t_{k,j}} t_{k,j} = |t_{j,j}|^2$$

par hypothèse de récurrence, et alors

$$0 = (TT^* - T^*T)_{j,j} = \sum_{k=j+1}^n |t_{j,k}|^2.$$

Conséquence de la factorisation de Schur

Corollaire 2.2 (Diagonalisation d'une matrice hermitienne)

Soit $A \in \mathcal{M}_n(\mathbb{C})$ hermitienne. Alors il existe $U \in \mathcal{M}_n(\mathbb{C})$ unitaire de sorte que la matrice $D = U^*AU$ est diagonale et composée des valeurs propres de A .

Si $A \in \mathcal{M}_n(\mathbb{R})$ alors on peut choisir $U, D \in \mathcal{M}_n(\mathbb{R})$.

Nous déduisons qu'une matrice hermitienne admet une base orthonormée de vecteurs propres.

Décomposition en valeurs singulières (1)

Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$ une matrice rectangulaire.

Lemme 2.3

*Pour toute matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$, les valeurs propres de A^*A sont réelles et ≥ 0 .*

Lemme 2.4

Soient $A \in \mathcal{M}_{m,n}(\mathbb{K})$ et $B \in \mathcal{M}_{n,m}(\mathbb{K})$, alors les valeurs propres non nulles de AB et de BA sont les mêmes (comptant multiplicité).

Décomposition en valeurs singulières (1)

Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$ une matrice rectangulaire.

Lemme 2.3

*Pour toute matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$, les valeurs propres de A^*A sont réelles et ≥ 0 .*

Lemme 2.4

Soient $A \in \mathcal{M}_{m,n}(\mathbb{K})$ et $B \in \mathcal{M}_{n,m}(\mathbb{K})$, alors les valeurs propres non nulles de AB et de BA sont les mêmes (comptant multiplicité).

Définition 2.5 (Valeurs singulières d'une matrice)

*Les valeurs singulières $\mu_1 \geq \mu_2 \geq \dots \geq \mu_n$ d'une matrice $A \in \mathcal{M}_{m,n}(\mathbb{K})$ sont les racines carrées positives ou nulles des valeurs propres de A^*A .*

Théorème 2.6 (Valeurs singulières d'une matrice normale)

Les valeurs singulières d'une matrice normale $A \in \mathcal{M}_n(\mathbb{K})$ sont les modules de ses valeurs propres.

Décomposition en valeurs singulières (2)

Théorème 2.7 (Décomposition en valeurs singulières, SVD)

Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$ avec r valeurs singulières positives. Alors il existe $U \in \mathcal{M}_m(\mathbb{K})$ et $V \in \mathcal{M}_n(\mathbb{K})$ toutes les deux unitaires et $\Sigma \in \mathcal{M}_{m,n}(\mathbb{K})$ "diagonale" tel que

$$A = U\Sigma V^*, \quad \Sigma = \begin{bmatrix} \tilde{\Sigma} & 0 \\ 0 & 0 \end{bmatrix}$$

avec $\tilde{\Sigma} = \text{diag}(\mu_1, \mu_2, \dots, \mu_r)$, où $\mu_1 \geq \dots \geq \mu_r > 0$. En particulier, $\text{rank}(A) = r \leq \min(m, n)$.

Remarques 2.8 (Schéma de calcul de la SVD)

- Calculer les éléments propres (μ_j^2, v_j) de A^*A avec $\mu_1 \geq \dots \geq \mu_r > \mu_{r+1} = \dots = \mu_n = 0$ et $\{v_1, \dots, v_n\}$ base orthonormée du \mathbb{K}^n ;
- Calculer $u_j = Av_j / \mu_j$ pour $j = 1, \dots, r$, ainsi que $\{u_{r+1}, \dots, u_m\}$ base orthonormée de $\ker(A^*) = \ker(AA^*)$;
- Poser $U = (u_1, \dots, u_m)$, $V = (v_1, \dots, v_n)$, Σ comme avant.

NB : Pour se ramener de $n > m$ à $n \leq m$, on peut calculer la SVD $A^* = V\Sigma^*U^*$.

On conclut que si A est réelle alors aussi U et V peuvent être réelles.

SVD (3) : approcher A par une matrice de faible rang

Si on mesure la distance entre deux matrices de même taille par la norme spectrale $\|\cdot\|_2$ discutée au chapitre suivant, on montrera aux TD le résultat important suivant.

Théorème 2.9 (Théorème de Eckhart-Young)

Pour $k \in \{0, \dots, \min(m, n)\}$ la matrice de rang $\leq k$ la plus proche de A est donnée par

$$B = U \begin{bmatrix} \text{diag}(\mu_1, \dots, \mu_k) & 0 \\ 0 & 0 \end{bmatrix} V^* = \sum_{j=1}^k u_j \mu_j v_j^*,$$

de distance donnée par μ_{k+1} .

Pour calculer B il suffit alors de connaître seulement k éléments propres de $A^* A$. Stocker B au lieu de A nécessitera alors $(m+n)k$ emplacements mémoire au lieu de mn , ce qui peut signifier un large gain, en particulier dans le contexte de l'ACP en big data où $k = 2$.

On verra en TP un exemple de compression d'images (un élément d'une matrice correspond au niveau gris du pixel associé).

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algorithme de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Définition 3.1 (norme matricielles)

On se donne pour tout entier $n \geq 1$ une norme vectorielle $||| \cdot |||$ sur \mathbb{K}^n . Une norme $\| \cdot \|$ sur $\mathcal{M}_{m,n}(\mathbb{K})$ est dite

- **compatible avec la norme vectorielle** $||| \cdot |||$ si

$$\forall A \in \mathcal{M}_{m,n}(\mathbb{K}) \forall x \in \mathbb{K}^n : \quad |||Ax||| \leq \|A\| |||x|||;$$

- **sous-multiplicative ou norme matricielle** si

$$\forall A \in \mathcal{M}_{m,n}(\mathbb{K}) \forall B \in \mathcal{M}_{n,\ell}(\mathbb{K}) : \quad \|AB\| \leq \|A\| \|B\|.$$

Normes matricielles et normes compatibles

Définition 3.1 (norme matricielles)

On se donne pour tout entier $n \geq 1$ une norme vectorielle $||| \cdot |||$ sur \mathbb{K}^n . Une norme $\| \cdot \|$ sur $\mathcal{M}_{m,n}(\mathbb{K})$ est dite

- **compatible avec la norme vectorielle** $||| \cdot |||$ si

$$\forall A \in \mathcal{M}_{m,n}(\mathbb{K}) \forall x \in \mathbb{K}^n : \quad |||Ax||| \leq \|A\| |||x|||;$$

- **sous-multiplicative ou norme matricielle** si

$$\forall A \in \mathcal{M}_{m,n}(\mathbb{K}) \forall B \in \mathcal{M}_{n,\ell}(\mathbb{K}) : \quad \|AB\| \leq \|A\| \|B\|.$$

Exemple 3.2 (norme de Frobenius aussi dite norme de Schur)

L'expression

$$\|A\|_s = \sqrt{\sum_{j=1}^m \sum_{k=1}^n |a_{j,k}|^2} = \sqrt{\text{trace}(A^*A)}$$

donne une norme matricielle sur $\mathcal{M}_{m,n}(\mathbb{K})$ qui est compatible avec la norme vectorielle euclidienne $\| \cdot \|_2$.

Lemme 3.3

A toute norme matricielle $\| \cdot \|$ on peut associer une norme vectorielle qui lui soit compatible. En particulier, $\|A\| \geq \rho(A)$ pour tout $A \in \mathcal{M}_n(\mathbb{K})$.

Normes subordonnées

Théorème 3.4 (norme subordonnée)

On se donne pour tout entier $n \geq 1$ une norme vectorielle $\|\cdot\|$ sur \mathbb{K}^n , et $A \in \mathcal{M}_{m,n}(\mathbb{K})$. L'expression

$$\|A\| = \sup_{x \in \mathbb{K}^n, \|x\|=1} \|Ax\| = \max_{x \in \mathbb{K}^n, \|x\|=1} \|Ax\|$$

donne bien une norme matricielle compatible avec la norme vectorielle $\|\cdot\|$ dite **norme matricielle subordonnée à la norme vectorielle $\|\cdot\|$** .

Pour toute norme matricielle subordonnée, $\|I_n\| = 1$. Pour la norme de Schur, $\|I_n\|_S = \sqrt{n}$, elle n'est alors pas subordonnée.

Normes subordonnées

Théorème 3.4 (norme subordonnée)

On se donne pour tout entier $n \geq 1$ une norme vectorielle $\|\cdot\|$ sur \mathbb{K}^n , et $A \in \mathcal{M}_{m,n}(\mathbb{K})$. L'expression

$$\|A\| = \sup_{x \in \mathbb{K}^n, \|x\|=1} \|Ax\| = \max_{x \in \mathbb{K}^n, \|x\|=1} \|Ax\|$$

donne bien une norme matricielle compatible avec la norme vectorielle $\|\cdot\|$ dite **norme matricielle subordonnée à la norme vectorielle $\|\cdot\|$** .

Pour toute norme matricielle subordonnée, $\|I_n\| = 1$. Pour la norme de Schur, $\|I_n\|_S = \sqrt{n}$, elle n'est alors pas subordonnée.

Définition 3.5 (normes subordonnées usuelles)

Pour nos normes vectorielles usuelles $\|\cdot\|_p$, $p = 1, 2, \infty$, on note par $\|A\|_p$ aussi la norme matricielle subordonnée. La norme matricielle $\|\cdot\|_2$ est aussi appelée **norme spectrale**.

Théorème 3.6 (formules pour les normes subordonnées usuelles)

Pour $A \in \mathcal{M}_{m,n}(\mathbb{K})$ nous avons $\|A\|_2 = \sqrt{\rho(A^*A)}$ (la plus grande valeur singulière), et

$$\|A\|_1 = \max_{k=1, \dots, n} \sum_{j=1}^m |a_{j,k}|, \quad \|A\|_\infty = \max_{j=1, \dots, m} \sum_{k=1}^n |a_{j,k}|.$$

Corollaire 3.7 (propriétés de la norme spectrale)

- a) Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$, alors $\|A^*\|_2 = \|A\|_2$.
- b) Soit $A \in \mathcal{M}_n(\mathbb{K})$ une matrice hermitienne, alors $\|A\|_2 = \rho(A)$.
- c) Soient $U \in \mathcal{M}_m(\mathbb{K})$, $V \in \mathcal{M}_n(\mathbb{K})$ unitaires et $A \in \mathcal{M}_{m,n}(\mathbb{K})$, alors $\|UAV\|_2 = \|A\|_2$.

Peut-on être plus précis dans l'inégalité $\|A\| \geq \rho(A)$ du lemme 3.3 ?

Propriétés des normes et du rayon spectral

Corollaire 3.7 (propriétés de la norme spectrale)

- a) Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$, alors $\|A^*\|_2 = \|A\|_2$.
- b) Soit $A \in \mathcal{M}_n(\mathbb{K})$ une matrice hermitienne, alors $\|A\|_2 = \rho(A)$.
- c) Soient $U \in \mathcal{M}_m(\mathbb{K})$, $V \in \mathcal{M}_n(\mathbb{K})$ unitaires et $A \in \mathcal{M}_{m,n}(\mathbb{K})$, alors $\|UAV\|_2 = \|A\|_2$.

Peut-on être plus précis dans l'inégalité $\|A\| \geq \rho(A)$ du lemme 3.3 ?

Théorème 3.8 (théorème de Gelfant)

Soient $A \in \mathcal{M}_n(\mathbb{K})$, et $\epsilon > 0$. Alors on peut construire une norme matricielle subordonnée $\|\cdot\|_*$ avec $\|A\|_* \leq \rho(A) + \epsilon$.

En particulier, pour toute norme matricielle $\|\cdot\|$ et tout $A \in \mathcal{M}_n(\mathbb{K})$,

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

Propriétés des normes et du rayon spectral

Corollaire 3.7 (propriétés de la norme spectrale)

- a) Soit $A \in \mathcal{M}_{m,n}(\mathbb{K})$, alors $\|A^*\|_2 = \|A\|_2$.
- b) Soit $A \in \mathcal{M}_n(\mathbb{K})$ une matrice hermitienne, alors $\|A\|_2 = \rho(A)$.
- c) Soient $U \in \mathcal{M}_m(\mathbb{K})$, $V \in \mathcal{M}_n(\mathbb{K})$ unitaires et $A \in \mathcal{M}_{m,n}(\mathbb{K})$, alors $\|UAV\|_2 = \|A\|_2$.

Peut-on être plus précis dans l'inégalité $\|A\| \geq \rho(A)$ du lemme 3.3 ?

Théorème 3.8 (théorème de Gelfant)

Soient $A \in \mathcal{M}_n(\mathbb{K})$, et $\epsilon > 0$. Alors on peut construire une norme matricielle subordonnée $\|\cdot\|_*$ avec $\|A\|_* \leq \rho(A) + \epsilon$.

En particulier, pour toute norme matricielle $\|\cdot\|$ et tout $A \in \mathcal{M}_n(\mathbb{K})$,

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

Théorème 3.9 (série de von Neumann)

Pour tout $A \in \mathcal{M}_n(\mathbb{K})$, la série $\sum_{k=0}^{\infty} A^k$ converge ssi $\rho(A) < 1$. Dans ce cas, $I - A$ est inversible, et la limite de la série est donnée par $(I - A)^{-1}$.

Si de plus $\|A\| < 1$ pour une norme matricielle alors $\|(I - A)^{-1} - I\| \leq \|A\|/(1 - \|A\|)$.

Conditionnement : motivation.

Dans la suite de ce cours on note par $\|\cdot\|$ une norme vectorielle et la norme matricielle subordonnée (pour les normes usuelles on ajoute un indice $p \in \{1, 2, \infty\}$).

Les erreurs de représentation de nombres machines, mais aussi des erreurs de mesure font que, au lieu de résoudre un système $Ax = b$ pour $A \in \mathcal{M}_n(\mathbb{K})$ et $b \in \mathbb{K}^n$ donné, on résout plutôt sur ordinateur le système perturbé $(A + \Delta A)(x + \Delta x) = b + \Delta b$, et on se demande si la solution obtenue $x + \Delta x$ est proche de la solution désirée x .

Exemple 3.10

On obtient des solutions $x = (-4, 4)^T$, $x + \Delta x = (0.5, 1)^T$ assez éloignées pour les données "proches"

$$A = \begin{bmatrix} 4.2186 & 6.3279 \\ 3.1415 & 4.7123 \end{bmatrix}, \quad b = \begin{bmatrix} 8.4372 \\ 6.2832 \end{bmatrix}, \quad A + \Delta A = \begin{bmatrix} 4.2188 & 6.3279 \\ 3.1416 & 4.7124 \end{bmatrix}, \quad b + \Delta b = \begin{bmatrix} 8.4373 \\ 6.2832 \end{bmatrix}$$

Conditionnement : motivation.

Dans la suite de ce cours on note par $\|\cdot\|$ une norme vectorielle et la norme matricielle subordonnée (pour les normes usuelles on ajoute un indice $p \in \{1, 2, \infty\}$).

Les erreurs de représentation de nombres machines, mais aussi des erreurs de mesure font que, au lieu de résoudre un système $Ax = b$ pour $A \in \mathcal{M}_n(\mathbb{K})$ et $b \in \mathbb{K}^n$ donné, on résout plutôt sur ordinateur le système perturbé $(A + \Delta A)(x + \Delta x) = b + \Delta b$, et on se demande si la solution obtenue $x + \Delta x$ est proche de la solution désirée x .

Exemple 3.10

On obtient des solutions $x = (-4, 4)^T$, $x + \Delta x = (0.5, 1)^T$ assez éloignées pour les données "proches"

$$A = \begin{bmatrix} 4.2186 & 6.3279 \\ 3.1415 & 4.7123 \end{bmatrix}, \quad b = \begin{bmatrix} 8.4372 \\ 6.2832 \end{bmatrix}, \quad A + \Delta A = \begin{bmatrix} 4.2188 & 6.3279 \\ 3.1416 & 4.7124 \end{bmatrix}, \quad b + \Delta b = \begin{bmatrix} 8.4373 \\ 6.2832 \end{bmatrix}$$

Lemme 3.11 (Pire amplification des erreurs relatives)

$$\|A\| \|A^{-1}\| = \sup_{b, \Delta b} \left\{ \frac{\|\Delta x\|}{\|x\|} \Big/ \frac{\|\Delta b\|}{\|b\|} : Ax = b, A(x + \Delta x) = b + \Delta b \right\}.$$

Définition et propriétés du conditionnement

Définition 3.12 (conditionnement)

Pour $A \in \mathcal{M}_n(\mathbb{K})$ on définit le conditionnement $\text{cond}(A) = \|A\| \|A^{-1}\|$, en particulier $\text{cond}_p(A) = \|A\|_p \|A^{-1}\|_p$ pour $p \in \{1, 2, \infty\}$.

Lemme 3.13 (propriétés du conditionnement)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ inversible, alors

- a) $\text{cond}(A) \geq 1$;
- b) $\forall \lambda \in \mathbb{K} \setminus \{0\} : \text{cond}(\lambda A) = \text{cond}(A)$;
- c) $\text{cond}(A^{-1}) = \text{cond}(A)$;
- d) $\text{cond}_2(A) = \mu_1 / \mu_n$ rapport entre plus grande et plus petite valeur singulière ;
- e) $\text{cond}_2(A) = 1$ pour A unitaire.

On dira que A est bien (resp. mal) conditionné si $\text{cond}(A) \approx 1$ (resp. $\text{cond}(A) \gg 1$).

Définition et propriétés du conditionnement

Définition 3.12 (conditionnement)

Pour $A \in \mathcal{M}_n(\mathbb{K})$ on définit le conditionnement $\text{cond}(A) = \|A\| \|A^{-1}\|$, en particulier $\text{cond}_p(A) = \|A\|_p \|A^{-1}\|_p$ pour $p \in \{1, 2, \infty\}$.

Lemme 3.13 (propriétés du conditionnement)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ inversible, alors

- $\text{cond}(A) \geq 1$;
- $\forall \lambda \in \mathbb{K} \setminus \{0\} : \text{cond}(\lambda A) = \text{cond}(A)$;
- $\text{cond}(A^{-1}) = \text{cond}(A)$;
- $\text{cond}_2(A) = \mu_1 / \mu_n$ rapport entre plus grande et plus petite valeur singulière ;
- $\text{cond}_2(A) = 1$ pour A unitaire.

On dira que A est bien (resp. mal) conditionné si $\text{cond}(A) \approx 1$ (resp. $\text{cond}(A) \gg 1$).

Corollaire 3.14 (distance aux matrices non inversibles)

Pour toute matrice $A \in \mathcal{M}_n(\mathbb{K})$ inversible

$$\frac{1}{\text{cond}_2(A)} = \min \left\{ \frac{\|A - B\|_2}{\|A\|_2} : B \in \mathcal{M}_n(\mathbb{K}) \text{ non inversible.} \right\}$$

Estimations d'erreur pour systèmes perturbés

Théorème 3.15

Soient $A, \Delta A \in \mathcal{M}_n(\mathbb{K})$ avec A inversible et $\|A^{-1}\Delta A\| < 1$. Avec $b, \Delta b \in \mathbb{K}^n$, $b \neq 0$, on considère les deux systèmes

$$\begin{cases} Ax = b \\ (A + \Delta A)(x + \Delta x) = b + \Delta b \end{cases}$$

Alors

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \|A^{-1}\Delta A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

Estimations d'erreur pour systèmes perturbés

Théorème 3.15

Soient $A, \Delta A \in \mathcal{M}_n(\mathbb{K})$ avec A inversible et $\|A^{-1}\Delta A\| < 1$. Avec $b, \Delta b \in \mathbb{K}^n$, $b \neq 0$, on considère les deux systèmes

$$\begin{cases} Ax = b \\ (A + \Delta A)(x + \Delta x) = b + \Delta b \end{cases}$$

Alors

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\text{cond}(A)}{1 - \|A^{-1}\Delta A\|} \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right).$$

Remarques 3.16

Comme $\|A^{-1}\Delta A\| \leq \text{cond}(A)\|\Delta A\|/\|A\|$, nous concluons que le système perturbé admet une solution proche de celle de $Ax = b$ tant que

$$\text{cond}(A) \left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|} \right) \ll 1.$$

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algo de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Déjà vu avant : l'élimination de Gauss

Dans les trois chapitres suivants nous souhaitons résoudre un système à n inconnues x_1, \dots, x_n et n équations

$$i = 1, 2, \dots, n : \quad a_{i,1}x_1 + a_{i,2}x_2 + \dots + a_{i,n}x_n = b_i,$$

qui peut s'écrire comme $Ax = b$, avec $A \in \mathcal{M}_n(\mathbb{K})$ supposée inversible, et $b \in \mathbb{K}^n$ donnés. L'hypothèse sur A donne l'existence et unicité de notre inconnue x .

Déjà vu avant : l'élimination de Gauss

Dans les trois chapitres suivants nous souhaitons résoudre un système à n inconnues x_1, \dots, x_n et n équations

$$i = 1, 2, \dots, n : \quad a_{i,1}x_1 + a_{i,2}x_2 + \dots + a_{i,n}x_n = b_i,$$

qui peut s'écrire comme $Ax = b$, avec $A \in \mathcal{M}_n(\mathbb{K})$ supposée inversible, et $b \in \mathbb{K}^n$ donnés. L'hypothèse sur A donne l'existence et unicité de notre inconnue x .

Algorithme 4.1 (élimination de Gauss avec pivotage naturel)

Idée : Avec $A = A^{(1)}$, $b = b^{(1)}$, transformer pour $k = 1, 2, \dots, n-1$ le système $A^{(k)}x = b^{(k)}$ en un système $A^{(k+1)}x = b^{(k+1)}$ équivalent, en créant dans $A^{(k+1)}$ des zéros en colonne k en dessous de la diagonale.

Objectif : Résoudre le système plus simple $A^{(n)}x = b^{(n)}$ avec $A^{(n)}$ triangulaire supérieure par une remontée.

pour $k = 1, \dots, n-1$

on suppose l'hypothèse de pivotage naturel que pivot $a_{k,k}^{(k)} \neq 0$

pour $i = k+1, \dots, n$

calculer multiplicateur $\ell_{i,k} = a_{i,k}^{(k)} / a_{k,k}^{(k)}$

soustraire $\ell_{i,k}$ fois la $k^{\text{ème}}$ équation (ligne pivot) de la $i^{\text{ème}}$ équation

Un exemple et (a droite) un nouveau schéma de calcul

Exemple 4.2 (un exemple 3×3)

| | | | $A^{(k)}$ | $b^{(k)}$ |
|---------|----------------------------|--------------------|-----------|-----------|
| $k = 1$ | $4x_1 + 8x_2 + 12x_3 = 4$ | / | 4 | 8 12 4 |
| | $3x_1 + 8x_2 + 13x_3 = 5$ | / | 3 | 8 13 5 |
| | $2x_1 + 9x_2 + 18x_3 = 11$ | / | 2 | 9 18 11 |
| $k = 2$ | $4x_1 + 8x_2 + 12x_3 = 4$ | / | 4 | 8 12 4 |
| | $0x_1 + 2x_2 + 4x_3 = 2$ | $\ell_{2,1} = 3/4$ | 0 | 2 4 2 |
| | $0x_1 + 5x_2 + 12x_3 = 9$ | $\ell_{3,1} = 2/4$ | 0 | 5 12 9 |
| $k = 3$ | $4x_1 + 8x_2 + 12x_3 = 4$ | / | 4 | 8 12 4 |
| | $0x_1 + 2x_2 + 4x_3 = 2$ | / | 0 | 2 4 2 |
| | $0x_1 + 0x_2 + 2x_3 = 4$ | $\ell_{3,2} = 5/2$ | 0 | 0 2 4 |

- Les pivots sont encadrés par des rectangles, ils sont bien non nuls ;
- les premières k lignes de $[A^{(k)}, b^{(k)}]$ et $[A^{(k+1)}, b^{(k+1)}]$ sont identiques ;
- on note bien que $A^{(3)}$ est triangulaire supérieure ;
- le système $A^{(3)}x = b^{(3)}$ peut être résolu par une remontée :
 - par la troisième équation : $x_3 = 4/2 = 2$,
 - par la deuxième équation : $x_2 = (2 - 4x_3)/2 = -3$,
 - par la première équation : $x_1 = (4 - 8x_2 - 12x_3)/4 = 1$.

Forme de $A^{(k)}$ dans le cas général.

Lemme 4.3 (une étape d'élimination)

Sous l'hypothèse de pivotage naturel, nous avons $A^{(k+1)} = L^{(k)}A^{(k)}$ et $b^{(k+1)} = L^{(k)}b^{(k)}$ pour $k = 1, \dots, n-1$, avec $L^{(k)}$, $A^{(k)}$ et $b^{(k)}$ ayant la forme

$$\begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & 0 \\ 0 & 1 & 0 & & & \vdots \\ \vdots & 0 & 1 & 0 & & \vdots \\ \vdots & \vdots & -\ell_{k+1,k} & 1 & \ddots & \vdots \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & -\ell_{n,k} & 0 & \cdots & 0 \end{bmatrix}, \begin{bmatrix} a_{1,1}^{(1)} & \cdots & \cdots & a_{1,k}^{(1)} & \cdots & a_{1,n}^{(1)} \\ 0 & a_{2,2}^{(2)} & \cdots & a_{1,k}^{(2)} & \cdots & a_{2,n}^{(2)} \\ \vdots & \ddots & \ddots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ 0 & \cdots & 0 & a_{k+1,k}^{(k)} & \cdots & a_{k+1,n}^{(k)} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} \end{bmatrix}, \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_k^{(k)} \\ b_{k+1}^{(k)} \\ \vdots \\ b_n^{(k)} \end{bmatrix}.$$

En particulier $A^{(n)}$ est triangulaire supérieure.

Lemme 4.4

$$L := (L^{(1)})^{-1}(L^{(2)})^{-1} \dots (L^{(n-1)})^{-1} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \ell_{2,1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ \ell_{n,1} & \cdots & \ell_{n,n-1} & 1 \end{bmatrix}$$

La décomposition LU : existence et unicité

Définition 4.5 (définition d'une décomposition LU)

On dira que $A \in \mathcal{M}_n(\mathbb{K})$ admet une décomposition LU si $A = LU$ avec $L \in \mathcal{M}_n(\mathbb{K})$ triangulaire inférieure à diagonale unité, et $U \in \mathcal{M}_n(\mathbb{K})$ triangulaire supérieure.

Théorème 4.6 (unicité)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ inversible. Alors une décomposition LU est unique.

Théorème 4.7 (existence)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ inversible. Alors nous avons équivalence entre

- A admet une décomposition LU ;
- pour $k = 1, 2, \dots, n-1$, la **sous-matrice principale** $[A]_k$ composée des premières k lignes et colonnes de A est inversible ;
- l'hypothèse de pivotage naturel est valable : $a_{k,k}^{(k)} \neq 0$ pour $k = 1, \dots, n-1$.

Dans ce cas, $U = A^{(n)}$ est donné dans le lemme 4.3, et L dans le lemme 4.4.

Remarques 4.8 (utilité d'une décomposition LU)

- $Ax = b$ revient à résoudre $Ly = b$ (par descente) et $Ux = \textcolor{red}{y}$ (par remontée).
- Calcul efficace de $\det(A)$ et l'inverse A^{-1} (voir TD) etc...

Pivotage partiel

Si l'hypothèse de pivotage naturel n'est pas valable, il faudra envisager de permutez avant chaque étape d'élimination. Voici une stratégie dite **pivotage partiel**.

Algorithme 4.9 (élimination de Gauss avec pivotage partiel)

Pour $k = 1, \dots, n - 1$

Chercher l'indice π_k du plus grand élément en module parmi les $a_{i,k}^{(k)}$, $i = k, \dots, n$.

Permuter l'équation d'indice k et l'équation d'indice π_k .

Pour $i = k + 1, \dots, n$

Calculer multiplicateur $\ell_{i,k} = a_{i,k}^{(k)} / a_{k,k}^{(k)}$

Soustraire $\ell_{i,k}$ fois la $k^{\text{ème}}$ équation (**ligne pivot**) de la $i^{\text{ème}}$ équation

On notera par $P^{(k)} \in \mathcal{M}_n(\mathbb{R})$ la matrice de transposition qui échange deux éléments d'indice k et π_k , et laisse les autres composantes invariantes. $P^{(k)}$ est obtenu en remplaçant les colonnes d'indice k et π_k de l'identité I_n par e_{π_k} , et e_k , respectivement. Les formules dans le lemme 4.3 prennent alors la forme $A^{(k+1)} = L^{(k)} P^{(k)} A^{(k)}$ et $b^{(k+1)} = L^{(k)} P^{(k)} b^{(k)}$ pour $k = 1, \dots, n - 1$.

Théorème 4.10 (factorisation LU et pivotage partiel)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ une matrice inversible. Alors la matrice A admet une décomposition LU à permutation près, c'est-à-dire $PA = LU$, où $P = P^{(n-1)} \dots P^{(1)}$ est une matrice de permutation, et L, U sont comme avant à condition que l'on permute simultanément tous les multiplicateurs avec les équations.

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algo de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Remarques 5.1 (stockage sur place)

Une fois $a_{i,j}^{(k+1)}$ calculé, on n'a plus besoin de $a_{i,j}^{(k)}$ et on stockera $a_{i,j}^{(k+1)}$ à sa place, de même pour le second membre qui peut être considéré comme une colonne supplémentaire de A . Cela revient à supprimer l'indice (k) dans l'algo, mais pour plus de clarté on va se servir d'un tableau $M \in \mathcal{M}_{n,n+1}(\mathbb{K})$ initialisé par $M = [A, b]$ pour le stockage sur place.

On stockera aussi $\ell_{i,k}$ à la position (i, k) de M (à la place d'un zéro dans $A^{(k+1)}$).

Remarques 5.1 (stockage sur place)

Une fois $a_{i,j}^{(k+1)}$ calculé, on n'a plus besoin de $a_{i,j}^{(k)}$ et on stockera $a_{i,j}^{(k+1)}$ à sa place, de même pour le second membre qui peut être considéré comme une colonne supplémentaire de A . Cela revient à supprimer l'indice (k) dans l'algo, mais pour plus de clarté on va se servir d'un tableau $M \in \mathcal{M}_{n,n+1}(\mathbb{K})$ initialisé par $M = [A, b]$ pour le stockage sur place.

On stockera aussi $\ell_{i,k}$ à la position (i, k) de M (à la place d'un zéro dans $A^{(k+1)}$).

Remarques 5.2 (vectorisation)

Avec des listes indiquant les indices ligne et colonne des sous-matrices, les formules d'élimination de Gauss

$$\forall i, j > k : \quad \ell_{i,k} = \frac{a_{i,k}^{(k)}}{a_{k,k}^{(k)}}, \quad a_{i,j}^{(k+1)} = a_{i,j}^{(k)} - \ell_{i,k} a_{k,j}^{(k)}, \quad b_i^{(k+1)} = b_i^{(k)} - \ell_{i,k} b_k^{(k)},$$

deviennent les formules **vectorisées** $M[k+1:n, k] = M[k+1:n, k]/M[k, k]$, et pour $i \in [k+1:n]$

$$M[i, k+1:n+1] = M[i, k+1:n+1] - M[i, k] * M[k, k+1:n+1].$$

. Ces formules vectorisées sont plus rapides à exécuter sous python.

Triangulation de Gauss

Algorithme 5.3 (Triangulation de Gauss et descente, pivotage partiel)

Initialiser $M = [A, b]$, $n = \text{ordre de } A$.

Pour $k = 1, 2, \dots, n - 1$

Chercher $\pi_k \in [k:n]$ tel que $|M[\pi_k, k]| = \max(|M[k:n, k]|)$

Permuter les lignes k et π_k de M

$M[k+1:n, k] = M[k+1:n, k] / M[k, k]$

Pour $i = k + 1, \dots, n$

$M[i, k+1:n+1] = M[i, k+1:n+1] - M[i, k] * M[k, k+1:n+1]$

On pourrait supprimer aussi la boucle i en calculant directement

$M[k+1:n, k+1:n+1] = M[k+1:n, k+1:n+1] - M[k+1:n, k] * M[k, k+1:n+1]$.

Triangulation de Gauss

Algorithme 5.3 (Triangulation de Gauss et descente, pivotage partiel)

Initialiser $M = [A, b]$, $n = \text{ordre de } A$.

Pour $k = 1, 2, \dots, n - 1$

Chercher $\pi_k \in [k:n]$ tel que $|M[\pi_k, k]| = \max(|M[k:n, k]|)$

Permuter les lignes k et π_k de M

$M[k+1:n, k] = M[k+1:n, k] / M[k, k]$

Pour $i = k + 1, \dots, n$

$M[i, k+1:n+1] = M[i, k+1:n+1] - M[i, k] * M[k, k+1:n+1]$

On pourrait supprimer aussi la boucle i en calculant directement

$M[k+1:n, k+1:n+1] = M[k+1:n, k+1:n+1] - M[k+1:n, k] * M[k, k+1:n+1]$.

Avec M (et le tableau π) en sortie, nous obtenons la factorisation $PA = LU$ et $Lb^{(n)} = Pb$, avec la matrice de permutation P comme avant, et

$$L = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{2,1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ m_{n,1} & \cdots & m_{n,n-1} & 1 \end{bmatrix}, \quad U = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,n} \\ 0 & m_{2,2} & \cdots & m_{2,n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & m_{n,n} \end{bmatrix}, \quad b^{(n)} = \begin{bmatrix} m_{1,n+1} \\ m_{2,n+1} \\ \vdots \\ m_{n,n+1} \end{bmatrix},$$

Remontée et complexité

Après avoir calculé M par l'algo 5.3, l'algo suivant résout $Ux = b^{(n)}$ (remontée).

Algorithme 5.4 (Remontée pour trouver solution de $Ax = b$)

Initialiser $n = \text{nombre de lignes de } M$, $x = 0 \in \mathbb{R}^n$

Pour $i = n, n-1, \dots, 2, 1$

*$x[i] = (M[i, n+1] - M[i, i+1:n] * x[i+1:n]) / M[i, i]$*

Remontée et complexité

Après avoir calculé M par l'algorithme 5.3, l'algorithme suivant résout $Ux = b^{(n)}$ (remontée).

Algorithme 5.4 (Remontée pour trouver solution de $Ax = b$)

Initialiser $n = \text{nombre de lignes de } M$, $x = 0 \in \mathbb{R}^n$

Pour $i = n, n-1, \dots, 2, 1$

$x[i] = (M[i, n+1] - M[i, i+1:n] * x[i+1:n]) / M[i, i]$

Théorème 5.5 (complexité de l'algorithme de Gauss)

L'algorithme 5.3 de triangulation nécessite $\mathcal{O}(n^2)$ espaces mémoire et $\frac{2}{3}n^3 + \mathcal{O}(n^2)$ opérations élémentaires. L'algorithme 5.4 de la remonté nécessite $\mathcal{O}(n)$ espaces mémoire et $n^2 + \mathcal{O}(n)$ opérations élémentaires.

Démonstration.

Commençons à compter le nombre d'opérations pour la remontée

$$\sum_{i=1}^n \left(\underbrace{2}_{\text{division et soustraction}} + \underbrace{(2(n-i)-1)}_{\text{produit scalaire}} \right) = n^2 + \mathcal{O}(n),$$

et pour la triangulation

$$\sum_{k=1}^n \left(\underbrace{n-k}_{\text{divisions}} + \underbrace{2(n-k)(n-k+1)}_{\text{produits et soustractions}} \right) = 2 \sum_{\ell=0}^{n-1} \ell^2 + \mathcal{O}(n^2) = \frac{2}{3}n^3 + \mathcal{O}(n^2).$$

L'algo de Gauss en précision finie.

Théorème 5.6 (sans preuve)

Soient \widehat{L} , \widehat{U} , \widehat{P} calculés sur ordinateur avec précision machine ϵ , alors

$$\|\widehat{P}A - \widehat{L}\widehat{U}\|_\infty \leq 2\epsilon n^2 \gamma(A),$$

avec le facteur de grossissement $\gamma(A) = \max_{i,j,k} |A_{i,j}^{(k)}|$.

- a) Sans pivotage, $\gamma(A)$ peut être arbitrairement plus grand que $\|A\|_\infty$.
- b) Dans le cas de pivotage partiel, $\gamma(A) \leq 2^{n-1} \|A\|_\infty$ (voir TD).
- c) Si A est hermitienne définie positive alors $\gamma(A) \leq \|A\|_\infty$.

L'algo de Gauss en précision finie.

Théorème 5.6 (sans preuve)

Soient \widehat{L} , \widehat{U} , \widehat{P} calculés sur ordinateur avec précision machine ϵ , alors

$$\|\widehat{P}A - \widehat{L}\widehat{U}\|_\infty \leq 2\epsilon n^2 \gamma(A),$$

avec le facteur de grossissement $\gamma(A) = \max_{i,j,k} |A_{i,j}^{(k)}|$.

- a) Sans pivotage, $\gamma(A)$ peut être arbitrairement plus grand que $\|A\|_\infty$.
- b) Dans le cas de pivotage partiel, $\gamma(A) \leq 2^{n-1} \|A\|_\infty$ (voir TD).
- c) Si A est hermitienne définie positive alors $\gamma(A) \leq \|A\|_\infty$.

Exemple 5.7

Avec pivotage naturel, $\delta > 0$ "petit", et arithmétique exacte

$$A = \begin{bmatrix} \delta & 1 \\ 1 & 1 \end{bmatrix} = LU, \quad L = \begin{bmatrix} 1 & 0 \\ 1/\delta & 1 \end{bmatrix}, \quad U = \begin{bmatrix} \delta & 1 \\ 0 & 1 - 1/\delta \end{bmatrix}.$$

et en précision finie, pivotage naturel, $\delta \ll \epsilon$,

$$\widehat{L} = L, \quad \widehat{U} = \begin{bmatrix} \delta & 1 \\ 0 & -1/\delta \end{bmatrix}, \quad \widehat{P} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \|\widehat{P}A - \widehat{L}\widehat{U}\|_\infty = 1.$$

Et avec pivotage partiel ?

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algorithme de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Exploiter la symétrie

Théorème 6.1 (décomposition de Crout)

Si $A \in \mathcal{M}_n(\mathbb{K})$ inversible et hermitienne admet une décomposition $A = LU$, alors elle admet aussi une unique décomposition $A = LDL^$ avec D diagonale et réelle.*

Exploiter la symétrie

Théorème 6.1 (décomposition de Crout)

Si $A \in \mathcal{M}_n(\mathbb{K})$ inversible et hermitienne admet une décomposition $A = LU$, alors elle admet aussi une unique décomposition $A = LDL^$ avec D diagonale et réelle.*

Théorème 6.2 (décomposition de Cholesky)

Si $A \in \mathcal{M}_n(\mathbb{K})$ est hermitienne définie positive alors elle admet une unique décomposition $A = CC^$ avec C triangulaire inférieure ayant des éléments diag. > 0 .*

Exploiter la symétrie

Théorème 6.1 (décomposition de Crout)

Si $A \in \mathcal{M}_n(\mathbb{K})$ inversible et hermitienne admet une décomposition $A = LU$, alors elle admet aussi une unique décomposition $A = LDL^*$ avec D diagonale et réelle.

Théorème 6.2 (décomposition de Cholesky)

Si $A \in \mathcal{M}_n(\mathbb{K})$ est hermitienne définie positive alors elle admet une unique décomposition $A = CC^*$ avec C triangulaire inférieure ayant des éléments diag. > 0 .

Remarques 6.3

Comment calculer des telles décompositions si on sait qu'elles existent ? Prenons le cas de la décomposition de Crout, où les inconnues sont les éléments de L sous la diagonale (il y a des 1 sur la diagonale et les 0 au dessus de la diagonale), ainsi que les éléments de D sur la diagonale. Alors pour $1 \leq j \leq i \leq n$ on obtient terme par terme

$$a_{i,j} = \sum_{k=1}^j \ell_{i,k} d_{k,k} \overline{\ell_{j,k}}, \quad \ell_{i,i} = \ell_{j,j} = 1.$$

En parcourant $(i, j) = (1, 1), (2, 1), \dots, (n, 1), (2, 2), (3, 2), \dots, (n, 2), \dots, (n, n)$ on peut résoudre pour la seule inconnue $d_{i,i}$ si $i = j$, et $\ell_{i,j}$ si $i > j$ (car les autres quantités dans notre formule ont été déjà calculées avant).

On conclut que **calculer la factorisation de Cholesky** nécessite $n^3/3 + O(n^2)$ op. arithm., la moitié de ce que l'on avait trouvé pour la triangulation de Gauss.

Exploiter des 0 dans A

Après tout, le but de l'élimination de Gauss est de créer des zéros dans A .

Ne-peut-on pas réduire la complexité si A contient déjà beaucoup de zéros, par exemple A est une matrice tridiagonale, c'est-à-dire, $a_{i,j} = 0$ pour $|i - j| > 1$?

Théorème 6.4 (théorème du front)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ inversible admettant une décomposition $A = LU$. Notons par

$$\text{front}(A) = (j(1), j(2), \dots, j(n)),$$

avec $j(i)$ l'indice colonne du premier élément non nul dans la ligne i de A . Alors $\text{front}(L) = \text{front}(A)$ et $\text{front}(U^T) = \text{front}(A^T)$.

Démonstration.

Technique similaire à la remarque 6.3.



Exploiter des 0 dans A

Après tout, le but de l'élimination de Gauss est de créer des zéros dans A.

Ne-peut-on pas réduire la complexité si A contient déjà beaucoup de zéros, par exemple A est une matrice tridiagonale, c'est-à-dire, $a_{i,j} = 0$ pour $|i - j| > 1$?

Théorème 6.4 (théorème du front)

Soit $A \in \mathcal{M}_n(\mathbb{K})$ inversible admettant une décomposition $A = LU$. Notons par

$$\text{front}(A) = (j(1), j(2), \dots, j(n)),$$

avec $j(i)$ l'indice colonne du premier élément non nul dans la ligne i de A. Alors $\text{front}(L) = \text{front}(A)$ et $\text{front}(U^T) = \text{front}(A^T)$.

Démonstration.

Technique similaire à la remarque 6.3. □

Exemple 6.5

Soit A une matrice tridiagonale irreductible, c'est-à-dire, $a_{i,j} = 0$ pour $|i - j| > 1$ et $a_{i,j} \neq 0$ pour $|i - j| = 1$. Alors $\text{front}(A) = \text{front}(A^T) = (1, 1, 2, 3, \dots, n - 1)$.

Par conséquent, $\text{front}(L) = (1, 1, 2, 3, \dots, n - 1)$ et L est une matrice bidiagonale inférieure à diagonale unité. Aussi $\text{front}(U^T) = (1, 1, 2, 3, \dots, n - 1)$ et U est une matrice bidiagonale supérieure.

On conclut que le calcul de L et U nécessite $O(n)$ opérations arithmétiques, bien moins que ce que l'on avait trouvé pour la triangulation de Gauss.

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algo de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Position du problème des moindres carrés

Dans les 2 chapitres suivants on considérera seulement $\mathbb{K} = \mathbb{R}$, et on se donne $A \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $\text{rank}(A) = n \leq m$, et $b \in \mathbb{R}^m$.

Définition 7.1 (le problème des moindres carrés)

Trouver $\underline{x} \in \mathbb{R}^n$ de sorte que $\|A\underline{x} - b\|_2 \leq \|Ax - b\|_2$ pour tout $x \in \mathbb{R}^n$.

Position du problème des moindres carrés

Dans les 2 chapitres suivants on considérera seulement $\mathbb{K} = \mathbb{R}$, et on se donne $A \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $\text{rank}(A) = n \leq m$, et $b \in \mathbb{R}^m$.

Définition 7.1 (le problème des moindres carrés)

Trouver $\underline{x} \in \mathbb{R}^n$ de sorte que $\|A\underline{x} - b\|_2 \leq \|Ax - b\|_2$ pour tout $x \in \mathbb{R}^n$.

Exemple 7.2 (la droite de régression)

En physique ou en statistique on se pose le problème comment faire passer au mieux une droite par un nuage de $m \geq 2$ points $\{(t_j, b_j)^T \in \mathbb{R}^2 : j = 1, \dots, m\}$, que l'on suppose non alignés sur une droite verticale. On cherche alors une droite de la forme $g(t) = x_1 + x_2 t$ avec abscisse $x_1 \in \mathbb{R}$ et pente $x_2 \in \mathbb{R}$ de sorte que, avec la distance des valeurs $\delta_j = g(t_j) - b_j$ en t_j , la quantité

$$\sum_{j=1}^m \delta_j^2 = \|Ax - b\|_2^2, \quad \text{avec} \quad A = \begin{bmatrix} 1 & t_1 \\ 1 & t_2 \\ \vdots & \vdots \\ 1 & t_m \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

soit la plus petite possible. Notons que $\text{rank}(A) = 2$ par hypothèse sur le nuage de points. En TD/TP, on cherchera à faire passer des fonctions plus compliquées par un nuage de points.

L'existence et l'unicité d'une solution

Théorème 7.3 (solution d'un problème de moindres carrés)

L'unique solution \underline{x} du problème des moindres carrés est l'unique solution \underline{x} du système des équations normales $A^T A \underline{x} = A^T b$.

Nous donnons une preuve algébrique, alternativement on pourrait donner une preuve analytique en calculant les points annulant le gradient de

$\mathbb{R}^n \ni x \mapsto f(x) = \|Ax - b\|_2^2$ et déterminer la nature du Hessien.

L'existence et l'unicité d'une solution

Théorème 7.3 (solution d'un problème de moindres carrés)

L'unique solution \underline{x} du problème des moindres carrés est l'unique solution \underline{x} du système des équations normales $A^T A \underline{x} = A^T b$.

Nous donnons une preuve algébrique, alternativement on pourrait donner une preuve analytique en calculant les points annulant le gradient de $\mathbb{R}^n \ni x \mapsto f(x) = \|Ax - b\|_2^2$ et déterminer la nature du Hessien.

Définition 7.4 (décomposition QR (pleine ou économique))

*On dira que $A = QR$ est une **décomposition QR (pleine)** si*

$$Q \in \mathcal{M}_m(\mathbb{R}) \text{ orthogonale}, \quad R = \begin{bmatrix} \tilde{R} \\ 0_{m-n,n} \end{bmatrix} \in \mathcal{M}_{m,n}(\mathbb{R}),$$

avec $\tilde{R} \in \mathcal{M}_n(\mathbb{R})$ triangulaire supérieure à diagonale > 0 .

*On dira que $A = \tilde{Q}\tilde{R}$ est une **décomposition QR économique** si $\tilde{Q} \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $\tilde{Q}^T \tilde{Q} = I_n$ (\tilde{Q} est à colonnes orthonormées), et \tilde{R} comme avant.*

L'existence et l'unicité d'une solution

Théorème 7.3 (solution d'un problème de moindres carrés)

L'unique solution \underline{x} du problème des moindres carrés est l'unique solution \underline{x} du système des équations normales $A^T A \underline{x} = A^T b$.

Nous donnons une preuve algébrique, alternativement on pourrait donner une preuve analytique en calculant les points annulant le gradient de $\mathbb{R}^n \ni x \mapsto f(x) = \|Ax - b\|_2^2$ et déterminer la nature du Hessien.

Définition 7.4 (décomposition QR (pleine ou économique))

*On dira que $A = QR$ est une **décomposition QR (pleine)** si*

$$Q \in \mathcal{M}_m(\mathbb{R}) \text{ orthogonale}, \quad R = \begin{bmatrix} \tilde{R} \\ 0_{m-n,n} \end{bmatrix} \in \mathcal{M}_{m,n}(\mathbb{R}),$$

avec $\tilde{R} \in \mathcal{M}_n(\mathbb{R})$ triangulaire supérieure à diagonale > 0 .

*On dira que $A = \tilde{Q}\tilde{R}$ est une **décomposition QR économique** si $\tilde{Q} \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $\tilde{Q}^T \tilde{Q} = I_n$ (\tilde{Q} est à colonnes orthonormées), et \tilde{R} comme avant.*

Lemme 7.5

Il existe une et une seule décomposition QR économique $A = \tilde{Q}\tilde{R}$.

L'existence et l'unicité d'une solution

Théorème 7.3 (solution d'un problème de moindres carrés)

L'unique solution \underline{x} du problème des moindres carrés est l'unique solution \underline{x} du système des équations normales $A^T A \underline{x} = A^T b$.

Nous donnons une preuve algébrique, alternativement on pourrait donner une preuve analytique en calculant les points annulant le gradient de $\mathbb{R}^n \ni x \mapsto f(x) = \|Ax - b\|_2^2$ et déterminer la nature du Hessien.

Définition 7.4 (décomposition QR (pleine ou économique))

*On dira que $A = QR$ est une **décomposition QR (pleine)** si*

$$Q \in \mathcal{M}_m(\mathbb{R}) \text{ orthogonale}, \quad R = \begin{bmatrix} \tilde{R} \\ 0_{m-n,n} \end{bmatrix} \in \mathcal{M}_{m,n}(\mathbb{R}),$$

avec $\tilde{R} \in \mathcal{M}_n(\mathbb{R})$ triangulaire supérieure à diagonale > 0 .

*On dira que $A = \tilde{Q}\tilde{R}$ est une **décomposition QR économique** si $\tilde{Q} \in \mathcal{M}_{m,n}(\mathbb{R})$ avec $\tilde{Q}^T \tilde{Q} = I_n$ (\tilde{Q} est à colonnes orthonormées), et \tilde{R} comme avant.*

Lemme 7.5

Il existe une et une seule décomposition QR économique $A = \tilde{Q}\tilde{R}$.

Remarques 7.6

Dans la définition 7.4, on doit prendre comme \tilde{Q} la matrice formée par les premières n colonnes de Q . Par contre, les autres colonnes de Q ne sont pas uniques.

Corollaire 7.7

La solution \underline{x} du problème des moindres carrés est l'unique solution \underline{x} du système $\tilde{R}\underline{x} = \tilde{Q}^T b$, avec $A = \tilde{Q}\tilde{R}$ une décomposition QR économique.

Corollaire 7.7

La solution \underline{x} du problème des moindres carrés est l'unique solution \underline{x} du système $\tilde{R}\underline{x} = \tilde{Q}^T b$, avec $A = \tilde{Q}\tilde{R}$ une décomposition QR économique.

Remarques 7.8

Le passage par une décomposition QR économique est préférable à la résolution du système des équations normales car

$$\text{cond}_2(A^T A) = \text{cond}_2(\tilde{R}^T \tilde{R}) = \left(\text{cond}_2(\tilde{R}) \right)^2 \gg \text{cond}_2(\tilde{R})$$

Si $\text{cond}_2(\tilde{R})$ est encore trop élevé, on cherche à régulariser, par exemple par une méthode de Tichonov où on cherche à minimiser $x \mapsto \|Ax - b\|_2^2 + \lambda \|x\|_2^2$ pour un $\lambda > 0$ approprié. Alternativement, on cherche à approcher d'abord A par une matrice de rang plus faible, parfois par des heuristiques comme la décomposition QR avec pivotage des colonnes, ou par des techniques inspirées par la SVD et le théorème de Eckhart-Young.

L'algorithme de Gram-Schmidt

Avec l'hypothèse habituelle $\text{rank}(A) = n$, notre décomposition QR économique $A = \tilde{Q}\tilde{R}$ implique que $\text{Im}(A) = \text{Im}(\tilde{Q})$, autrement dit, les colonnes $q_1, \dots, q_n \in \mathbb{R}^m$ de \tilde{Q} forment une base orthonormée de l'espace engendré par les colonnes (libres) $a_1, \dots, a_n \in \mathbb{R}^m$ de A . Une telle base orthonormée est construite par l'algo de Gram-Schmidt vu en deuxième année.

L'algorithme de Gram-Schmidt

Avec l'hypothèse habituelle $\text{rank}(A) = n$, notre décomposition QR économique $A = \tilde{Q}\tilde{R}$ implique que $\text{Im}(A) = \text{Im}(\tilde{Q})$, autrement dit, les colonnes $q_1, \dots, q_n \in \mathbb{R}^m$ de \tilde{Q} forment une base orthonormée de l'espace engendré par les colonnes (libres) $a_1, \dots, a_n \in \mathbb{R}^m$ de A . Une telle base orthonormée est construite par l'algo de Gram-Schmidt vu en deuxième année.

Algorithme 7.9 (décomposition QR économique par GS modifié)

Objectif : Avec $A = (a_1, \dots, a_n)$, construire $\tilde{Q} = (q_1, \dots, q_n)$ à colonnes q_j orthonormées, ainsi que les éléments non nuls de \tilde{R} d'une décomposition QR économique $A = \tilde{Q}\tilde{R}$.

Pour $k = 1, \dots, n$ faire

 poser $y = a_k$

 pour $j = 1, \dots, k-1$ faire

$\tilde{r}_{j,k} = (q_j, y)$, $y = y - \tilde{r}_{j,k} q_j$ (rendre y orthogonal à q_j)

$\tilde{r}_{k,k} = \|y\|_2 > 0$, normaliser $q_k = y/\tilde{r}_{k,k}$.

L'algorithme de Gram-Schmidt

Avec l'hypothèse habituelle $\text{rank}(A) = n$, notre décomposition QR économique $A = \tilde{Q}\tilde{R}$ implique que $\text{Im}(A) = \text{Im}(\tilde{Q})$, autrement dit, les colonnes $q_1, \dots, q_n \in \mathbb{R}^m$ de \tilde{Q} forment une base orthonormée de l'espace engendré par les colonnes (libres) $a_1, \dots, a_n \in \mathbb{R}^m$ de A . Une telle base orthonormée est construite par l'algo de Gram-Schmidt vu en deuxième année.

Algorithme 7.9 (décomposition QR économique par GS modifié)

Objectif : Avec $A = (a_1, \dots, a_n)$, construire $\tilde{Q} = (q_1, \dots, q_n)$ à colonnes q_j orthonormées, ainsi que les éléments non nuls de \tilde{R} d'une décomposition QR économique $A = \tilde{Q}\tilde{R}$.

Pour $k = 1, \dots, n$ faire

 poser $y = a_k$

 pour $j = 1, \dots, k-1$ faire

$\tilde{r}_{j,k} = (q_j, y)$, $y = y - \tilde{r}_{j,k} q_j$ (rendre y orthogonal à q_j)

$\tilde{r}_{k,k} = \|y\|_2 > 0$, normaliser $q_k = y / \tilde{r}_{k,k}$.

Remarques 7.10

- Pour k fixe, on doit calculer k produits scalaires et k combinaisons linéaires dans \mathbb{R}^m , donc en total un nombre d'opérations arithmétique de $\sum_{k=1}^n (4km + \mathcal{O}(m)) = 2mn^2 + \mathcal{O}(mn)$ (+ n racines).
- Attention, même si $\tilde{r}_{k,k} \neq 0$, il peut être petit... en fait $|\tilde{r}_{k,k}| / \|\tilde{R}\|_2 \geq 1 / \text{cond}_2(\tilde{R})$.
- En précision finie, on peut avoir perte d'orthogonalité $\|I_n - \tilde{Q}^T \tilde{Q}\|_2$ de l'ordre de grandeur $\epsilon \text{cond}_2(\tilde{R})$. Ceci peut être encore plus grand dans GS classique=version vectorisée où on remplace la boucle j par
$$\tilde{R}[1:k-1, k] = \tilde{Q}[:, 1:k-1]^T A[:, k]; y = A[:, k] - \tilde{Q}[:, 1:k-1] * \tilde{R}[1:k-1, k].$$

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algorithme de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

On suppose pour un instant que, pour tout $\ell \geq 2$ et $y \in \mathbb{R}^\ell$ il existe une matrice orthogonale $H(y) \in \mathcal{M}_\ell(\mathbb{R})$ de sorte que $H(y)y$ est un multiple du premier vecteur canonique e_1 dans \mathbb{R}^ℓ . Ces matrices seront construites par la suite.

Théorème 8.1 (une étape d'élimination dans une factorisation QR pleine)

Avec des matrices orthogonales $H^{(k)}$ comme ci-dessous, nous posons $A^{(1)} = A$ et $A^{(k+1)} = H^{(k)}A^{(k)}$ pour $k = 1, \dots, p-1$, $p = \min(m, n+1)$. Alors $A^{(k)}$ aura la forme

$$y^{(k)} = \begin{bmatrix} a_{1,1}^{(k)} \\ a_{2,1}^{(k)} \\ \vdots \\ a_{m,1}^{(k)} \end{bmatrix}, \quad H^{(k)} = \left[\begin{array}{c|c} I_{k-1} & 0 \\ \hline 0 & H(y^{(k)}) \end{array} \right], \quad A^{(k)} = \begin{bmatrix} a_{1,1}^{(k)} & \cdots & \cdots & a_{1,k}^{(k)} & \cdots & a_{1,n}^{(k)} \\ 0 & a_{2,2}^{(k)} & \cdots & a_{1,k}^{(k)} & \cdots & a_{2,n}^{(k)} \\ \vdots & \ddots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & a_{m,k}^{(k)} & \cdots & a_{m,n}^{(k)} \end{bmatrix}.$$

En particulier, avec une matrice $E = \text{diag}(\pm 1, \dots, \pm 1) \in \mathcal{M}_m(\mathbb{R})$ appropriée, $R = EA^{(p)} \in \mathcal{M}_{m,n}(\mathbb{R})$ et $Q^T = EH^{(p-1)} \dots H^{(1)}$ nous obtenons la décomposition QR pleine $A = QR$.

NB : les premières $k-1$ lignes dans $A^{(k)}$ et $A^{(k+1)}$ sont les mêmes (possibilité de stockage sur place).

Matrices de Householder

Définition 8.2

Étant donné $w \in \mathbb{R}^m \setminus \{0\}$, la **matrice de Householder** $H = H_w$ est définie par

$$H = I_m - 2 \frac{ww^T}{w^T w}$$

Attention : $w^T w \in \mathbb{R}$ mais $ww^T \in \mathcal{M}_m(\mathbb{R})$.

Lemme 8.3 (propriétés d'une matrice de Householder)

Une matrice de Householder est symétrique et orthogonale. H représente une matrice de symétrie par rapport au hyperplan $\{x \in \mathbb{R}^m : (x, w) = 0\}$.

Lemme 8.4 (élimination avec une matrice de Householder)

Soit $y \in \mathbb{R}^m \setminus \{0\}$, et $\alpha = -\|y\|_2$ si $y_1 > 0$, et $\alpha = \|y\|_2$ si $y_1 \leq 0$. Alors le vecteur

$$w = y - \alpha e_1 \quad \text{avec} \quad w^T w = 2\alpha(\alpha - y_1)$$

est de sorte que la matrice de Householder $H = H_w$ vérifie $Hy = \alpha e_1$.

Remarques 8.5 (Calcul efficace d'un produit de H_w avec un vecteur)

Un produit entre la matrice de Householder H_w et $x \in \mathbb{R}^m$ peut être implémenté par les formules $\beta = 2(w^T x)/(w^T w)$ et $Hx = x - \beta w$, en $4m + O(1)$ opérations arithmétiques (si on ne compte pas la norme).

Une combinaison de 8.1–8.5 donne l'algorithme suivant où on stocke sur place les matrices $A^{(k)}$, ainsi dans H les produits partiels $H^{(k)} \dots H^{(1)}$. Ici on néglige E .

Algorithme 8.6 (décomp. QR pleine $A = QR$ avec algo de Householder)

Objectif : calculer $H = Q^T$ et $R = A^{(p)}$ (dans A)

Poser $(m; n) = \text{taille de } A$, $p = \min(m, n + 1)$, $H = I_m$

Pour $k = 1, \dots, p - 1$ faire

créer zéros en colonne k par le lemme 8.4, $y = y^{(k)}$

poser $y = A[k:m, k]$, $\alpha = \text{signe}(-y_1) \|y\|$, $\gamma = 1/(\alpha(\alpha - y_1))$, $w = y - \alpha e_1$

poser $A[k, k] = \alpha$, $A[k+1:m, k] = 0$

produit rapide $H^{(k)} A^{(k)}$ par les remarques 8.5

poser $\beta = \gamma w^T A[k:m, k+1:n]$, $A[k:m, k+1:n] = A[k:m, k+1:n] - w\beta$

produit rapide $H^{(k)} H$ par les remarques 8.5

poser $\beta = \gamma w^T H[k:m, 1:m]$,

poser $H[k:m, 1:m] = H[k:m, 1:m] - w\beta$

Lemme 8.7 (complexité pour l'algorithme de Householder)

Pour k fixe : $2(m - k) + \mathcal{O}(1)$ OA pour le calcul de w et sa norme,

$4(n - k)(m - k) + \mathcal{O}(m)$ OA pour A , et $4m(m - k) + \mathcal{O}(m)$ OA pour H . En total

$$\underbrace{\frac{4n^3}{3} + 2(m - n)n^2}_{\text{calcul de } R} + \underbrace{2mn^2 + 4(m - n)mn + \mathcal{O}(mn)}_{\text{calcul de } Q} \text{ OA} + n \text{ racines.}$$

Simplifications

Remarques 8.8 (résolution d'un problème de moindres carrés)

Si on veut juste résoudre notre problème de moindres carrés, on n'a pas besoin de Q mais de R et de $b^{(n)} = Q^* b$. Ceci peut se calculer par les formules $b^{(1)} = b$, et $b^{(k+1)} = H^{(k)} b^{(k)}$ pour $k = 1, \dots, n-1$ (ou dans une colonne supplémentaire de A).

Remarques 8.9 (simplifications pour le cas Hessenberg)

Si $m = n$ et A est une matrice de Hessenberg, c'est-à-dire, $a_{j,k} = 0$ pour $j > k+1$, on peut réduire la complexité en montrant par récurrence sur k que toutes les matrices $A^{(k)}$ dans le théorème 8.1 ont une forme Hessenberg. En particulier, seulement les deux premières composantes des vecteurs $y = y^{(k)}$ et w dans l'algorithme 8.6 sont non-nulles. Ceci implique que la matrice $H^{(k)}$ est l'identité I_m dont on a changé les éléments aux quatre positions $(k-1 : k, k-1 : k)$, et donc Q est aussi de forme Hessenberg. Un examen plus fin permet de réduire la complexité à $O(n^2)$ OA pour obtenir une décomposition QR de A .

Remarques 8.10 (simplifications pour le cas d'une matrice tridiagonale)

Si de plus A est une matrice tridiagonale, c'est-à-dire, aussi $a_{j,k} = 0$ pour $k > j+1$, en se basant sur l'étude précédente on peut montrer que $A^{(k)}$ contient au plus trois éléments non nuls par ligne, en ligne i aux positions $(i, i : i+2)$ si $i \leq k$, et aux positions $(i, i-1 : i+1)$ si $i > k$. En particulier, les éléments non nuls de R se trouvent sur la diagonale principale, et sur le première et deuxième super-diagonale, c'est-à-dire, $r_{j,k} = 0$ pour $j > k$ et pour $k > j+2$. Un examen plus fin permet de réduire la complexité à $O(n)$ OA pour obtenir une décomposition QR de A .

Les rotations de Givens

Toute variante de construction de matrices orthogonales annulant une partie d'un vecteur (voir avant le théorème 8.1) donnera lieu à une variante de l'algorithme de Householder.

Exemple 8.11 (Une rotation dans \mathbb{R}^2)

Pour tout $y = [y_1, y_2]^T \in \mathbb{R}$ il existe un angle φ de sorte que

$$Gy = \begin{bmatrix} \|y\|_2 \\ 0 \end{bmatrix}, \quad G = G(\varphi) = \begin{bmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{bmatrix}.$$

Définition 8.12 (définition d'une rotation de Givens)

Étant donné $1 \leq i < j \leq m$, une **rotation de Givens** $G^{(i,j)} = G^{(i,j)}(\varphi) \in \mathcal{M}_m(\mathbb{R})$ est obtenue en partant de l'identité I_m , où on remplace la sous-matrice à indices lignes/colonnes i et j par une rotation $G(\varphi)$ comme avant.

Remarques 8.13

Notons que tout produit de rotations de Givens est une matrice orthogonale, et que la multiplication $G^{(i,j)}B$ avec $B \in \mathcal{M}_{m,n}(\mathbb{R})$ est facile à implémenter (en $6n + \mathcal{O}(1)$ OE) car seulement les lignes i et j de B changent.

Créer des zéros avec les rotations de Givens

Etant donné $y \in \mathbb{R}^m$, on peut trouver des angles de sorte que

$$G^{(1,2)} \dots G^{(m-2,m-1)} G^{(m-1,m)} y = \|y\| e_1,$$

le facteur $G^{(m-1,m)}$ créant un zéro à la dernière position, le facteur $G^{(m-2,m-1)}$ créant un zéro à l'avant-dernière position (et laisse invariant le zéro à la dernière position) etc.

D'autres ordres de rotations de Givens sont imaginables, surtout si y comporte déjà beaucoup de zéros. Par exemple, si $y_3 = \dots = y_m = 0$ (voir le cas Hessenberg discuté avant), alors un produit $G^{(1,2)}y = \|y\|e_1$ suffit.

L'algorithme décrit dans le théorème 8.1 pour trouver une décomposition QR avec $Q = (Q^{(n)})^T$ et $R = A^{(n)}$ aura alors la forme suivante : on initialise $Q^{(1)} = I_m$, $A^{(1)} = A$, et on calcule pour $k = 1, \dots, n-1$

$$A^{(k+1)} = G^{(k,k+1)}(\varphi^{(k,k+1)}) \dots G^{(m-1,m)}(\varphi^{(m-1,m)}) A^{(k)},$$
$$Q^{(k+1)} = G^{(k,k+1)}(\varphi^{(k,k+1)}) \dots G^{(m-1,m)}(\varphi^{(m-1,m)}) Q^{(k)},$$

avec $\varphi^{(i-1,i)}$ pour $i = k+1, \dots, m$ choisi pour produire un zéro à la position (i, k) . D'une manière similaire on montre :

Corollaire 8.14

Pour tout $A \in \mathcal{M}_n(\mathbb{R})$ il existe une matrice $Q \in \mathcal{M}_n(\mathbb{R})$ orthogonale (produit de $(n-1)(n-2)/2$ rotations de Givens) de sorte que $Q^T A Q$ soit une matrice de Hessenberg, en complexité $O(n^3)$.

Ce résultat est à comparer avec la décomposition de Schur qui est plus coûteuse car elle nécessite le calcul d'éléments propres.

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algorithme de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

Éléments propres et leur utilité

Étant donné $A \in \mathcal{M}_n(\mathbb{C})$ un couple $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n$ est dit **élément propre** (à droite) de A si $v \neq 0$ et $Av = \lambda v$ (et élément propre à gauche si $(\bar{\lambda}, v)$ est l'élément propre de A^*).

Exemple 9.1

En mécanique, si $A = A^$ est la matrice de rigidité d'un système masses-ressorts, les valeurs propres nous donnent des fréquences de résonance de ce système.*

De même, pour une matrice A d'inertie, les vecteurs propres nous donnent les axes de symétrie d'un solide (par exemple l'axe de rotation d'une toupie).

Éléments propres et leur utilité

Étant donné $A \in \mathcal{M}_n(\mathbb{C})$ un couple $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n$ est dit **élément propre** (à droite) de A si $v \neq 0$ et $Av = \lambda v$ (et élément propre à gauche si $(\bar{\lambda}, v)$ est l'élément propre de A^*).

Exemple 9.1

En mécanique, si $A = A^$ est la matrice de rigidité d'un système masses-ressorts, les valeurs propres nous donnent des fréquences de résonance de ce système.*

De même, pour une matrice A d'inertie, les vecteurs propres nous donnent les axes de symétrie d'un solide (par exemple l'axe de rotation d'une toupie).

Exemple 9.2

Pour calculer la matrice la plus proche (en norme spectrale $\|\cdot\|_2$) de rang au plus k d'une matrice donnée A , selon le théorème de Eckhart-Young on a besoin des k éléments propres de $A^T A$ associées aux k plus grandes valeurs propres de $A^T A$.

Éléments propres et leur utilité

Étant donné $A \in \mathcal{M}_n(\mathbb{C})$ un couple $(\lambda, v) \in \mathbb{C} \times \mathbb{C}^n$ est dit **élément propre** (à droite) de A si $v \neq 0$ et $Av = \lambda v$ (et élément propre à gauche si $(\bar{\lambda}, v)$ est l'élément propre de A^*).

Exemple 9.1

En mécanique, si $A = A^*$ est la matrice de rigidité d'un système masses-ressorts, les valeurs propres nous donnent des fréquences de résonance de ce système.

De même, pour une matrice A d'inertie, les vecteurs propres nous donnent les axes de symétrie d'un solide (par exemple l'axe de rotation d'une toupie).

Exemple 9.2

Pour calculer la matrice la plus proche (en norme spectrale $\|\cdot\|_2$) de rang au plus k d'une matrice donnée A , selon le théorème de Eckhart-Young on a besoin des k éléments propres de $A^T A$ associées aux k plus grandes valeurs propres de $A^T A$.

Exemple 9.3

Selon Google PageRank, un score $u_j \in [0, +\infty)$ associé à une page web j est donné par la règle suivante : une page est importante si la moyenne des scores des pages contenant un lien vers j est importante (on écarte des pages sans lien sortant), autrement dit

$$Au = u, \quad u \neq 0, \quad a_{j,k} = \frac{\text{nombre de liens de } k \text{ vers } j}{\text{nombre de liens sortants de la page } k}.$$

Un tel u existe ? Oui, car $A^T e = e$, $e = (1, 1, \dots, 1)^T$.

Est-il unique (à normalisation $e^T u = 1$ près) ? Est-il positif ? Plus compliqué, voir le théorème de Perron-Frobenius.

NB : $1 \in \text{Sp}(A)$ et alors $1 = |1| \leq \rho(A) \leq \|A^T\|_\infty = 1$.

Stabilité des valeurs propres sous perturbations (1)

Exemple 9.4 (Un bloc de Jordan perturbé)

Pour $\epsilon > 0$ soit $A(\epsilon) :=$

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ \epsilon & 0 & \cdots & \cdots & 0 \end{bmatrix} \in \mathcal{M}_n(\mathbb{R}),$$

alors pour $n = 10$, $\epsilon = 10^{-10}$

$$\|A(\epsilon) - A(0)\|_2 = \|\epsilon e_n e_1^*\|_2 = \epsilon$$

Stabilité des valeurs propres sous perturbations (1)

Exemple 9.4 (Un bloc de Jordan perturbé)

Pour $\epsilon > 0$ soit $A(\epsilon) :=$

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ \epsilon & 0 & \cdots & \cdots & 0 \end{bmatrix} \in \mathcal{M}_n(\mathbb{R}),$$

alors pour $n = 10$, $\epsilon = 10^{-10}$

$$\|A(\epsilon) - A(0)\|_2 = \|\epsilon e_n e_1^*\|_2 = \epsilon$$

$$Sp(A(0)) = \{0\}, \quad Sp(A(\epsilon)) = \left\{ \frac{\exp(\pi i j/5)}{10} : j = 1, \dots, n \right\}$$

donc les matrices sont proches mais pas leurs valeurs propres.

Stabilité des valeurs propres sous perturbations (1)

Exemple 9.4 (Un bloc de Jordan perturbé)

Pour $\epsilon > 0$ soit $A(\epsilon) :=$

$$\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ \epsilon & 0 & \cdots & \cdots & 0 \end{bmatrix} \in \mathcal{M}_n(\mathbb{R}),$$

alors pour $n = 10$, $\epsilon = 10^{-10}$

$$\|A(\epsilon) - A(0)\|_2 = \|\epsilon e_n e_1^*\|_2 = \epsilon$$

$$Sp(A(0)) = \{0\}, \quad Sp(A(\epsilon)) = \left\{ \frac{\exp(\pi ij/5)}{10} : j = 1, \dots, n \right\}$$

donc les matrices sont proches mais pas leurs valeurs propres.

Théorème 9.5 (de Bauer-Fike)

Soient $A, B, V \in \mathcal{M}_n(\mathbb{C})$ avec $D = V^{-1}AV$ diagonal, alors

$$\forall \mu \in Sp(B) \exists \lambda \in Sp(A) \text{ t.q. } |\lambda - \mu| \leq \text{cond}_2(V) \|B - A\|_2.$$

Corollaire 9.6

Soient $A, B \in \mathcal{M}_n(\mathbb{C})$ avec A normale, alors

$$\forall \mu \in Sp(B) \ \exists \lambda \in Sp(A) \text{ t.q. } |\lambda - \mu| \leq \|B - A\|_2.$$

Pour certaines matrices A nous pouvons affirmer qu'une petite perturbation donne une matrice avec valeurs propres proches de ceux de A . D'après l'exemple 9.4, ceci peut être faux dans le cas général.

Stabilité des valeurs propres sous perturbations (2)

Corollaire 9.6

Soient $A, B \in \mathcal{M}_n(\mathbb{C})$ avec A normale, alors

$$\forall \mu \in Sp(B) \exists \lambda \in Sp(A) \text{ t.q. } |\lambda - \mu| \leq \|B - A\|_2.$$

Pour certaines matrices A nous pouvons affirmer qu'une petite perturbation donne une matrice avec valeurs propres proches de ceux de A . D'après l'exemple 9.4, ceci peut être faux dans le cas général.

Un $y \in \mathbb{C}^n \setminus \{0\}$ donné, est-il proche d'un vecteur propre de A ?

Théorème 9.7 (résidu)

Pour $\lambda \in \mathbb{C}$, $y \in \mathbb{C}^n \setminus \{0\}$ on définit le **résidu** $r(\lambda, y) = \lambda y - Ay$, alors

$$\arg \min \{ \|r(\lambda, y)\|_2 : \lambda \in \mathbb{C} \} = \frac{y^* A y}{y^* y} =: R_A(y) \quad \text{dit} \textbf{quotient de Rayleigh}.$$

Stabilité des valeurs propres sous perturbations (2)

Corollaire 9.6

Soient $A, B \in \mathcal{M}_n(\mathbb{C})$ avec A normale, alors

$$\forall \mu \in \text{Sp}(B) \exists \lambda \in \text{Sp}(A) \text{ t.q. } |\lambda - \mu| \leq \|B - A\|_2.$$

Pour certaines matrices A nous pouvons affirmer qu'une petite perturbation donne une matrice avec valeurs propres proches de ceux de A . D'après l'exemple 9.4, ceci peut être faux dans le cas général.

Un $y \in \mathbb{C}^n \setminus \{0\}$ donné, est-il proche d'un vecteur propre de A ?

Théorème 9.7 (résidu)

Pour $\lambda \in \mathbb{C}$, $y \in \mathbb{C}^n \setminus \{0\}$ on définit le **résidu** $r(\lambda, y) = \lambda y - Ay$, alors

$$\arg \min \{ \|r(\lambda, y)\|_2 : \lambda \in \mathbb{C} \} = \frac{y^* A y}{y^* y} =: R_A(y) \quad \text{dit quotient de Rayleigh.}$$

Aussi, si $V \in \mathcal{M}_n(\mathbb{C})$ avec $D = V^{-1} A V$ diagonal, alors

$$\text{dist}(\lambda, \text{Sp}(A)) \leq \text{cond}_2(V) \frac{\|r(\lambda, y)\|_2}{\|y\|_2}.$$

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algo de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

La méthode de la puissance (classique)

Cette méthode consiste, à partir de $x_0 \in \mathbb{C}^n$, de calculer $x_{k+1} = Ax_k$ pour $k = 0, 1, \dots$, et alors $x_k = A^k x_0$.

NB : dans la suite de ce cours, x_k ne désigne pas la k ième composante d'un vecteur mais le k ième vecteur d'une suite.

Théorème 10.1 (Convergence de la méthode de la puissance)

Soit $A \in \mathcal{M}_n(\mathbb{C})$ diagonalisable, avec valeurs propres λ_j vérifiant

$$|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|.$$

Notons par v_1 un vecteur propre à droite et par u_1 un vecteur propre à gauche de A associé à la valeur propre λ_1 . On choisit $x_0, y \in \mathbb{C}^n$ de sorte que $u_1^* x_0 \neq 0$ et $y^* v_1 \neq 0$. Alors

$$\frac{y^* x_{k+1}}{y^* x_k} = \lambda_1 + \mathcal{O}\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right)_{k \rightarrow \infty}, \quad \frac{x_k}{y^* x_k} = \frac{v_1}{y^* v_1} + \mathcal{O}\left(\left(\frac{\lambda_2}{\lambda_1}\right)^k\right)_{k \rightarrow \infty}.$$

Si de plus A est une matrice normale, alors

$$R_A(x_k) = \lambda_1 + \mathcal{O}\left(\left|\frac{\lambda_2}{\lambda_1}\right|^{2k}\right)_{k \rightarrow \infty}.$$

Retour à l'exemple 9.3 de Google PageRank

Rappel : on avait introduit une matrice $A \in \mathcal{M}_n(\mathbb{R})$ avec $A^T e = e$, $e = (1, 1, \dots, 1)^T$, et les scores à travers de l'élément propre $(1, v_1)$ de A .

Google introduit un “vecteur marketing” $w \in]0, 1[^n$ avec $e^T w = 1$, et

$$B = tA + (1 - t)we^T, \quad \text{avec } t = 0.7, \quad \text{et} \quad e^T B = e^T,$$

Le théorème Perron-Frobenius implique :

- B admet la valeur propre $\lambda_1 = \rho(B) = 1$, de multiplicité algébrique = 1 ;
- le vecteur propre v_1 associé à λ_1 est à composantes > 0 (les scores) ;
- Toute autre valeur propre λ_j de B pour $j = 2, \dots, n$ est de module $< \rho(B) = 1$.
- Plus précisément, avec (λ_j, v_j) pour $j \geq 2$ élément propre de B ,

$$e^T v_j = 0 \quad \implies \quad \lambda_j v_j = B v_j = t A v_j \quad \implies \quad |\lambda_j| \leq t \rho(A) = t,$$

et donc (éventuellement après permutation)

$$|\lambda_1| = 1 > t = 0.7 \geq |\lambda_2| \geq \dots \geq |\lambda_n|.$$

En prenant $x_0, y \in [0, +\infty[^n \setminus \{0\}$, alors l'ensemble des hypothèses du théorème 10.1 sont valables. On pourrait donc appliquer la méthode de la puissance sous la forme $x_{k+1} = tAx_k + y(e^T x_k)$ pour trouver le PageRank, avec un taux de convergence $|\lambda_2/\lambda_1| \leq 0.7$ intéressant.

Variations de la méthode de la puissance (1)

D'après la preuve du théorème 10.1, $y^* x_k$ se comporte comme une constante fois λ_1^k , et on a intérêt de normaliser $q_k = x_k / \|x_k\|$.

Algorithme 10.2 (la méthode de la puissance normalisée)

Choisir $z_0 \in \mathbb{C}^n \setminus \{0\}$, et pour $k = 0, 1, 2, \dots$

$$q_k = z_k / \|z_k\|, \quad z_{k+1} = Aq_k.$$

Sortie : $\frac{y^* z_{k+1}}{y^* q_k} = \frac{y^* x_{k+1}}{y^* x_k} \rightarrow \lambda_1$, $\frac{q_k}{y^* q_k} = \frac{x_k}{y^* x_k} \rightarrow \frac{v_1}{y^* v_1}$, $q_k^* z_{k+1} = R_A(x_k) \rightarrow \lambda_1$.

Complexité par itération : au pire $2n^2 + \mathcal{O}(n)$.

Parfois on dispose d'un $\mu \in \mathbb{R}$ (beaucoup) plus proche de la valeur propre λ_ℓ qu'aux autres valeurs propres λ_j pour $j \neq \ell$. Ici on peut appliquer la méthode de la puissance pour la matrice $(A - \mu I)^{-1}$.

Algorithme 10.3 (la méthode de la puissance inverse à paramètre fixe)

Choisir $z_0 \in \mathbb{C}^n \setminus \{0\}$, et pour $k = 0, 1, 2, \dots$

$$q_k = z_k / \|z_k\|, \quad \text{résoudre pour } z_{k+1} \text{ le système } (A - \mu I)z_{k+1} = q_k.$$

Sortie : $\frac{y^* z_{k+1}}{y^* q_k} \rightarrow \frac{1}{\lambda_\ell - \mu}$, $\frac{q_k}{y^* q_k} \rightarrow \frac{v_\ell}{y^* v_\ell}$, $q_k^* z_{k+1} = R_{(A - \mu I)^{-1}}(x_k) \rightarrow \frac{1}{\lambda_\ell - \mu}$.

Complexité par itération : $2n^2 + \mathcal{O}(n)$ (avec une seule décomposition $A - \mu I = LU$).

La méthode de la puissance inverse

En pratique on utilise (à partir d'un certain rang) un paramètre μ_k variable.

Algorithme 10.4 (la méthode de la puissance inverse à paramètre variable)

Choisir $z_0 \in \mathbb{C}^n \setminus \{0\}$, et pour $k = 0, 1, 2, \dots$ jusqu'à $\|z_k\|_2$ "grand"

$$q_k = z_k / \|z_k\|, \quad \mu_k = R_A(q_k), \quad \text{résoudre pour } z_{k+1} \text{ le système } (A - \mu_k I)z_{k+1} = q_k.$$

L'étude (locale) de convergence de ce dernier algorithme dépasse le cadre du cours, ici juste quelques remarques (sans preuves). On suppose que μ_0 est "proche" de λ_ℓ .

Remarques 10.5

- Ici on calcule à chaque itération une nouvelle décomposition LU. On peut réduire la complexité en passant d'abord de A à une matrice semblable A_0 de forme Hessenberg, voire le corollaire 8.14.
- La condition d'arrêt est motivée par le théorème 9.7

$$\text{dist}(\mu_k, \text{Sp}(A)) \leq \text{cond}_2(V) \frac{\|(A - \mu_k I)z_{k+1}\|_2}{\|z_{k+1}\|_2} = \frac{\text{cond}_2(V)}{\|z_{k+1}\|_2}.$$

- La quantité $\frac{q_k}{y^* q_k}$ approche bien $\frac{v_\ell}{y^* v_\ell}$ même si $A - \mu_k I$ est mal conditionné.
- Si A est normale : $\text{dist}(\mu_{k+1}, \text{Sp}(A)) = \mathcal{O}(\text{dist}(\mu_k, \text{Sp}(A))^3)$.
- Dans le cas général : $\text{dist}(\mu_{k+1}, \text{Sp}(A)) = \mathcal{O}(\text{dist}(\mu_k, \text{Sp}(A))^2)$.

- 1 Notions et Rappels
- 2 La factorisation de Schur et la SVD
- 3 Normes matricielles et conditionnement
- 4 Résolution de systèmes linéaires $Ax = b$ par l'élimination de Gauss et décomposition LU
- 5 L'algorithme de Gauss sur ordinateur
- 6 Comment exploiter une structure dans la matrice A ?
- 7 Le problème des moindres carrés et la décomposition QR
- 8 Calcul de la décomposition QR pleine par Householder et Givens
- 9 Calcul numérique de valeurs propres
- 10 La méthode de la puissance
- 11 La méthode QR

L'algorithme QR pour les matrices réelles

Comment calculer le spectre entier d'une matrice réelle $A \in \mathcal{M}_n(\mathbb{R})$?

Algorithme 11.1 (la méthode QR avec shift)

On initialise $A_0 = Q_0^T A Q_0$ avec Q_0 orthogonale, A_0 de forme Hessenberg.

Pour $k = 0, 1, 2, \dots$

On se donne un shift $\mu_k \in \mathbb{R}$

Calculer décomposition QR de $A_k - \mu_k I = Q_{k+1} R_{k+1}$.

Calculer $A_{k+1} = \mu_k I + R_{k+1} Q_{k+1}$.

NB : $A_{k+1} = Q_{k+1}^T A_k Q_{k+1}$, donc A et A_k sont semblables.

L'algorithme QR pour les matrices réelles

Comment calculer le spectre entier d'une matrice réelle $A \in \mathcal{M}_n(\mathbb{R})$?

Algorithme 11.1 (la méthode QR avec shift)

On initialise $A_0 = Q_0^T A Q_0$ avec Q_0 orthogonale, A_0 de forme Hessenberg.

Pour $k = 0, 1, 2, \dots$

On se donne un shift $\mu_k \in \mathbb{R}$

Calculer décomposition QR de $A_k - \mu_k I = Q_{k+1} R_{k+1}$.

Calculer $A_{k+1} = \mu_k I + R_{k+1} Q_{k+1}$.

NB : $A_{k+1} = Q_{k+1}^T A_k Q_{k+1}$, donc A et A_k sont semblables.

Lemme 11.2 (de préservation de forme)

- Q_{k+1} et A_{k+1} sont de forme Hessenberg, et plus précisément Q_{k+1} peut s'écrire comme un produit de $n - 1$ rotations de Givens.
- Si de plus A est symétrique, alors toute matrice A_k est symétrique et tridiagonale, et R_{k+1} ne contient que des éléments non-nuls sur la diagonale principale, ainsi que sur la première et deuxième super-diagonale.

L'algorithme QR pour les matrices réelles

Comment calculer le spectre entier d'une matrice réelle $A \in \mathcal{M}_n(\mathbb{R})$?

Algorithme 11.1 (la méthode QR avec shift)

On initialise $A_0 = Q_0^T A Q_0$ avec Q_0 orthogonale, A_0 de forme Hessenberg.

Pour $k = 0, 1, 2, \dots$

On se donne un shift $\mu_k \in \mathbb{R}$

Calculer décomposition QR de $A_k - \mu_k I = Q_{k+1} R_{k+1}$.

Calculer $A_{k+1} = \mu_k I + R_{k+1} Q_{k+1}$.

NB : $A_{k+1} = Q_{k+1}^T A_k Q_{k+1}$, donc A et A_k sont semblables.

Lemme 11.2 (de préservation de forme)

- Q_{k+1} et A_{k+1} sont de forme Hessenberg, et plus précisément Q_{k+1} peut s'écrire comme un produit de $n - 1$ rotations de Givens.
- Si de plus A est symétrique, alors toute matrice A_k est symétrique et tridiagonale, et R_{k+1} ne contient que des éléments non-nuls sur la diagonale principale, ainsi que sur la première et deuxième super-diagonale.

Corollaire 11.3 (complexité de la méthode QR)

L'algorithme QR nécessite $\mathcal{O}(n^3)$ OA pour l'initialisation, ainsi que $\mathcal{O}(n^2)$ par itération.

Si A est symétrique et creuse, la complexité peut être réduite à $\mathcal{O}(n^2)$ OA pour l'initialisation, ainsi que $\mathcal{O}(n)$ par itération.

Convergence de l'algorithme QR avec shift $\mu_k = 0$

Lemme 11.4

Posons $W_k = Q_0 Q_1 \dots Q_k$ matrice orthogonale, alors

$$AW_k = W_k A_k, \quad (A - \mu_k I)W_k = W_{k+1} R_{k+1}, \quad (A^T - \mu_k I)^{-1} W_k = W_{k+1} R_{k+1}^{-T}.$$

Convergence de l'algorithme QR avec shift $\mu_k = 0$

Lemme 11.4

Posons $W_k = Q_0 Q_1 \dots Q_k$ matrice orthogonale, alors

$$AW_k = W_k A_k, \quad (A - \mu_k I)W_k = W_{k+1} R_{k+1}, \quad (A^T - \mu_k I)^{-1} W_k = W_{k+1} R_{k+1}^{-T}.$$

Remarques 11.5

Considérons le cas $\mu_k = 0$ pour tout k , et supposons que $|\lambda_1| > \dots > |\lambda_n|$ pour les valeurs propres de A , plus conditions techniques sur les vecteurs propres. Alors

$$\forall j : \lim_{k \rightarrow \infty} (A_k)_{j,j} = \lambda_j, \quad \forall j < \ell : \quad (A_k)_{\ell,j} = \mathcal{O}(|\lambda_\ell / \lambda_j|^k)_{k \rightarrow \infty}$$

ce qui généralise la méthode de la puissance normalisée

$$AW_k e_1 = W_{k+1} (R_{k+1})_{1,1} e_1 \implies W_{k+1} e_1 = \frac{AW_k e_1}{\|AW_k e_1\|_2}.$$

Plus généralement, avec $\Delta_k = \text{diag}(\pm 1, \dots, \pm 1)$ approprié

$$\lim_{k \rightarrow \infty} W_k \Delta_k = W, \quad \lim_{k \rightarrow \infty} (W_k \Delta_k)^T A (W_k \Delta_k) = \lim_{k \rightarrow \infty} \Delta_k^T A_k \Delta_k = R$$

avec une factorisation de Schur $W^T A W = R$ triangulaire supérieure.

Exemples numériques méthode QR avec shift $\mu_k = 0$

Une matrice non symétrique d'ordre 4, 39 itérations

Une matrice symétrique d'ordre 4, 39 itérations

Convergence de l'algorithme QR avec shift de Rayleigh $\mu_k = (A_k)_{n,n}$

On souhaite accélérer la convergence en remplaçant la méthode de la puissance pour $W_k e_1$ et la matrice A par

$$(A^T - \mu_k I)^{-1} W_k e_n = W_{k+1} (R_{k+1}^{-T})_{n,n} e_n \implies W_{k+1} e_n = \frac{(A^T - \mu_k I)^{-1} W_k e_n}{\|(A^T - \mu_k I)^{-1} W_k e_n\|_2},$$

la méthode de la puissance inverse pour $W_k e_n$ et la matrice A^T qui localement converge bien plus rapidement pour le shift de Rayleigh $\mu_k = (A_k)_{n,n} = R_A(W_k e_n)$.

Convergence de l'algorithme QR avec shift de Rayleigh $\mu_k = (A_k)_{n,n}$

On souhaite accélérer la convergence en remplaçant la méthode de la puissance pour $W_k e_1$ et la matrice A par

$$(A^T - \mu_k I)^{-1} W_k e_n = W_{k+1} (R_{k+1}^{-T})_{n,n} e_n \implies W_{k+1} e_n = \frac{(A^T - \mu_k I)^{-1} W_k e_n}{\|(A^T - \mu_k I)^{-1} W_k e_n\|_2},$$

la méthode de la puissance inverse pour $W_k e_n$ et la matrice A^T qui localement converge bien plus rapidement pour le shift de Rayleigh $\mu_k = (A_k)_{n,n} = R_A(W_k e_n)$.

Remarques 11.6 (sur la déflation)

Par construction, la norme du résidu

$$\|A^T W_k e_n - \mu_k W_k e_n\| = \|(e_n^T W_k^T A - \mu_k e_n^T W_k^T) W_k\|_2 = \|A_k[n, 1:n-1]\|_2 = |(A_k)_{n,n-1}|$$

est petit ssi A_k est presque block-diagonal, autrement dit, $Sp(A_k)$ coïncide presque avec l'union entre $\{\mu_k\}$ et le spectre de $A_k[1:n-1, 1:n-1]$, la sous-matrice principale de A_k d'ordre $n-1$.

*Dès que l'on rencontre un résidu $|(A_k)_{n,n-1}|$ petit, on appliquera un processus dit **de déflation**, c'est-à-dire, on stockera μ_k comme une bonne approximation d'une valeur propre de A , et on supprimera dans A_k la dernière ligne et colonne (et dans W_k la dernière colonne), en diminuant n par 1.*

L'algorithme QR avec shift de Rayleigh et déflation.

Le lecteur intéressé pourra montrer que, essentiellement, $n - m$ déflations reviennent à continuer à travailler avec des matrices A_k d'ordre n , mais on replace $|(A_k)_{j,j-1}|$ par zéro pour $j = m + 1, m + 2, \dots, n$. Il en suit que Q_{k+1} est bloc diagonal, avec un premier bloc orthogonal d'ordre m , et un deuxième bloc égal à l'identité d'ordre $n - m$, et seulement la sous-matrice principale d'ordre m change en passant de A_k à A_{k+1} . On s'arrête quand $m = 1$.

L'algorithme QR avec shift de Rayleigh et déflation.

Le lecteur intéressé pourra montrer que, essentiellement, $n - m$ déflations reviennent à continuer à travailler avec des matrices A_k d'ordre n , mais on replace $|(A_k)_{j,j-1}|$ par zéro pour $j = m + 1, m + 2, \dots, n$. Il en suit que Q_{k+1} est bloc diagonal, avec un premier bloc orthogonal d'ordre m , et un deuxième bloc égal à l'identité d'ordre $n - m$, et seulement la sous-matrice principale d'ordre m change en passant de A_k à A_{k+1} . On s'arrête quand $m = 1$.

Algorithme 11.7 (la méthode QR avec shift de Rayleigh et déflation)

On initialise $A \leftarrow Q^T A Q$ de forme Hessenberg, Q orthogonale.

On pose $m = n$.

Pour $k = 0, 1, 2, \dots$

Si $|(A)_{m,m-1}|$ petit alors $(A)_{m,m-1} \leftarrow 0$, $m \leftarrow m - 1$.

Arrêt si $m = 1$.

Calculer shift $\mu_k = (A)_{m,m}$.

Calculer décomposition QR de $A - \mu_k I = QR$.

Calculer $A \leftarrow \mu_k I + RQ$

Complexité : on s'attend à 3-4 itération pour chaque valeur de m .

Exemples numériques méthode QR avec shift μ_k de Rayleigh

Une matrice non symétrique d'ordre 4

Une matrice symétrique d'ordre 4

Une matrice symétrique d'ordre 20

Remarques 11.8

Dans notre algorithme QR on ne stocke ni les matrices Q_k ni les matrices W_k , c'est-à-dire, en sortant de l'algorithme avec $m = 1$, on obtient sur la diagonale de A_k des bonnes approximations $\underline{\lambda}_j$ des valeurs propres λ_j de A . Pour calculer des bonnes approximations des vecteurs propres associés, on lancera pour chaque $j = 1, \dots, n$ la méthode de la puissance inverse à paramètre initial $\mu_0 = \underline{\lambda}_j$.

Remarques 11.8

Dans notre algorithme QR on ne stocke ni les matrices Q_k ni les matrices W_k , c'est-à-dire, en sortant de l'algorithme avec $m = 1$, on obtient sur la diagonale de A_k des bonnes approximations $\underline{\lambda}_j$ des valeurs propres λ_j de A . Pour calculer des bonnes approximations des vecteurs propres associés, on lancera pour chaque $j = 1, \dots, n$ la méthode de la puissance inverse à paramètre initial $\mu_0 = \underline{\lambda}_j$.

Remarques 11.9

*Si on veut seulement obtenir ℓ éléments propres de A , il existe une variante dite **méthode des sous-espaces itérés** où les matrices W_k sont à colonnes orthonormées, mais de taille $n \times \ell$, et les matrices A_k de taille $\ell \times \ell$. L'algorithme et l'étude de convergence ressemble à celui de la méthode QR.*

Pour cibler certaines valeurs propres (par exemple les plus grands en module), on peut faire appel à des méthodes de sous-espaces de Krylov comme ARpack, qui sont encore sujet de recherche actuelle.