

Calcul Scientifique

Recherche de couples propres

2019–2020

Recherche de valeurs propres et vecteurs propres

Définition

$\lambda \in \mathbb{C}$ valeur propre de A : $\exists x \in \mathbb{C}^n$, $x \neq 0$, $A \cdot x = \lambda x$

x est appelé vecteur propre associé à λ .

Pourquoi chercher des valeurs propres (et des vecteurs propres) ?

- Résultats ayant une signification physique : mode de vibration d'une structure, traitement du signal, ...
- Éléments de réponse pour vérifier une propriété numérique : conditionnement d'une matrice, convergence de méthodes itératives, ...

Plusieurs types de problèmes

Pourquoi chercher des valeurs propres (et des vecteurs propres) ?

- Rechercher toutes les valeurs propres
Exemple : les valeurs propres ont une signification physique
- Vérifier que les valeurs propres obéissent à une certaine propriété (le calcul exact n'est pas requis)
Exemple : toutes les valeurs propres en module sont inférieures à 1
- Calculer la (les) plus grande(s) des valeurs propres en module et/ou la (les) plus petite(s), ainsi qu'un vecteur propre associé
Exemple : calcul du nombre de conditionnement, algorithmes de classement de pages Web.

- ① Localisation des valeurs propres
- ② Algorithme de la puissance itérée / Cas d'une matrice symétrique \implies les valeurs propres sont obtenues successivement dans l'ordre décroissant de la valeur de leur module.
- ③ Algorithme de Jacobi / Cas d'une matrice symétrique \implies toutes les valeurs propres sont obtenues simultanément.

Localisation des valeurs propres

Théorème d'Hadamard-Gerchgorin

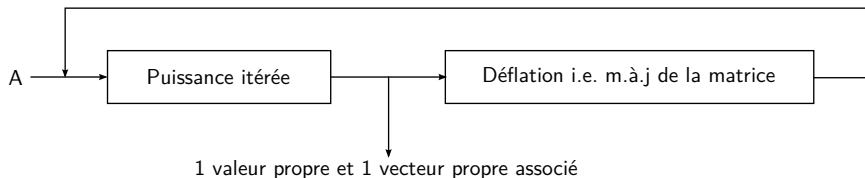
Soit $A \in \mathcal{M}_n(\mathbb{C})$, les valeurs propres de A ont des images dans le plan complexe qui appartiennent à $\bigcup_{i=1}^n D_i$ avec : $D_i = \left\{ z \in \mathbb{C} / |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}| \right\}$

Remarque : si $A \in \mathcal{M}_n(\mathbb{R})$ et si toutes les valeurs propres sont réelles, alors D_i et $\bigcup_{i=1}^n D_i$ sont des intervalles de \mathbb{R}

Corollaire au théorème d'Hadamard-Gerchgorin

$$\rho(A) \leq \max_{i=1, \dots, n} \sum_{j=1}^n |a_{ij}| \qquad \rho(A) \leq \max_{j=1, \dots, n} \sum_{i=1}^n |a_{ij}|$$

Algorithme de la puissance itérée



Principe

Hypothèse : toutes les valeurs propres sont réelles, non nulles et distinctes en module. Soient $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ les valeurs propres de A

- 1ère application de l'algorithme $\implies \lambda_1$ et un vecteur propre associé
- Opération de déflation : modification de la matrice
- 2ème application de l'algorithme $\implies \lambda_2$ et un vecteur propre associé
- En n passages, toutes les valeurs propres et une base de vecteurs propres associés

Autres conditions d'application

- Si une valeur propre est complexe, l'algorithme échoue
- Si un sous-espace propre est de dimension supérieure à 1, obtention plusieurs fois de la même valeur propre, les vecteurs propres forment une base du sous-espace propre
- Procédé itératif convergent (suite de vecteurs) avec mise en œuvre d'un test d'arrêt et convergence pas toujours assurée quelque soit le vecteur de départ

Algorithm 1 Méthode de la puissance itérée (Power method)

Input : Matrice $A \in \mathcal{M}_n(\mathbb{R})$

Output : (λ_1, v_1) couple propre associé à la plus grande (en module) valeur propre.

$x_0 \in \mathbb{R}^n$ donné et $p = 0$

$$\beta_p = x_p^T \cdot A \cdot x_p$$

repeat

$$y_{p+1} = A \cdot x_p$$

$$x_{p+1} = y_{p+1} / \|y_{p+1}\|$$

$$\beta_{p+1} = x_{p+1}^T \cdot A \cdot x_{p+1}$$

$$p = p + 1$$

until $|\beta_{p+1} - \beta_p| / |\beta_p| < \varepsilon$

$$\lambda_1 = \beta_{p+1} \text{ et } v_1 = x_{p+1}$$

Convergence de la suite : $X_{p+1} = A \cdot X_p / \|A \cdot X_p\|$

Elements de preuve

$$\begin{cases} X_1 = \frac{1}{\|A \cdot X_0\|} A \cdot \left(\sum_{i=1}^n c_i V_i \right); \alpha_1 = \frac{1}{\|A \cdot X_0\|}; X_1 = \alpha_1 \sum_{i=1}^n c_i \lambda_i V_i \\ X_p = \alpha_p \sum_{i=1}^n c_i \lambda_i^p V_i; X_p = \alpha_p \lambda_1^p \left(c_1 V_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^p V_2 + \dots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^p V_n \right) \end{cases}$$

$$\left. \begin{aligned} \lim_{p \rightarrow +\infty} \|X_{2p}\| &= \lim_{p \rightarrow +\infty} \alpha_{2p} \lambda_1^{2p} \|c_1 V_1\| = 1 \\ \lim_{p \rightarrow +\infty} \alpha_{2p} \lambda_1^{2p} &= \frac{1}{\|c_1 V_1\|} \end{aligned} \right\} \Rightarrow \begin{aligned} \lim_{p \rightarrow +\infty} X_{2p} &= \frac{c_1 V_1}{\|c_1 V_1\|} = W_1 \\ \text{où } W_1 &\text{ est un vecteur propre} \\ &\text{normé associé à } \lambda_1 \end{aligned}$$

Méthode applicable si $c_1 \neq 0$ (faible taux d'échec avec choix arbitraire de X_0)

$$\begin{cases} \lim_{p \rightarrow +\infty} X_{2p+1} &= W_1 \text{ ou } -W_1 \text{ suivant le signe de } \lambda_1 \\ \lim_{p \rightarrow +\infty} \beta_p &= W_1^T \cdot A \cdot W_1 = \lambda_1 \end{cases}$$

Opération de déflation (Cas d'une matrice symétrique)

Principes

Soit $B = A - \lambda_1 W_1 \cdot W_1^T$

- Rang de $B = n - 1$ ($B \cdot W_1 = 0$)
- B est symétrique
- B possède les mêmes valeurs propres $\lambda_2, \dots, \lambda_n$ que A et les mêmes vecteurs propres associés

L'application de l'algorithme de la puissance itérée à B produit λ_2 et W_2

$$C = B - \lambda_2 W_2 \cdot W_2^T \longrightarrow \lambda_3, W_3 \longrightarrow \dots \longrightarrow \lambda_n, W_n$$

Exercice

- 1 Soit $A \in \mathcal{M}_n(\mathbb{R})$, λ valeur propre de A et u vecteur propre associé. Soit $\alpha \in \mathbb{R}$ non valeur propre de A . Montrer que $\mu = \frac{1}{\lambda - \alpha}$ est valeur propre de $(A - \alpha I)^{-1}$ et que, pour cette matrice, u est vecteur propre associé à μ .
- 2 Soit A symétrique et inversible. Soient $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ les valeurs propres de A . On note $\|\cdot\|$ la norme vectorielle euclidienne.

Quelles valeurs l'algorithme suivant permet-il d'obtenir ?

$i = 0, x_i = \dots$

Boucler

Résolution du système $A \cdot y_{i+1} = x_i$

$$x_{i+1} = \frac{y_{i+1}}{\|y_{i+1}\|}$$

$$\beta_{i+1} = x_{i+1}^T \cdot A \cdot x_{i+1}$$

$$i = i + 1$$

Jusqu'à convergence

3. Quelles valeurs l'algorithme suivant permet-il d'obtenir ?

$i = 0, x_i = \dots$

Boucler

Résolution du système $(A - \alpha I) \cdot y_{i+1} = x_i$

$$x_{i+1} = \frac{y_{i+1}}{\|y_{i+1}\|}$$

$$\beta_{i+1} = x_{i+1}^T \cdot A \cdot x_{i+1}$$

$$i = i + 1$$

Jusqu'à convergence

Algorithme de Jacobi pour une matrice symétrique

Obtention simultanée de toutes les valeurs propres

Principes

Procédé itératif :
$$\begin{cases} A_1 = A \\ A_{k+1} = \Theta_k^{-1} \cdot A_k \cdot \Theta_k \end{cases}$$

... jusqu'à la convergence : $\lim_{k \rightarrow +\infty} A_k = D = \text{diag}(\lambda_1, \dots, \lambda_n)$

Choix de Θ_k ? Une **matrice orthogonale** ($\Theta_k^{-1} = \Theta_k^T$) car pour tout k :

- A_k est symétrique
- A_k possède les mêmes valeurs propres que A (vecteurs propres différents)

Pour obtenir les valeurs propres de A , il suffit donc que A_k converge vers une **matrice diagonale**.

Soient $A_k = [a_{ij}^{(k)}]$ et $S_k = \sum_{i=1}^n \sum_{j=1}^n (a_{ij}^{(k)})^2 - \sum_{i=1}^n (a_{ii}^{(k)})^2 \geq 0$

$\lim_{k \rightarrow +\infty} A_k = D \iff \lim_{k \rightarrow +\infty} S_k = 0$

Algorithme de Jacobi pour une matrice symétrique

Rotation de Givens

Θ_k choisie comme une **matrice de rotation**, i.e. définie à partir de 3 paramètres :

$$[i, j, \alpha] = [\text{numéro de ligne, numéro de colonne, angle de la rotation}]$$

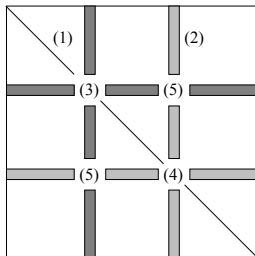
Soient $C = \cos(\alpha)$ et $S = \sin(\alpha)$

$$\Theta_k = \begin{array}{cc|c} & & \\ \hline & 1 & \\ & & 0 \\ & C & S \\ \hline & -S & C \\ & & \\ 0 & & 1 \\ & & \\ \hline & & \\ & & i \\ & & j \end{array}$$

Algorithme de Jacobi pour une matrice symétrique

Impact d'une itération

$A_{k+1} = \Theta_k^{-1} \cdot A_k \cdot \Theta_k$; quels sont les modifications entre k et $k + 1$?



$$(1) \quad a_{ip}^{(k+1)} = a_{pi}^{(k+1)} = C \cdot a_{ip}^{(k)} - S \cdot a_{jp}^{(k)} \quad \forall p = 1, \dots, n \quad p \neq i, j$$

$$(2) \quad a_{jp}^{(k+1)} = a_{pj}^{(k+1)} = S \cdot a_{ip}^{(k)} + C \cdot a_{jp}^{(k)} \quad \forall p = 1, \dots, n \quad p \neq i, j$$

$$(3) \quad a_{ii}^{(k+1)} = C^2 \cdot a_{ii}^{(k)} - 2 \cdot C \cdot S \cdot a_{ij}^{(k)} + S^2 \cdot a_{jj}^{(k)}$$

$$(4) \quad a_{jj}^{(k+1)} = S^2 \cdot a_{ii}^{(k)} + 2 \cdot C \cdot S \cdot a_{ij}^{(k)} + C^2 \cdot a_{jj}^{(k)}$$

$$(5) \quad a_{ij}^{(k+1)} = a_{ji}^{(k+1)} = (C^2 - S^2) \cdot a_{ij}^{(k)} + C \cdot S \cdot (a_{ii}^{(k)} - a_{jj}^{(k)})$$

Algorithme de Jacobi pour une matrice symétrique

Choix des paramètres de la rotation

Des transformations précédentes :

$$S_{k+1} - S_k = 2 \left(a_{ij}^{(k+1)} \right)^2 - 2 \left(a_{ij}^{(k)} \right)^2$$

Pour la convergence, les paramètres sont fixés t.q. $S_{k+1} - S_k$ soit le plus négatif possible :

- On maximise $a_{ij}^{(k)}$ \implies valeur de i et j

$$\left| a_{ij}^{(k)} \right| = \max_{l,m=1,\dots,n, l \neq m} \left| a_{lm}^{(k)} \right|$$

- On annule $a_{ij}^{(k+1)}$ \implies valeur de α

$$a_{ij}^{(k+1)} = (C^2 - S^2) \cdot a_{ij}^{(k)} + C \cdot S \cdot (a_{ii}^{(k)} - a_{jj}^{(k)}) = 0$$

- Si $a_{ij}^{(k)} = a_{ii}^{(k)}$ $\alpha = \text{signe}(a_{ij}^{(k)}) \frac{\pi}{4}$
- Si $a_{ij}^{(k)} \neq a_{ii}^{(k)}$ $\text{tg}(2\alpha) = \frac{2a_{ij}^{(k)}}{a_{ij}^{(k)} - a_{ii}^{(k)}}$ avec $|\alpha| < \frac{\pi}{4}$