

Méthodes numériques de calcul des valeurs et vecteurs propres

Cours d'Analyse Numérique (1 CSC)-2018/2019

Hamid HADDADOU

Plan

1 Introduction

- Exemples motivants
- Utilités des valeurs et vecteurs propres

2 Notion et localisation des valeurs propres

- Définition
- Localisation des valeurs propres

3 Méthodes directes de calcul des valeurs propres

- Limites des méthodes directes

4 Méthodes itératives de calcul des valeurs propres

5 Méthodes partielles

- La méthode de la puissance (itérée)
- La méthode de la puissance inverse

6 Méthodes itératives globales

- La méthode de Jacobi
- La méthode LR de Rutishauser
- La méthode QR de Francis

Exemple motivant 1

Exemple (L'image dans une glace)

- L'image dans une glace de tout vecteur collé à celle-ci est lui même.
- L'image dans une glace de tout vecteur perpendiculaire à celle-ci est le vecteur opposé (de même longueur mais de sens opposé).

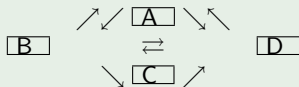
Sachant que l'image dans une glace est une application linéaire, ceci peut se traduire comme suit :

- 1 est une v.p. de cette application et tout vecteur collé à la glace en ait un vecteur propre associé.
- -1 est une v.p. de cette application et tout vecteur perpendiculaire à la glace en ait un vecteur propre associé.

Exemple motivant 2

Exemple (Classement des pages web)

On considère un réseau (graphe) de quatre pages web A, B, C et D avec



L'importance d'une page web P est définie par $\mu_P = \sum_{j=1}^n \frac{1}{N_{P_j}} \mu_{P_j}$, où les P_j ,

$1 \leq j \leq n$ sont les pages qui ont un lien qui pointe vers P et N_{P_j} est le nombre de liens qui sont émis par la page P_j . Le système linéaire $M\mu = \mu$ lié au cas ci-dessus est

$$\begin{pmatrix} 0 & \frac{1}{3} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{3} & 0 & 0 \\ \frac{1}{2} & \frac{1}{3} & 1 & 0 \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \\ \mu_D \end{pmatrix} = \begin{pmatrix} \mu_A \\ \mu_B \\ \mu_C \\ \mu_D \end{pmatrix}.$$

Ainsi, μ représente un vecteur propre de la matrice du système associé à la valeur propre 1. Ce problème admet une solution (car 1 est réellement une v.p. de la matrice ci-dessus!!!!).

Utilités des valeurs et vecteur propres

Les valeurs et vecteurs propres jouent un rôle très important dans beaucoup de domaines. Pour les domaines appliqués voir les exemples motivants ci-dessus. Par ailleurs, en mathématiques par exemple,

- les valeurs et vecteurs propres interviennent dans la résolution des systèmes linéaires (chapitres 1-2),
- les valeurs et vecteurs propres interviennent en analyse des données,
- les valeurs et vecteurs propres peuvent intervenir dans la résolution des systèmes d'équations différentielles.

Pour calculer les valeurs propres il y'a deux types de méthodes, à savoir, les méthodes directes et les méthodes itératives.

- **Les méthodes directes** permettent d'obtenir les valeurs propres **en exploitant que le polynôme caractéristique** de la matrice.
- **Les méthodes itératives** permettent d'obtenir les valeurs propres **en exploitant la matrice elle même**.

Définition

Définition

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Alors, on dit qu'un complexe λ est valeurs propre (V.P.) de A si $\exists X \in \mathbb{C}^n \setminus \{0\}$ tel que $AX = \lambda X$, X est appelé vecteur propre de A associé à λ

Remarque

- Les valeurs propres d'une matrice $A \in \mathcal{M}_n(\mathbb{R})$ **symétrique** sont toutes réelles.
- Deux matrices semblables ont les mêmes valeurs propres.

Exemple

On a $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} -1 \\ 1 \end{pmatrix} = - \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ et $\begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 3 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$.

Donc, si on pose $A = \begin{pmatrix} 1 & 2 \\ 2 & 1 \end{pmatrix}$, il vient

- ${}^t(-1 \ 1)$ est un vecteur propre de A associé à la valeur propre -1 ,
- ${}^t(1 \ 1)$ est un vecteur propre de A associé à la valeur propre 3 .

Localisation des valeurs propres

On a le résultat suivant :

Théorème (Gershgorin-Hadamard)

Soit $D_i = \{z \in \mathbb{C} \text{ tq } |z - a_{ii}| \leq \sum_{j=1, j \neq i}^n |a_{ij}|\}$. Alors, on a

$$Sp(A) \subset \bigcup_{i=1}^n D_i,$$

Remarque

Comme A et tA ont les mêmes valeurs propres, alors $Sp(A) \subset \bigcup_{i=1}^n C_i$,

où $C_i = \{z \in \mathbb{C} \text{ tq } |z - a_{ii}| \leq \sum_{i=1, i \neq j}^n |a_{ij}|\}$.

Exemple

Les valeurs propres de la matrice $A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 1 \\ 3 & 1 & 1 \end{pmatrix}$ sont dans $[-4,6]$.

Calcul des valeurs propres via le polynôme caractéristique

Définition

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Le polynôme caractéristique de A est défini par

$$P_A(X) = \det(A - XI_n).$$

Proposition

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Alors, on a $(\lambda \text{ valeur propre de } A) \Leftrightarrow (P_A(\lambda) = 0)$.

Remarque

Il y a des méthodes qui donnent P_A sans utiliser le déterminant. Par exemple, la méthode de Krylov qui se base sur le théorème de Cayley-Hamilton qui dit que

$$P_A(A) = 0_{\mathcal{M}_n(\mathbb{R})}.$$

Illustration de la méthode de Krylov

Considérons la matrice

$$A = \begin{pmatrix} 2 & -1 & -2 \\ 1 & 1 & 1 \\ 2 & 3 & -1 \end{pmatrix}.$$

$P_A(X)$ peut s'écrire sous la forme $P_A(X) = -X^3 + a_2X^2 + a_1X + a_0$. On prend $v \in \mathbb{R}_*^n$, par exemple, $v = {}^t(1 \ 1 \ 1)$ il vient du fait que $P_A(A) = 0_{\mathcal{M}_n(\mathbb{R})}$:

$$A^3v = a_0v + a_1Av + a_2A^2v.$$

Mais,

$$Av = \begin{pmatrix} -1 \\ 3 \\ 4 \end{pmatrix}, \quad A^2v = A(Av) = \begin{pmatrix} -13 \\ 6 \\ 3 \end{pmatrix} \text{ et } A^3v = A(A^2v) = \begin{pmatrix} -38 \\ -4 \\ -11 \end{pmatrix}.$$

il vient alors

$$\begin{cases} a_0 - a_1 - 13a_2 = -38, \\ a_0 + 3a_1 + 6a_2 = -4, \\ a_0 + 4a_1 + 3a_2 = -11 \end{cases} \Rightarrow \begin{cases} a_0 = -13, \\ a_1 = -1, \\ a_2 = 2 \end{cases} \Rightarrow P_A(X) = -X^3 + 2X^2 - X - 13.$$

Limites des méthodes directes

A ne pas faire !

Si la taille de la matrice est grande ou si les coefficients ne sont pas entiers il est vivement déconseillé de calculer les valeurs propres à l'aide du polynôme caractéristique (en le calculant par le déterminant ou autre technique) car ça donne lieu à un algorithme instable (très sensible aux erreurs d'arrondi).

Dans le cas des méthodes qui exploitent que le polynômes caractéristiques, deux difficultés surgissent, la première est que le polynôme caractéristique d'une matrice A est donné par $\det(A - XI)$ et la seconde est la détermination des racines. Pour calculer les coefficients du polynôme caractéristique sans passer par le déterminant, on peut citer les méthodes de Krylov, de Leverrier et de Souriau-Faddeev. En revanche pour la détermination des racines (si n n'est pas très grand) on peut utiliser les méthodes de résolution des équations non linéaires qui vont être présentées dans le chapitre 4. Mais, pour $n > 10$ cette façon de faire donne lieu à des algorithmes **instables** (à cause du calcul en virgule flottante) d'où sa limite d'utilisation en pratique.

En effet, considérons l'exemple (proposé par J. H. Wilkinson dans les années 60), le problème du calcul des valeurs propres de la matrice diagonale $A = \text{diag}(1, \dots, n)$, dont le polynôme caractéristique est

$$P_A(\lambda) = (1 - \lambda) \dots (n - \lambda) = (-1)^n \lambda^n + a_{n-1} \lambda^{n-1} + \dots + a_1 \lambda + a_0.$$

Limites des méthodes directes (suite)

Supposons que les coefficients calculés sont entachés d'erreurs . Nous calculons alors les racines du polynôme

$$P_A(\lambda) = (-1)^n \lambda^n + \widetilde{a}_{n-1} \lambda^{n-1} + \dots + \widetilde{a}_1 \lambda + \widetilde{a}_0.$$

La perturbation des coefficients provoque une erreur dans le calcul des racines du polynôme. On fait remarquer qu'en réalité, pour n grand, le calcul numérique des valeurs propres ne se ramène pas au calcul numérique des racines d'un polynôme, mais c'est l'inverse qui est vrai. Plus précisément, chaque équation polynômiale peut être associée à un polynôme de la forme $P(X) = a_0 + a_1 X + a_2 X^2 + \dots + a_{n-1} X^{n-1} + X^n$, et ça correspond exactement au $(-1)^n P_A(X)$ où A est la matrice suivante (appelée matrice de Frobenius ou **compagnon de P**)

$$\begin{pmatrix} 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & \cdots & 0 & a_1 \\ & 0 & 1 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & a_{n-1} \end{pmatrix}$$

Voir l'exercice 3.

Méthodes itératives de calcul des valeurs propres

Les méthodes itératives proposent d'exploiter la matrice au lieu du polynôme caractéristique pour avoir plus d'informations (le polynôme caractéristique contient $(n + 1)$ informations par contre la matrice contient n^2 informations). Dans ce chapitre, on s'intéresse à ces dernières méthodes. Il existe deux types de ces méthodes. D'une manière plus précise Soient A une matrice de $\mathcal{M}_n(\mathbb{R})$ et $\lambda_1, \dots, \lambda_n$ ces valeurs propres tels que $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Méthodes itératives partielles

Les méthodes partielles visent le calcul

- de la plus grande valeur propre en module λ_1 (si elle est unique) !,
- la plus petite valeur propre en module λ_n (si elle est unique) !,
- ou encore la valeur propre la plus proche d'une valeur donnée.

Ces méthodes permettent de trouver le vecteur propre correspondant.

Méthodes itératives globales

Les méthodes globales s'intéressent à trouver le spectre tout entier c'est à dire $\lambda_1, \dots, \lambda_n$.

Ces méthodes présentent l'inconvénient de ne pas donner en général directement les vecteurs propres.

Méthodes itératives de calcul des valeurs propres

Les méthodes itératives proposent d'exploiter la matrice au lieu du polynôme caractéristique pour avoir plus d'informations (le polynôme caractéristique contient $(n + 1)$ informations par contre la matrice contient n^2 informations). Dans ce chapitre, on s'intéresse à ces dernières méthodes. Il existe deux types de ces méthodes. D'une manière plus précise Soient A une matrice de $\mathcal{M}_n(\mathbb{R})$ et $\lambda_1, \dots, \lambda_n$ ces valeurs propres tels que $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Méthodes itératives partielles

Les méthodes partielles visent le calcul

- de la plus grande valeur propre en module λ_1 (si elle est unique) !,
- la plus petite valeur propre en module λ_n (si elle est unique) !,
- ou encore la valeur propre la plus proche d'une valeur donnée.

Ces méthodes permettent de trouver le vecteur propre correspondant.

Méthodes itératives globales

Les méthodes globales s'intéressent à trouver le spectre tout entier c'est à dire $\lambda_1, \dots, \lambda_n$.

Ces méthodes présentent l'inconvénient de ne pas donner en général directement les vecteurs propres.

Méthodes itératives de calcul des valeurs propres

Les méthodes itératives proposent d'exploiter la matrice au lieu du polynôme caractéristique pour avoir plus d'informations (le polynôme caractéristique contient $(n + 1)$ informations par contre la matrice contient n^2 informations). Dans ce chapitre, on s'intéresse à ces dernières méthodes. Il existe deux types de ces méthodes. D'une manière plus précise Soient A une matrice de $\mathcal{M}_n(\mathbb{R})$ et $\lambda_1, \dots, \lambda_n$ ces valeurs propres tels que $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Méthodes itératives partielles

Les méthodes partielles visent le calcul

- de la plus grande valeur propre en module λ_1 (si elle est unique) !,
- la plus petite valeur propre en module λ_n (si elle est unique) !,
- ou encore la valeur propre la plus proche d'une valeur donnée.

Ces méthodes permettent de trouver le vecteur propre correspondant.

Méthodes itératives globales

Les méthodes globales s'intéressent à trouver le spectre tout entier c'est à dire $\lambda_1, \dots, \lambda_n$.

Ces méthodes présentent l'inconvénient de ne pas donner en général directement les vecteurs propres.

Méthodes itératives de calcul des valeurs propres

Les méthodes itératives proposent d'exploiter la matrice au lieu du polynôme caractéristique pour avoir plus d'informations (le polynôme caractéristique contient $(n + 1)$ informations par contre la matrice contient n^2 informations). Dans ce chapitre, on s'intéresse à ces dernières méthodes. Il existe deux types de ces méthodes. D'une manière plus précise Soient A une matrice de $\mathcal{M}_n(\mathbb{R})$ et $\lambda_1, \dots, \lambda_n$ ces valeurs propres tels que $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Méthodes itératives partielles

Les méthodes partielles visent le calcul

- de la plus grande valeur propre en module λ_1 (si elle est unique) !,
- la plus petite valeur propre en module λ_n (si elle est unique) !,
- ou encore la valeur propre la plus proche d'une valeur donnée.

Ces méthodes permettent de trouver le vecteur propre correspondant.

Méthodes itératives globales

Les méthodes globales s'intéressent à trouver le spectre tout entier c'est à dire $\lambda_1, \dots, \lambda_n$.

Ces méthodes présentent l'inconvénient de ne pas donner en général directement les vecteurs propres.

La méthode de la puissance (itérée)

Algorithme de la puissance itérée

Détermination de la plus grande valeur propre au moyen de l'algorithme suivant

$$\begin{cases} q^{(0)} \in \mathbb{R}^n \text{ tel que } \|q^{(0)}\|_2 = 1, \\ x^{(k)} = Aq^{(k-1)}, \forall k \in \mathbb{N}^* \\ q^{(k)} = \frac{1}{\|x^{(k)}\|_2} x^{(k)}. \end{cases}$$

Théorème (CS de convergence de l'algorithme de la puissance itérée)

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Supposons que

- A est diagonalisable,
- la valeur propre la plus grande en module de A est unique (pas nécessairement simple) notée λ_1 .

Alors, pour $q^{(0)} \in \mathbb{R}^n$ non orthogonal au sous-espace propre à gauche relatif à λ_1 , on a

- 1 $q = \lim_{k \rightarrow \infty} \left(\frac{\lambda_1}{|\lambda_1|} \right)^k q^{(k)}$ est un vecteur propre associé à λ_1 et $\|q\|_2 = 1$,
- 2 $\lim_{k \rightarrow \infty} \|Aq^{(k)}\|_2 = |\lambda_1|$,
- 3 $\lim_{k \rightarrow \infty} \alpha^k = \lambda_1$ avec $\alpha^{(k)} \doteq x^{(k+1)} q^{(k)}$.

La méthode de la puissance (itérée)

Algorithme de la puissance itérée

Détermination de la plus grande valeur propre au moyen de l'algorithme suivant

$$\begin{cases} q^{(0)} \in \mathbb{R}^n \text{ tel que } \|q^{(0)}\|_2 = 1, \\ x^{(k)} = Aq^{(k-1)}, \forall k \in \mathbb{N}^* \\ q^{(k)} = \frac{1}{\|x^{(k)}\|_2} x^{(k)}. \end{cases}$$

Théorème (CS de convergence de l'algorithme de la puissance itérée)

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Supposons que

- A est diagonalisable,
- la valeur propre la plus grande en module de A est unique (pas nécessairement simple) notée λ_1 .

Alors, pour $q^{(0)} \in \mathbb{R}^n$ non orthogonal au sous-espace propre à gauche relatif à λ_1 , on a

$$\textcircled{1} \quad q = \lim_{k \rightarrow \infty} \left(\frac{\lambda_1}{|\lambda_1|} \right)^k q^{(k)} \text{ est un vecteur propre associé à } \lambda_1 \text{ et } \|q\|_2 = 1,$$

$$\textcircled{2} \quad \lim_{k \rightarrow \infty} \|Aq^{(k)}\|_2 = |\lambda_1|,$$

$$\textcircled{3} \quad \lim_{k \rightarrow \infty} \alpha^k = \lambda_1 \text{ avec } \alpha^{(k)} \doteq x^{(k+1)} q^{(k)}.$$

La méthode de la puissance (itérée)

Algorithme de la puissance itérée

Détermination de la plus grande valeur propre au moyen de l'algorithme suivant

$$\begin{cases} q^{(0)} \in \mathbb{R}^n \text{ tel que } \|q^{(0)}\|_2 = 1, \\ x^{(k)} = Aq^{(k-1)}, \forall k \in \mathbb{N}^* \\ q^{(k)} = \frac{1}{\|x^{(k)}\|_2} x^{(k)}. \end{cases}$$

Théorème (CS de convergence de l'algorithme de la puissance itérée)

Soit $A \in \mathcal{M}_n(\mathbb{R})$. Supposons que

- A est diagonalisable,
- la valeur propre la plus grande en module de A est unique (pas nécessairement simple) notée λ_1 .

Alors, pour $q^{(0)} \in \mathbb{R}^n$ non orthogonal au sous-espace propre à gauche relatif à λ_1 , on a

❶ $q = \lim_{k \rightarrow \infty} \left(\frac{\lambda_1}{|\lambda_1|} \right)^k q^{(k)}$ est un vecteur propre associé à λ_1 et $\|q\|_2 = 1$,

❷ $\lim_{k \rightarrow \infty} \|Aq^{(k)}\|_2 = |\lambda_1|$,

❸ $\lim_{k \rightarrow \infty} \alpha^k = \lambda_1$ avec $\alpha^{(k)} \doteq x^{(k+1)} q^{(k)}$.

Remarque et Exemple (à la main !)

Remarque

- Dans la pratique le choix de $q^{(0)}$ avec $\|q\|_2 = 1$ est au hasard, ce qui ne pose pas de problème grâce (paradoxalement) aux erreurs d'arrondi (bénéfiques!!!).
- La limite qui donne q est donnée dans le sens de valeurs d'adhérence.

Exemple

La matrice $A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{pmatrix}$ admet une seul v.p. de plus grande module

$$\lambda_1 = 2(1 + \cos(\frac{\pi}{4})) \approx 3.41421356.$$

λ_1 est unique + A diagonalisable (car symétrique) \Rightarrow Méthode applicable

Si on prend $q^{(0)} = {}^t(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$, la méthode de la puissance itérée donne

$$\begin{cases} x^{(1)} = \frac{1}{\sqrt{3}} {}^t(3, 4, 3) & \text{et} & q^{(1)} = \frac{1}{\sqrt{34}} {}^t(3, 4, 3), \\ x^{(2)} = \frac{1}{\sqrt{34}} {}^t(10, 14, 10) & \text{et} & q^{(2)} = \frac{1}{\sqrt{196}} {}^t(10, 14, 10). \end{cases}$$

Ainsi, on peut "approcher" la valeur propre la plus grande en module par

$$\begin{cases} \alpha^{(1)} = {}^t x^{(1)} q^{(0)} = \frac{10}{3} \approx 3.333333333, \\ \alpha^{(2)} = {}^t x^{(2)} q^{(1)} = \frac{116}{34} \approx 3.411764705. \end{cases}$$

Résultats utiles !

Les deux résultats (à admettre) représentent des conditions suffisantes simples à vérifier pour qu'une matrice soit diagonalisable ou sa valeur propre de plus grand module λ_1 soit unique.

Proposition

Toute matrice symétrique est diagonalisable.

Théorème (de Perron)

Soit $A \in \mathcal{M}_n(\mathbb{R})$ **strictement positive** (ie tout ses coefficients sont strictement positifs). Alors,

- ① $\rho(A)$ est une valeur propre de A ,
- ② $\rho(A) > 0$,
- ③ $\rho(A)$ est simple,
- ④ $\rho(A)$ est dominante (unique),
- ⑤ il existe un vecteur propre X dont toutes les composantes sont strictement positives associé à la valeur propre $\rho(A)$,
- ⑥ tout vecteur propre positif de A est un multiple positif du vecteur de Perron qui est défini (d'une manière unique) par $P = \frac{1}{\|X\|} X$.

Remarque et Exemple (avec matlab)

Remarque

- Dans le théorème qui donne la convergence de l'algorithme de la puissance itérée, on peut prendre $q^{(0)}$ quelconque même $\|q\|_2 \neq 1$ grâce à normalisation effectué à partir de la deuxième itération.
- Si le problème aux valeurs propres est bien conditionné (cette notion est différente d'un système linéaire bien), on considère le test d'arrêt des incréments $|\alpha^{(k)} - \alpha^{(k-1)}| \leq \varepsilon$ (ε est la précision).

Exemple

Considérons la matrice $A = \text{hilb}(4)$, où $a_{ij} = \frac{1}{i+j-1}$.

- 1 Vérifier théoriquement que l'algorithme de la puissance itérée appliqué à A converge.
- 2 Calculer numériquement la valeur propre dominante de A en utilisant cet algorithme avec 4 chiffres décimaux (illustrer la convergence).

La méthode de la puissance inverse

Idée et Algorithme

Détermination de la plus petite valeur propre d'une matrice. Si A est diagonalisable dont les v.p. sont telles que $0 < |\lambda_1| \leq |\lambda_2| \leq \dots \leq |\lambda_n|$. Donc, A est inversible et A^{-1} est diagonalisable dont les valeurs propres sont telles que

$$\left| \frac{1}{\lambda_n} \right| \leq \left| \frac{1}{\lambda_{n-1}} \right| \leq \dots \leq \left| \frac{1}{\lambda_1} \right|.$$

Si λ_1 est unique on peut appliquer l'algorithme de la puissance à A^{-1} pour approcher $\frac{1}{\lambda_1}$. L'algorithme de la puissance inverse sera donné par

$$\begin{cases} q^{(0)} \in \mathbb{R}^n \text{ tel que } \|q^{(0)}\|_2 = 1, \\ \text{Trouver } x^{(k)} \text{ tel que } Ax^{(k)} = q^{(k-1)}, \forall k \in \mathbb{N}^* \\ q^{(k)} = \frac{1}{\|x^{(k)}\|_2} x^{(k)}. \end{cases}$$

Remarque

Dans la méthode de la puissance inverse on ne calcule pas A^{-1} mais on résout des systèmes (Trouver $x^{(k)}$ tel que $Ax^{(k)} = q^{(k-1)}$).

Méthode de Jacobi et Rotations de Givens

Rotations de Givens

Pour tout couple (p, q) tel que $1 \leq p < q \leq n$ et $\theta \in [0, 2\pi[$ la matrice de rotation de Givens $Q_{p,q}(\theta)$ est définie par

$$Q_{p,q}(\theta) = \begin{pmatrix} 1 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ 0 & \ddots & \ddots & & & & & & & & \\ & \ddots & 1 & 0 & \dots & \dots & \dots & \dots & \dots & \dots & 0 \\ & & 0 & \cos \theta & 0 & \dots & 0 & \sin \theta & 0 & & \\ & & & 0 & 1 & 0 & \dots & 0 & & & \\ & & & \vdots & \ddots & \ddots & \ddots & \vdots & & & \\ & & & 0 & & \ddots & 1 & 0 & & & \\ & & & -\sin \theta & 0 & \dots & 0 & \cos \theta & 0 & & \\ & & 0 & \dots & \dots & \dots & \dots & 0 & 1 & \ddots & \\ & & & & & & & & \ddots & \ddots & 0 \\ 0 & & & & & & & & & 0 & 1 \end{pmatrix}$$

Méthode de Jacobi et Rotations de Givens (suite)

Cette méthode concerne les matrices symétriques. La méthode de Jacobi propose de construire à l'aide des rotations de Givens une suite de matrices $(A^{(k)})_{k \in \mathbb{N}^*}$ orthogonalement semblables à la matrice d'origine qui converge vers une matrice diagonale.

Pour une matrice symétrique $A \in \mathcal{M}_n(\mathbb{R})$ et pour $p < q$, on peut trouver une valeur de θ dans $] -\frac{\pi}{4}, \frac{\pi}{4}]$ tel que la matrice

$$M = {}^t Q_{p,q}(\theta) A Q_{p,q}(\theta)$$

vérifie $m_{pq} = 0$ et $m_{qp} = 0$. En effet

→ Si $a_{pq} = 0$, on prend $\theta = 0$.

→ Si $a_{pq} \neq 0$. Donc, si on pose

$$\beta = \frac{a_{qq} - a_{pp}}{2a_{pq}}, t = \begin{cases} -\beta - \sqrt{\beta^2 + 1} & \text{si } \beta < 0, \\ -\beta + \sqrt{\beta^2 + 1} & \text{si } \beta \geq 0, \end{cases}$$

on détermine enfin θ comme solution de

$$\cos \theta = \frac{1}{\sqrt{1+t^2}} \quad \text{et} \quad \sin \theta = \frac{t}{\sqrt{1+t^2}} \quad (\text{on a aussi } \frac{1}{\tan 2\theta} = \beta).$$

L'algorithme

Algorithme

La méthode de Jacobi consiste à construire une suite de matrices symétriques $(A^{(k)})_{k \in \mathbb{N}^*}$ orthogonalement semblables à A . Plus précisément, on prend $A^{(0)} = A$ et pour tout k , (p, q) tq $1 \leq p < q \leq n$ et A_k est construite par

$$A^{(k+1)} = {}^t Q_{p,q}(\theta) A^{(k)} Q_{p,q}(\theta)$$

où θ est choisit dans $] -\frac{\pi}{4}, \frac{\pi}{4}]$ tq $a_{pq}^{(k+1)} = a_{qp}^{(k+1)} = 0$.

Remarque

- La matrice $A^{(k)}$ est orthogonalement semblable à A , avec $A^{(k)} = {}^t Q^{(k)} A Q^{(k)}$ où $Q^{(k)}$ est le produit de k matrices de Givens. Alors $Q^{(k)}$ converge, en général, vers une matrice Q dont les colonnes constituent une base orthonormale de vecteurs propres de A .
- La diagonale de A est transformée de la manière suivante :
 - les coefficients d'indices différents de (p, p) et de (q, q) ne sont pas changés.
 - le coefficient d'indice (p, p) est transformée en $a_{pp} - a_{pq} \tan \theta$ et le coefficient d'indice (q, q) est transformé en $a_{qq} + a_{pq} \tan \theta$.

Convergence de la méthode de Jacobi et exemple d'application

Théorème

Si on choisit (p, q) tel que $a_{pq}^{(k)}$ soit le coefficient extradiagonal de $A^{(k)}$ de plus grand module alors la méthode de Jacobi (dite dans ce cas classique) converge vers une matrice diagonale.

Exemple

$$A = \begin{pmatrix} 1 & 2 & 2 \\ 2 & 1 & 2 \\ 2 & 2 & 1 \end{pmatrix}.$$

$$A^{(1)} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 3 & \frac{4}{\sqrt{2}} \\ 0 & \frac{4}{\sqrt{2}} & 1 \end{pmatrix}, \quad Q^{(1)} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix},$$

et

$$A^{(2)} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad Q^{(2)} = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{6}} \\ 0 & \frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \end{pmatrix}.$$

Ainsi, les valeurs propres de A sont -1 , 5 et -1 , et les colonnes de $Q^{(2)}$ constituent une base de vecteurs propres associé.

Algorithme LR de Rutishauser

Cette méthode est basée sur la d écomposition LU et elle est définie comme suit : Supposons que A est inversible et admet une factorisation LU .

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = L_1 U_1.$$

On définit alors

$$A_2 \doteq U_1 L_1.$$

Etape2 : On détermine (si possible !) L_2 et U_2 telles que

$$A_2 = L_2 U_2.$$

On définit alors

$$A_3 \doteq U_2 L_2.$$

Etape k : On a $A_k \doteq U_{k-1} L_{k-1}$ et on détermine (si possible !) L_k et U_k telles que A_k

$$A_k = L_k U_k.$$

On définit alors

$$A_{k+1} \doteq U_k L_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Algorithme LR de Rutishauser

Cette méthode est basée sur la d écomposition LU et elle est définie comme suit : Supposons que A est inversible et admet une factorisation LU .

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = L_1 U_1.$$

On définit alors

$$A_2 \doteq U_1 L_1.$$

Etape2 : On détermine (si possible !) L_2 et U_2 telles que

$$A_2 = L_2 U_2.$$

On définit alors

$$A_3 \doteq U_2 L_2.$$

Etape k : On a $A_k \doteq U_{k-1} L_{k-1}$ et on détermine (si possible !) L_k et U_k telles que A_k

$$A_k = L_k U_k.$$

On définit alors

$$A_{k+1} \doteq U_k L_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Algorithme LR de Rutishauser

Cette méthode est basée sur la décomposition LU et elle est définie comme suit : Supposons que A est inversible et admet une factorisation LU .

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = L_1 U_1.$$

On définit alors

$$A_2 \doteq U_1 L_1.$$

Etape2 : On détermine (si possible !) L_2 et U_2 telles que

$$A_2 = L_2 U_2.$$

On définit alors

$$A_3 \doteq U_2 L_2.$$

Etape k : On a $A_k \doteq U_{k-1} L_{k-1}$ et on détermine (si possible !) L_k et U_k telles que A_k

$$A_k = L_k U_k.$$

On définit alors

$$A_{k+1} \doteq U_k L_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Algorithme LR de Rutishauser

Cette méthode est basée sur la décomposition LU et elle est définie comme suit : Supposons que A est inversible et admet une factorisation LU .

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = L_1 U_1.$$

On définit alors

$$A_2 \doteq U_1 L_1.$$

Etape2 : On détermine (si possible !) L_2 et U_2 telles que

$$A_2 = L_2 U_2.$$

On définit alors

$$A_3 \doteq U_2 L_2.$$

Etape k : On a $A_k \doteq U_{k-1} L_{k-1}$ et on détermine (si possible !) L_k et U_k telles que A_k

$$A_k = L_k U_k.$$

On définit alors

$$A_{k+1} \doteq U_k L_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Convergence de LR

Remarque (Très importante !!)

Toutes les matrices A_k sont semblables à A . En effet, comme toutes les L_k sont inversibles, on a

$$A_2 \doteq U_1 L_1 \implies L_1 A_2 = L_1 U_1 L_1 = A_1 L_1 \implies A_2 = (L_1)^{-1} A_1 L_1.$$

Ainsi, $A_{k+1} = (L_1 \dots L_k)^{-1} A_1 (L_1 \dots L_k)$. Donc, en posant $R_k = L_1 \dots L_k$, il vient

$$A_{k+1} = R_k^{-1} A_1 R_k \doteq R_k^{-1} A R_k.$$

Théorème

Supposons que A est inversible, les valeurs propres de A sont réelles et distincts en valeur absolue, et $\forall k \in \mathbb{N}^*$ A_k admet une la factorisation LU. Alors

- la suite $(A_k)_{k \in \mathbb{N}^*}$ converge avec $\lim_{k \rightarrow \infty} A_k = A_\infty$,
- A_∞ est une matrice triangulaire supérieure,
- les éléments diagonaux de A_∞ sont les valeurs propres de A ,
- si v est un vecteur propre de A_∞ associée à une valeur propre λ , alors $R_\infty v$ est un vecteur propre de A avec $R_\infty \doteq \lim_{k \rightarrow \infty} R_k$

Algorithme QR de Francis

Cette méthode est basée sur la décomposition QR et elle est définie comme suit :

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = Q_1 R_1.$$

On définit alors

$$A_2 \doteq R_1 Q_1.$$

Etape2 : On détermine Q_2 et R_2 telles que

$$A_2 = Q_2 R_2.$$

On définit alors

$$A_3 \doteq R_2 Q_2.$$

Etape k : On a $A_k \doteq R_{k-1} Q_{k-1}$ et on détermine Q_k et R_k telles que A_k

$$A_k = Q_k R_k.$$

On définit alors

$$A_{k+1} \doteq R_k Q_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Algorithme QR de Francis

Cette méthode est basée sur la décomposition QR et elle est définie comme suit :

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = Q_1 R_1.$$

On définit alors

$$A_2 \doteq R_1 Q_1.$$

Etape2 : On détermine Q_2 et R_2 telles que

$$A_2 = Q_2 R_2.$$

On définit alors

$$A_3 \doteq R_2 Q_2.$$

Etape k : On a $A_k \doteq R_{k-1} Q_{k-1}$ et on détermine Q_k et R_k telles que A_k

$$A_k = Q_k R_k.$$

On définit alors

$$A_{k+1} \doteq R_k Q_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Algorithme QR de Francis

Cette méthode est basée sur la décomposition QR et elle est définie comme suit :

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = Q_1 R_1.$$

On définit alors

$$A_2 \doteq R_1 Q_1.$$

Etape2 : On détermine Q_2 et R_2 telles que

$$A_2 = Q_2 R_2.$$

On définit alors

$$A_3 \doteq R_2 Q_2.$$

Etape k : On a $A_k \doteq R_{k-1} Q_{k-1}$ et on détermine Q_k et R_k telles que A_k

$$A_k = Q_k R_k.$$

On définit alors

$$A_{k+1} \doteq R_k Q_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Algorithme QR de Francis

Cette méthode est basée sur la décomposition QR et elle est définie comme suit :

Etape1 : Soit $A \in \mathcal{M}_n(\mathbb{R})$. On pose $A_1 = A$ et on la factorise sous la forme

$$A_1 = Q_1 R_1.$$

On définit alors

$$A_2 \doteq R_1 Q_1.$$

Etape2 : On détermine Q_2 et R_2 telles que

$$A_2 = Q_2 R_2.$$

On définit alors

$$A_3 \doteq R_2 Q_2.$$

Etape k : On a $A_k \doteq R_{k-1} Q_{k-1}$ et on détermine Q_k et R_k telles que A_k

$$A_k = Q_k R_k.$$

On définit alors

$$A_{k+1} \doteq R_k Q_k$$

En itérant ce procédé, on obtient une suite $(A_k)_{k \in \mathbb{N}^*}$.

Convergence de l'algorithme QR

Remarque

Dans la pratique on préfère utiliser la méthode QR au lieu LR, à cause de la précision et le fait qu'on peut toujours factoriser sous la forme QR à l'inverse de LU.

Remarque (Très importante !!)

Toutes les matrices A_k sont semblables à A .

Théorème

Soit A une matrice de $\mathcal{M}_n(\mathbb{R})$ symétrique définie positive. Alors, la suite des matrices $(A_k)_{k \in \mathbb{N}^}$ définie par la méthode QR converge vers une matrice diagonale.*

Corollaire

Les valeurs propres d'une matrice $A \in \mathcal{M}_n(\mathbb{R})$ symétrique définie positive sont les éléments diagonaux de la matrice limite de la suite $(A_k)_{k \in \mathbb{N}^}$.*

Utilisation de la méthode QR dans la pratique

Sous cette forme naïve, l'algorithme est très mauvais numériquement si la matrice A est pleine et surtout si sa taille n est grande. En effet, il est très coûteux car chaque décomposition QR demande de l'ordre de n^3 opérations et il est aussi peu stable si la décomposition est obtenue par le procédé de Gram-Schmidt. Pour ça en pratique on applique l'algorithme qu'aux matrices de Hessenberg. On transforme d'abord la matrice initiale A en une matrice orthogonale semblable de type Hessenberg supérieure et on applique à cette dernière pour obtenir les valeurs propres de A .

Théorème (Cas particulier)

Si la matrice initiale A est Hessemberg supérieure, c'est à dire : H est de la forme

$$H = \begin{pmatrix} * & * & \cdots & * & * \\ * & * & \cdots & * & * \\ 0 & * & \cdots & * & * \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & * & * \end{pmatrix}$$

alors tous les A_k données par l'algorithme QR sont aussi Hessemberg supérieures.

Pour justifier cette modification de l'algorithme, on donne le résultat suivant :

Théorème

Soit A une matrice de $\mathcal{M}_n(\mathbb{R})$. Il existe une matrice orthogonale $Q \in \mathcal{M}_n(\mathbb{R})$ telle que

$${}^tQAQ = H.$$

où la matrice H est Hessemberg supérieure.